



docker

1장

Docker 이해 및 설치

전체 내용

Docker란?

Docker 설치하기

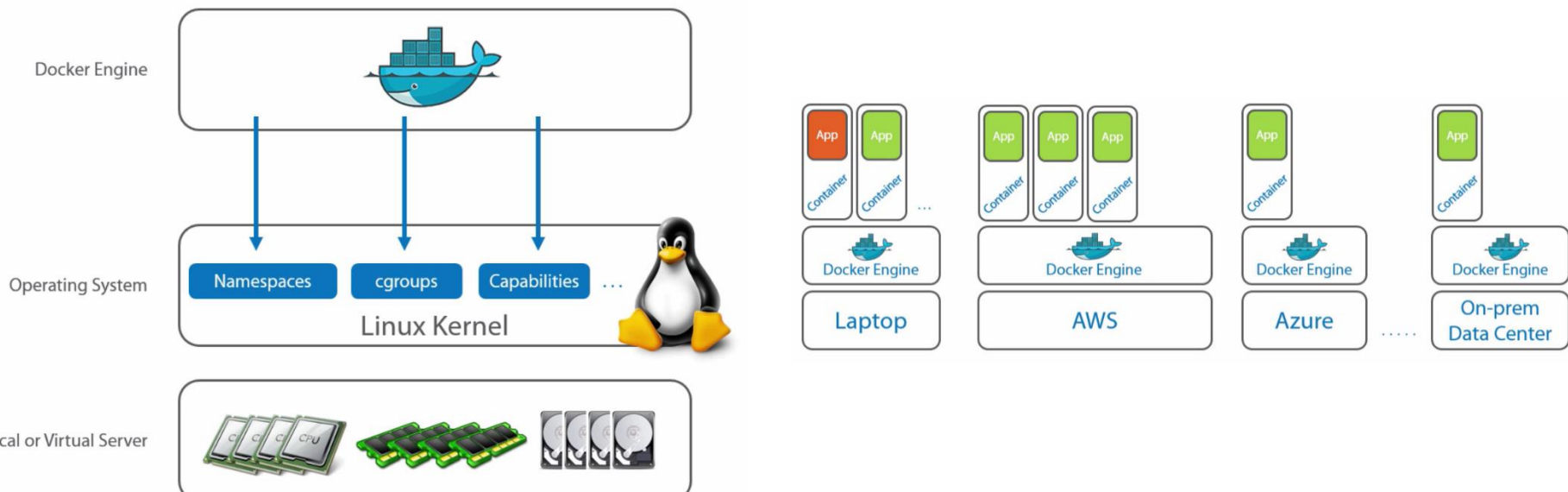
Docker 도움말
활용하기

쉬어 가는 코너

1-Docker란?

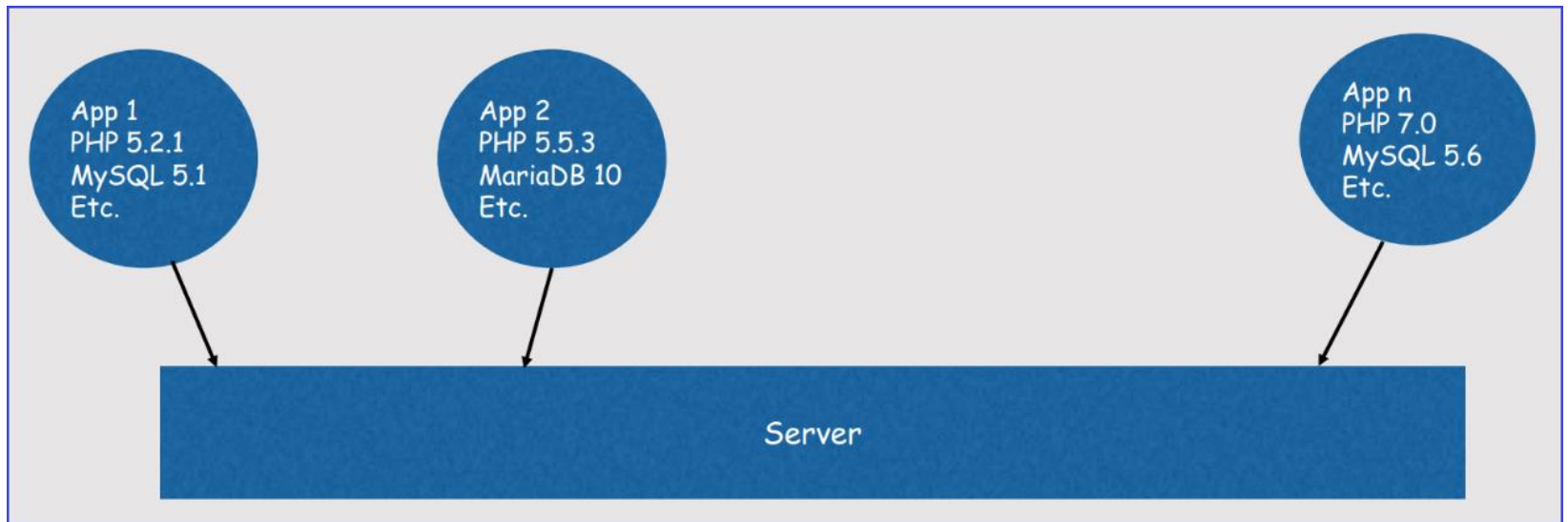
- Docker란?

- 2013년도에 Docker Inc.에서 출시하였고, Go Language로 작성되었다
- Docker는 Linux Container에 여러 기능을 추가하여 Application을 Container 형식으로 쉽게 사용할 수 있는 Open Source Project다
- Docker는 VM에 비해 **성능 손실이 거의 없고**, 상대적으로 **가볍고**, 어디서든 실행할 수 있고(**이식성**), 이미지를 **중앙에 저장하여** 편리하게 사용할 수 있다
- Public Cloud에서 운영하는 수 많은 VM에 Application을 설치하고 구성하는 문제를 해결한 것이 Docker 기술이다



1-Docker란?

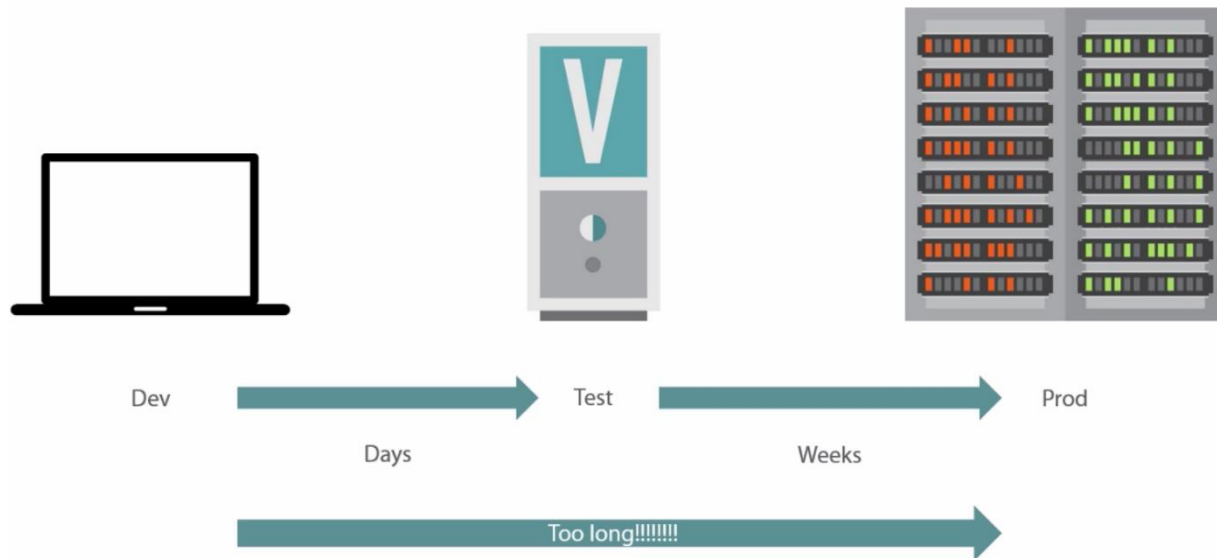
- 개발 부분에서 Docker의 필요성
 - 하나의 서버에 여러 개의 Project 운영하기
 - PHP Web Application을 개발을 하여 운영을 할 때, 한 서버에 여러 개의 Web site를 운영할 수 없다. 그 이유는 각 Project마다 다른 버전의 PHP를 사용하고, Database 버전도 다른 것을 사용하기 때문이다



- 이것을 해결하기 위해 Host 컴퓨터에 여러 개의 VM을 설치하여 운영할 수도 있지만, VM 운영을 무겁고, 많은 자원을 사용하므로 성능 부분에 문제가 있다

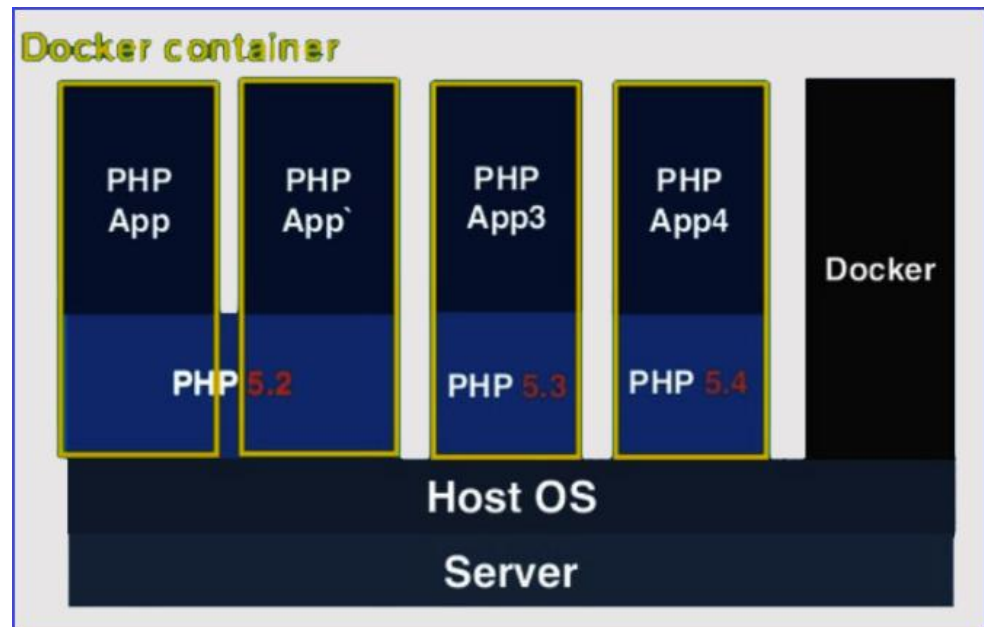
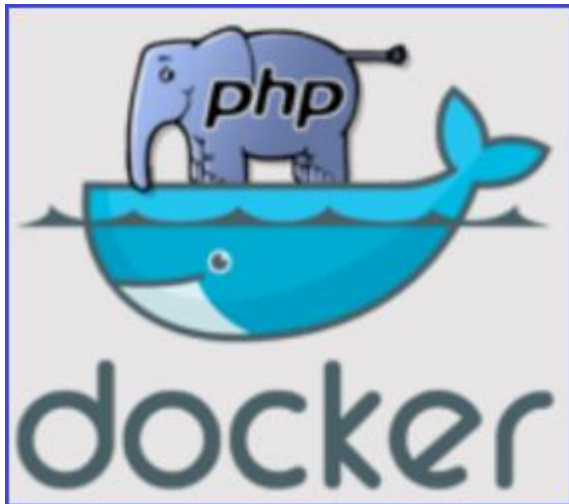
1-Docker란?

- 개발 부분에서 Docker의 필요성
 - “개발, 테스트, 서비스 환경을 하나로 통일”하여 효율적으로 관리할 필요가 있다
 - 개발 환경(개발자 노트북)에서 OS, Middleware를 설치하고 개발한 프로그램을 개발자가 설치 및 운영을 한 후 다음 담당자에게 넘긴다
 - 테스트 환경(회사 서버 컴퓨터)에서도 동일하게 작업을 하여 테스트를 한다
 - 실제 운영 서버에 개발한 프로그램을 설치하여 운영할 때 정상적으로 안되는 경우가 종종 있다



1-Docker란?

- 개발 부분에서 Docker를 사용하여 문제점 해결하기
 - docker는 각각 필요한 software에 호환되는 환경을 구축할 수 있다
 - docker는 각 Project별로 disk 공간을 다르게 사용할 수 있다
 - 필요한 software도 쉽게 구현할 수 있다
 - 필요한 software를 쉽게 배포 및 이식할 수 있다
 - 회사 서버에서 제작한 Docker 이미지를 AWS, Azure 등에서도 동일하게 서비스할 수 있다 (Docker 이식성)



1-Docker란?

- 운영 부분에서 Docker의 필요성
 - 부하가 많이 걸리는 대규모 Web Service를 제공할 때 하나의 물리적 서버로서는 한계가 있다
 - 이를 해결하기 위해 Cloud Infra를 사용할 수도 있다
 - 하지만 간단하게 docker swarm(clustering 기술)을 사용하여 동적으로 대처를 할 수 있다

2-Docker 설치하기

Docker 설치하기

Docker 정보 확인하기

Docker 관리자 추가하기

2-Docker 설치하기

- Docker 설치하기

- CentOS 7에 Docker 설치하기

- **curl -sSL http://get.docker.com | sh**

- Docker 서비스의 자동 시작 구성 여부 확인하기

- **systemctl list-unit-files | grep docker**

- 시스템이 시작할 때 자동으로 Docker 서비스 시작하기

- **systemctl enable docker**

systemctl enable docker --now

- Docker 서비스 시작하기

- **systemctl start docker**

- Docker 서비스 실행 여부 확인하기

- **systemctl status docker**

2-Docker 설치하기

- Docker 정보 확인하기

- 실행된 Docker 서버 및 클라이언트 버전 확인하기

- **docker version**

- Docker 세부 정보 확인하기

- **docker info**

```
[root@centos1 ~]# docker info
Containers: 0
Running: 0
Paused: 0
Stopped: 0
Images: 0
Server Version: 17.06.0-ce
```

- 다운로드 받은 docker image도 없고, 실행된 docker container도 없다

- Docker의 **Root Directory** 내용 확인

- `ls -l /var/lib/docker`

```
[root@centos1 docker]# ls -l /var/lib/docker
total 0
drwx-----. 2 root root  6 Aug 14 21:30 containers
drwx-----. 4 root root 40 Aug 14 21:30 devicemapper
drwx-----. 3 root root 25 Aug 14 21:30 image
drwxr-x---. 3 root root 18 Aug 14 21:30 network
```

```
Total Memory: 1.797GiB
Name: centos1
ID: A2IL:AYX0:IDI2:K3Z2:UTB6:FHV4
Docker Root Dir: /var/lib/docker
Debug Mode (client): false
```

3-Docker 도움말 활용하기

docker help 사용하기

하위 메뉴가 “있는” 명령어 도움말 사용하기

하위 메뉴가 “없는” 명령어 도움말 사용하기

3-Docker 도움말 활용하기

- docker help 사용하기

- docker 명령어를 모두 외울 수가 없으므로 도움말을 잘 활용해야 한다
- **docker help** (=docker --help)
- Management command는 하위 메뉴가 있다

```
Management Commands:
builder          Manage builds
config           Manage Docker configs
container        Manage containers
engine           Manage the docker engine
image            Manage images
network          Manage networks
```

- command는 하위 메뉴가 없다

```
Commands:
attach           Attach local standard input, output,
build            Build an image from a Dockerfile
commit           Create a new image from a container'
cp              Copy files/folders between a contain
create           Create a new container
diff             Inspect changes to files or director
events           Get real time events from the server
exec             Run a command in a running container
```

3-Docker 도움말 활용하기

- 하위 메뉴가 **있는** 명령어 도움말 사용하기
 - **docker image** --help
 - docker image **ls**
 - docker image **ls --help**
 - docker image **ls --no-trunc**
 - **docker image rm** --help
 - docker image rm **alpine**
 - **docker container** --help
 - docker container **ls**
 - docker container **ls -a**
 - **docker container rm** mycon (= **docker rm** mycon)
 - ## alias가 있어서 동일한 명령어가 있을 수 있다
 - **docker system** --help
 - docker system **df**
 - docker system **info** (=docker info)
 - docker system **prune**
 - docker system prune **-a**

3-Docker 도움말 활용하기

- 하위 메뉴가 **없는** 명령어 도움말 사용하기
 - **docker run --help**
 - docker run **alpine**
 - docker run **-it --name myalpine** alpine **/bin/sh**
 - docker run **-d -p 80:80** nginx
 - **docker stats --help**
 - **docker stats -a**
 - **docker run -d -p 80:80** nginx
 - **docker run -it** alpine **/bin/sh**
 - **ping www.google.com**
 - **docker rename --help**
 - docker rename ae2c89eecb95 **mycon**
 - docker ps
 - **docker exec --help**
 - docker exec **-it mycon** **/bin/sh**

4-쉬어 가는 코너

docker create, docker run, docker exec의 차이점

4-쉬어 가는 코너

- docker create, docker run, docker exec의 차이점
 - **docker create**
 - container를 생성만 하고 **start**는 하지 않는다
 - **docker run**
 - container를 생성한 후 **start**까지 하는 것이다
 - **docker exec**
 - 이미 실행 중인 **container**에 명령어를 실행하는 것이다
- 각 명령어의 도움말을 참고한다
 - docker create **--help**
 - docker run **--help**
 - docker exec **--help**

4-쉬어 가는 코너

- docker create, docker run, docker exec의 차이점
 - 예제
 - **docker create** --name busybox-con1 -it **busybox** /bin/sh
 - docker ps (##container가 없음)
 - docker ps -a (##container가 중지됨)
 - **docker start** busybox-con1
 - docker ps (##container가 실행되고 있음)
 - **docker run** --name busybox-con2 -it **busybox** /bin/sh
 - container가 실행되어 /bin/sh에 접속함
 - ping www.google.com -c 3
 - exit
 - docker ps (##busybox-con1만 실행중임)
 - **docker exec** busybox-con1 **mkdir /yslee**
 - **docker exec -it** busybox-con1 **/bin/sh**