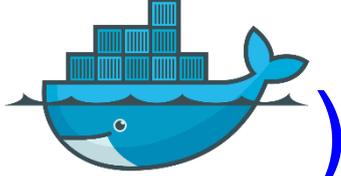


도커(  ) 사용법  
docker

발표자: 권 순 홍([soonhong@pel.smuc.ac.kr](mailto:soonhong@pel.smuc.ac.kr))

지도교수: 이 종 혁([jonghyouk@smu.ac.kr](mailto:jonghyouk@smu.ac.kr))

상명대학교 소프트웨어학과

# 목 차

---

- 개념이론
  - 컨테이너
  - 도커
- 도커의 사용법
  - 실습

# 개념이론

- 가상 머신(VM, Virtual Machine)

- 정의

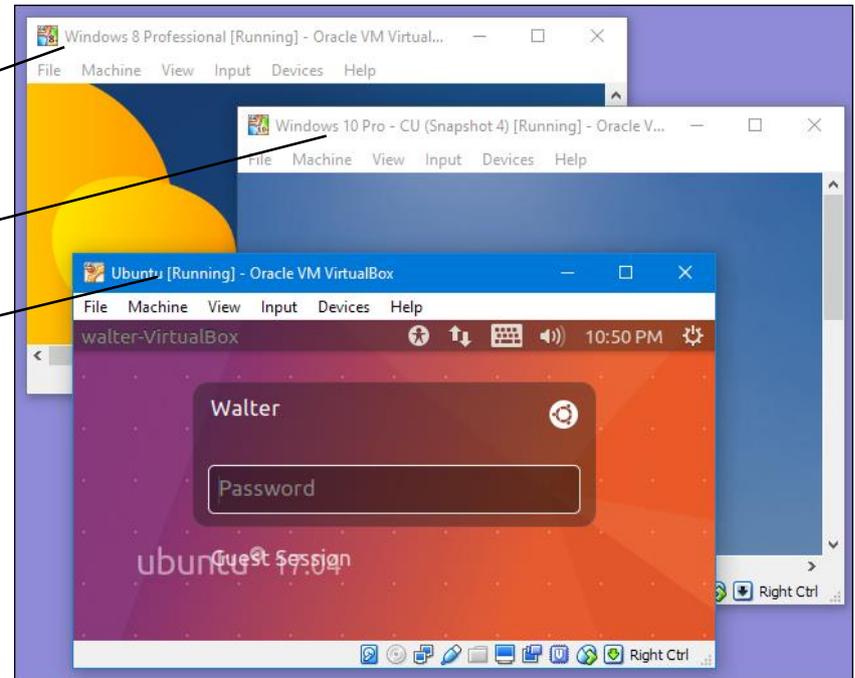
- 컴퓨터의 하드웨어를 소프트웨어적으로 구현하여 운영체제나 응용프로그램을 실행시키는 환경

- 컴퓨터 안의 컴퓨터!

Window 8

Window 10

Ubuntu 17.04



vmware®

<vmware>



<oracle virtual box>



<parallels workstation>

# 개념이론

---

- 가상 머신(VM, Virtual Machine)
  - 그래서 이게.. 뭐가 좋은거죠...?
    - 동시성
      - 한 대의 컴퓨터에서 여러 운영체제 실행 가능
    - 격리성
      - 하나의 가상 머신이 바이러스에 감염되더라도 다른 가상 머신 또는 host pc에 영향을 미치지 않음 (악성코드 분석)
    - 캡슐화
      - 한번 구성된 가상 머신은 복제되어 여러 컴퓨터에 여러 개로 복제되어 사용 가능함
    - 원하는 시점으로 저장 및 복원 가능
      - 바이러스 감염 이전 상태 또는 에러 발생 이전 상태로 돌아갈 수 있음

# 개념이론

- 가상 머신(VM, Virtual Machine)

- 원리

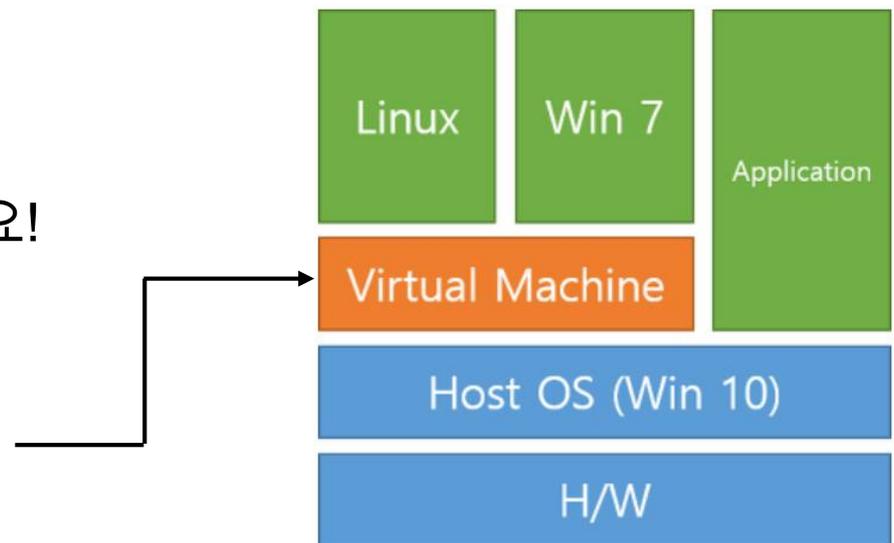
- 호스트 운영체제 (Host OS, Host Operating System)

- 실제 물리적인 하드웨어에 설치된 운영체제

- 게스트 운영체제(Guest OS, Guest Operating System)

- Host OS 위에서 구동되는 운영체제

- 우리가 일반적으로 사용하는 vmware, virtual box가 호스트형 가상화를 사용하는 대표적인 예 예요!

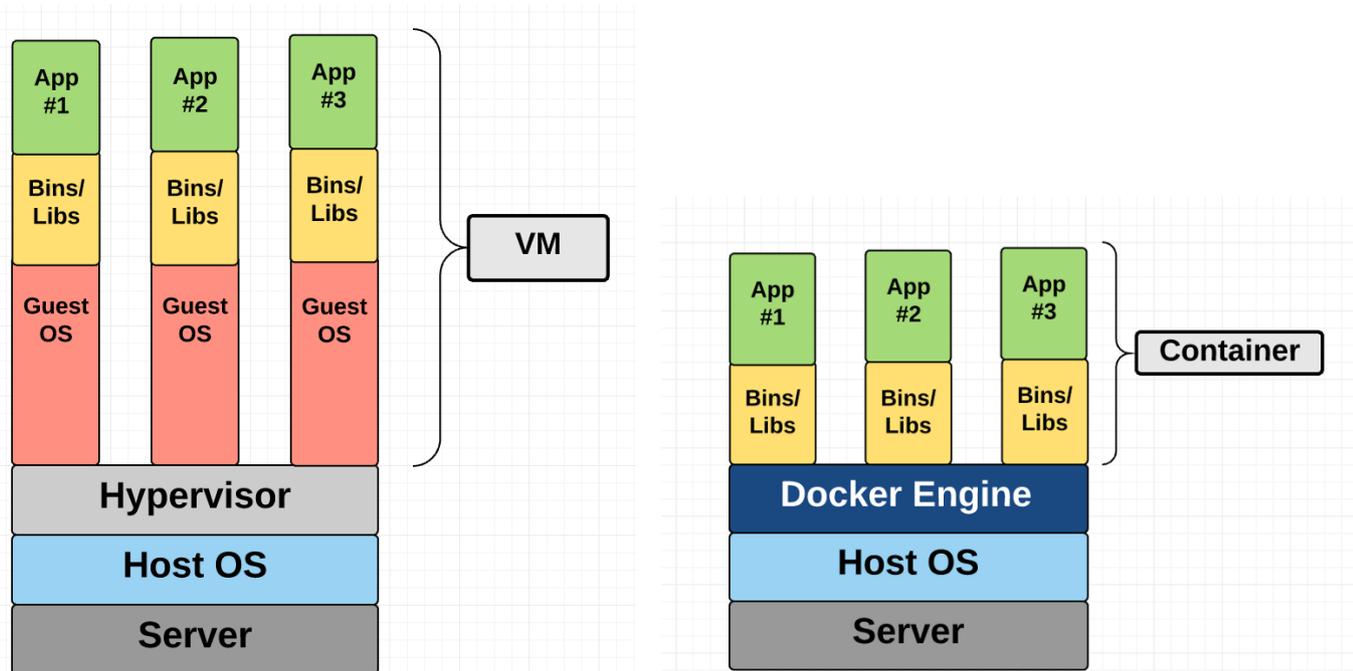


# 개념이론

- 컨테이너(container)

- 정의

- 애플리케이션이 실행되는데 필요한 파일들만 패키징 되어 실행되는 격리 환경
  - 운영체제를 Host OS와 공유하기 때문에 Guest OS 필요 없음!



# 개념이론

---

- 컨테이너(container)

- 장점

- 효율성 및 신속성

- “서비스 수대로 가상머신을 구동하려니 시간도 오래 걸리고 관리도 힘든걸...”

- 개발 및 실행 환경 이전으로 인한 충돌 문제 해결

- “노트북으로 개발 할 때는 잘 되더니 회사 테스트 환경에서는 또 안 되네...”

- 마이크로 서비스화 실현

- “애플리케이션 크기가 너무 커서 배포가 너무 오래 걸리네...”

# 개념이론

- 컨테이너(container)

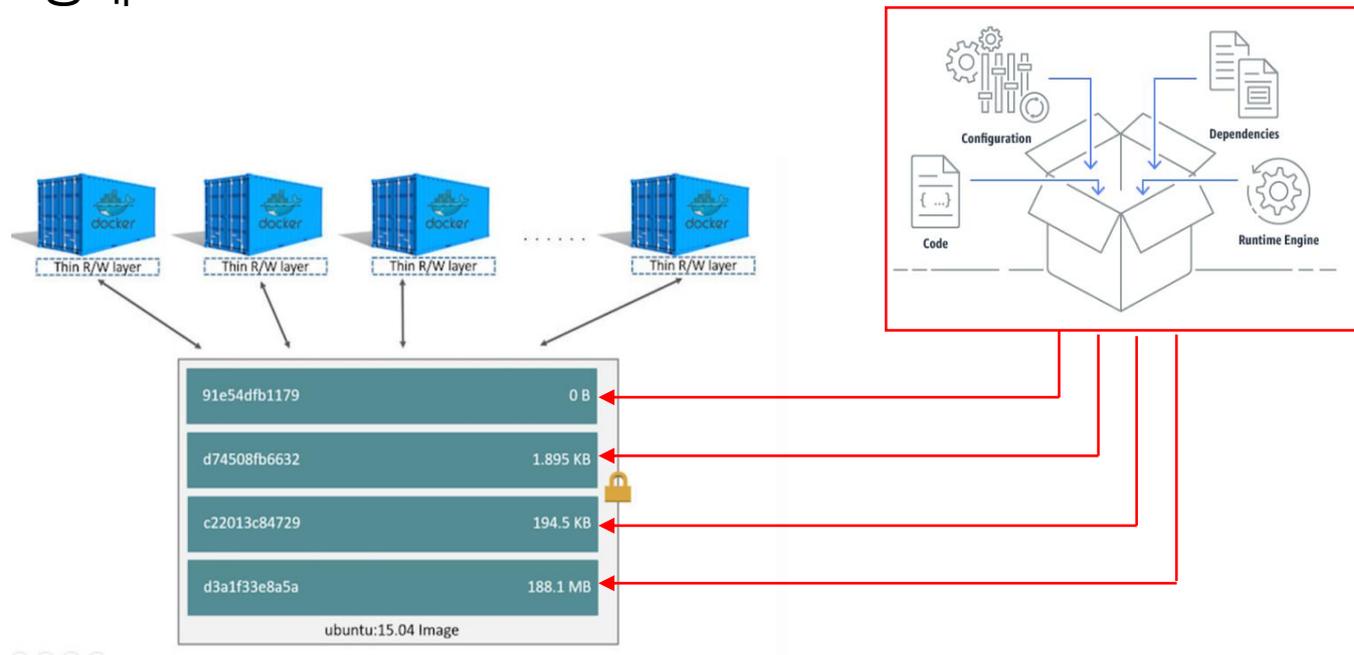
- 원리

- 이미지 파일(image file)이 실행됨

- 하나의 이미지 파일은 여러 개의 이미지 레이어로 구성

- e.g., Ubuntu 15.04 이미지 파일은 4개의 이미지 파일들로 구성

- 서비스 운영에 필요한 프로그램, 소스코드, 컴파일 된 실행파일을 묶은 패키지 형태



# 개념이론

---

- 도커(docker)

- 정의

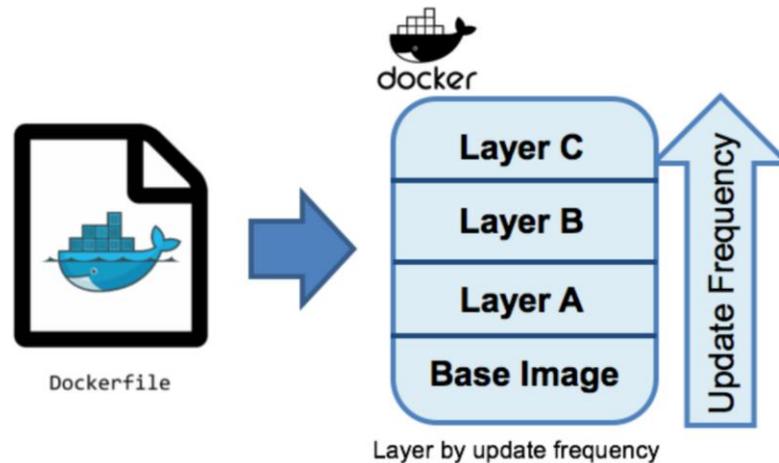
- 소프트웨어 빌드 및 실행을 위한 리눅스 컨테이너 플랫폼
  - 소프트웨어 설치, 실행, 삭제 과정이 매우 단순화됨
  - 환경(커널 버전, 운영체제 버전)에 구애 받지 않고 개발할 수 있는 플랫폼 제공
    - 아주아주아주아주x100 핫하다!!
- GitHub와 같은 Docker hub를 통해 이미지 공유 가능
  - <https://hub.docker.com/>

# 개념이론

- 도커(docker)

- 장점

- 여러 개의 이미지 파일로 묶여진 도커 이미지(docker image)는 window든 mac이든 linux든 똑같이 실행됨



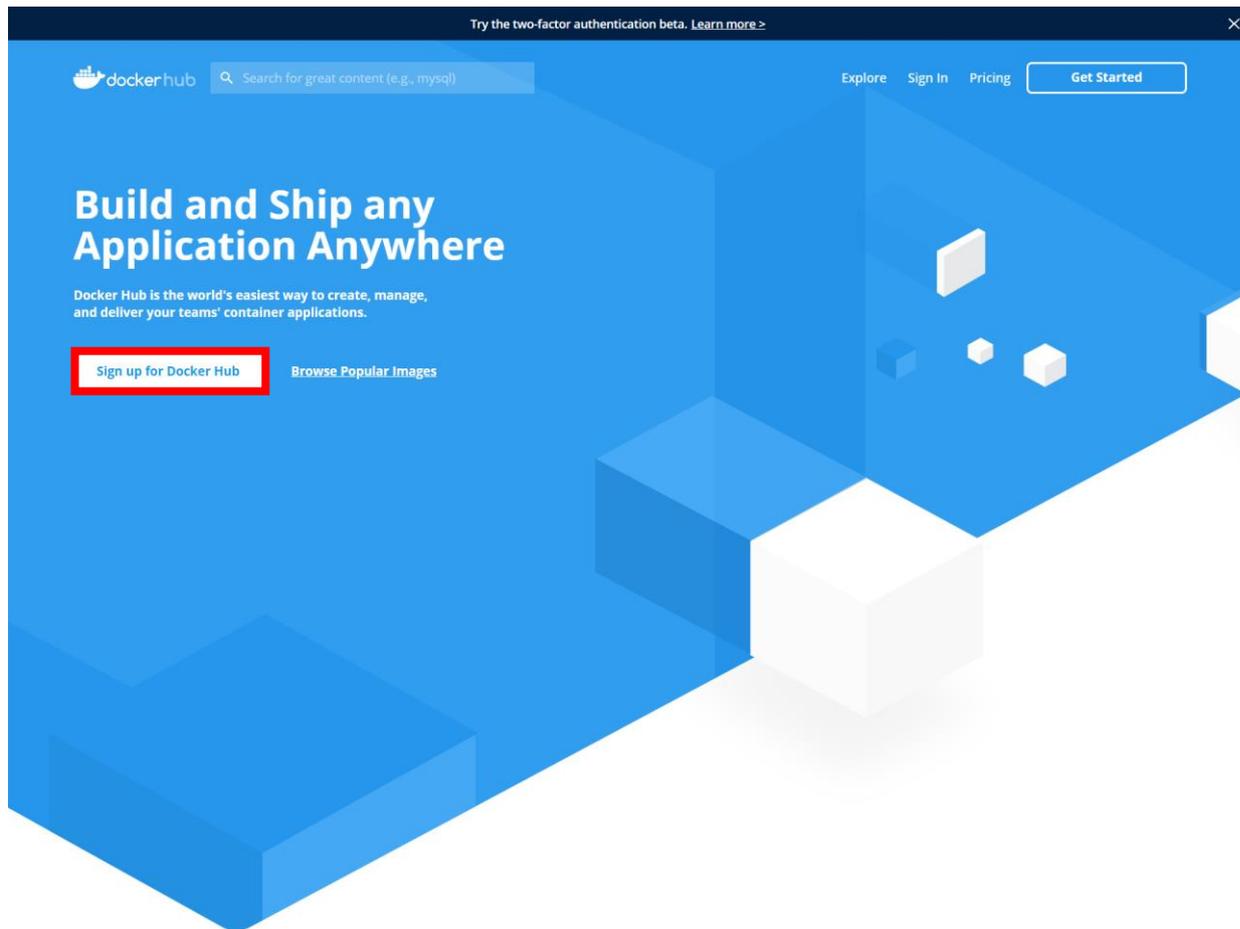
- 이미지 레이어 단위로 변경 내용이 추가 및 삭제되어 관리가 수월함

- 고래가 귀여움....  

# Docker 사용법

- 실습

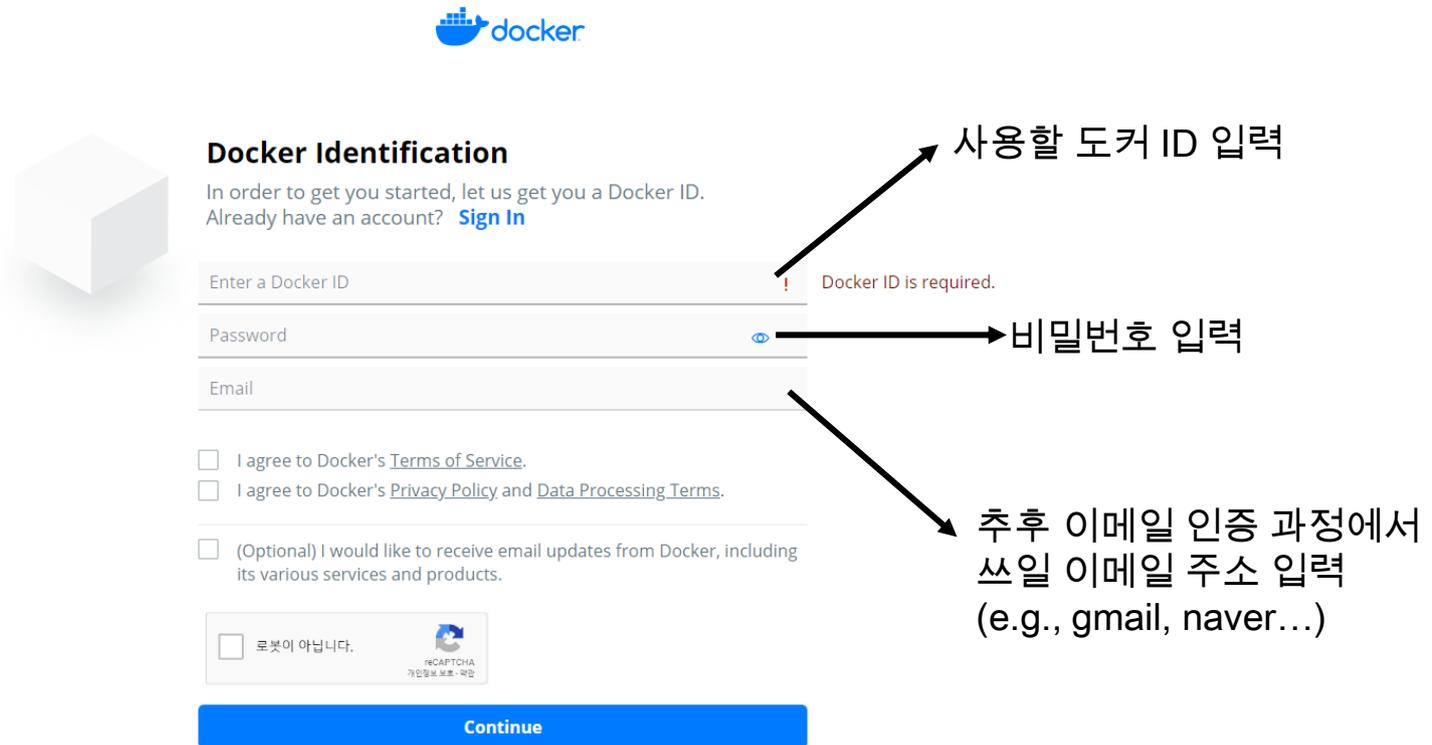
- Docker Hub 가입 절차(1/5)



# Docker 사용법

- 실습

- Docker Hub 가입 절차(2/5)



The screenshot shows the Docker Identification registration page. It features a form with three input fields: 'Enter a Docker ID', 'Password', and 'Email'. Below the form are three checkboxes for terms of service and an optional email update preference. A reCAPTCHA widget is at the bottom left, and a blue 'Continue' button is at the bottom center. Three arrows point from Korean text annotations to the form fields: one to the Docker ID field, one to the Password field, and one to the Email field.

**Docker Identification**  
In order to get you started, let us get you a Docker ID.  
Already have an account? [Sign In](#)

Enter a Docker ID ! Docker ID is required.

Password 👁

Email

I agree to Docker's [Terms of Service](#).

I agree to Docker's [Privacy Policy](#) and [Data Processing Terms](#).

(Optional) I would like to receive email updates from Docker, including its various services and products.

로봇이 아닙니다.  reCAPTCHA  
개인정보 보호 · 학습

**Continue**

사용할 도커 ID 입력

비밀번호 입력

추후 이메일 인증 과정에서  
쓰일 이메일 주소 입력  
(e.g., gmail, naver...)

# Docker 사용법

- 실습

- Docker Hub 가입 절차(3/5)



**Thank You!**  
**Your account is created.**

We want you to have the best Docker experience.

Complete this information, and help us help you personalize your Docker experience.

## Complete your profile

Tell us more about yourself.

\*First Name

이름 입력

\*Last Name

성 입력

\*Company

Sangmyung University 입력

\*Country

국가 입력

## Tailor your experience

Tell us about your role and where you are in your container development journey.

\*Your Role

역할 입력

(e.g., Developer or Developer Manager)

\*Docker Experience

도커 사용 여부 입력

Continue

In order to continue, all fields are required.

! First name is required.

# Docker 사용법

- 실습

- Docker Hub 가입 절차(4/5)



The image shows two screenshots from the Docker Hub registration process. The left screenshot displays the instruction: "Please Check Your Inbox" with the text "You will need to verify your email address for security purposes. Email verification ensures the protection of your account." Below this is an envelope icon and a "Return to Home Page" link. The right screenshot shows the "Docker Identification" form. It includes fields for "Enter a Docker ID", "Password", and "Email". The "Email" field is highlighted with a red box. Below the form are checkboxes for agreeing to terms of service and privacy policy, and a CAPTCHA. A "Continue" button is at the bottom. A black arrow points from the text "앞서 작성한 이메일을 통해 인증 수행" (Perform authentication through the email address you wrote earlier) to the "Email" field in the registration form.

**Please Check Your Inbox**  
You will need to verify your email address for security purposes.  
Email verification ensures the protection of your account.

**Docker Identification**  
In order to get you started, let us get you a Docker ID.  
Already have an account? [Sign In](#)

Enter a Docker ID Docker ID is required.

Password

Email

I agree to Docker's [Terms of Service](#).

I agree to Docker's [Privacy Policy and Data Processing Terms](#).

(Optional) I would like to receive email updates from Docker, including its various services and products.

로봇이 아닙니다. 

[Continue](#)

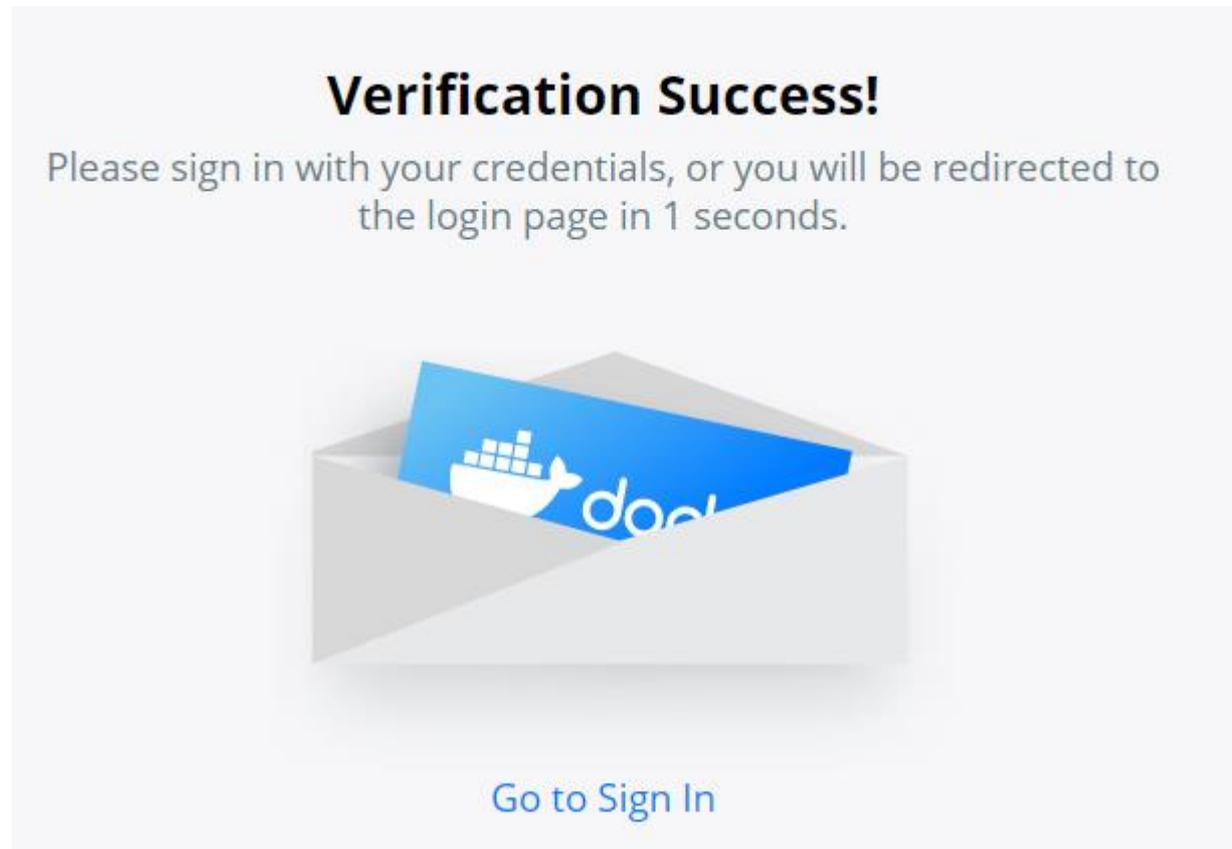
앞서 작성한 이메일을 통해 인증 수행

[Return to Home Page](#)

# Docker 사용법

---

- 실습
  - Docker Hub 가입 절차(5/5)



# Docker 사용법

---

- 실습

1. 패키지 설치

- Docker를 사용하기 위해 필요한 패키지를 다운로드
  - e.g., golang, git, curl

```
tester@ubuntu:~$ sudo apt install golang git curl -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
golang is already the newest version (2:1.6-1ubuntu4).
git is already the newest version (1:2.7.4-0ubuntu1.6).
The following additional packages will be installed:
  libcurl3-gnutls
The following packages will be upgraded:
  curl libcurl3-gnutls
```

# Docker 사용법

## • 실습

### 2. 도커 설치

- 자동 설치 스크립트를 통해 다운로드 수행
  - `$ curl -f https://get.docker.com/ | sudo sh`

```
tester@ubuntu:~$ curl -f https://get.docker.com/ | sudo sh
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           % Dload  % Upload   Total   Spent    Left   Speed
100 13216  100 13216    0     0 39473      0 --:--:-- --:--:-- --:--:-- 39450
# Executing docker install script, commit: f45d7c11389849ff46a6b4d94e0dd1ffe8ca32c1
+ sh -c apt-get update -qq >/dev/null
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq apt-transport-https ca-certificates curl >/dev/null
+ sh -c curl -fsSL "https://download.docker.com/linux/ubuntu/gpg" | apt-key add -qq - >/dev/null
+ sh -c echo "deb [arch=amd64] https://download.docker.com/linux/ubuntu xenial stable" > /etc/apt/sources.list.d/docker.list
+ sh -c apt-get update -qq >/dev/null
+ [ -n ]
+ sh -c apt-get install -y -qq --no-install-recommends docker-ce >/dev/null
+ sh -c docker version
Client: Docker Engine - Community
 Version:           19.03.5
 API version:       1.40
 Go version:        go1.12.12
 Git commit:        633a0ea838
 Built:             Wed Nov 13 07:50:12 2019
 OS/Arch:           linux/amd64
 Experimental:      false

Server: Docker Engine - Community
 Engine:
  Version:          19.03.5
  API version:      1.40 (minimum version 1.12)
  Go version:       go1.12.12
  Git commit:       633a0ea838
  Built:           Wed Nov 13 07:48:43 2019
  OS/Arch:         linux/amd64
  Experimental:    false
 containerd:
  Version:         1.2.10
  GitCommit:       b34a5c8af56e510852c35414db4c1f4fa6172339
 runc:
  Version:        1.0.0-rc8+dev
  GitCommit:      3e425f80a8c931f88e6d94a8c831b9d5aa481657
 docker-init:
  Version:        0.18.0
  GitCommit:      fec3683
```

# Docker 사용법

---

- 실습

## 2. 도커 설치

- 도커는 기본적으로 root권한을 필요로 함
- 사용자를 docker 그룹에 추가
  - Root가 아닌 사용자가 sudo없이 도커를 사용하기 위함
  - `$ sudo usermod -aG docker $USER`

```
tester@ubuntu:~$ sudo usermod -aG docker $USER
```

# Docker 사용법

- 실습

## 2. 도커 설치

- 도커가 성공적으로 설치되었는지 확인

```
tester@ubuntu:~$ docker version
```

```
Client: Docker Engine - Community
Version: 19.03.5
API version: 1.40
Go version: go1.12.12
Git commit: 633a0ea838
Built: Wed Nov 13 07:50:12 2019
OS/Arch: linux/amd64
Experimental: false
```

```
Server: Docker Engine - Community
Engine:
Version: 19.03.5
API version: 1.40 (minimum version 1.12)
Go version: go1.12.12
Git commit: 633a0ea838
Built: Wed Nov 13 07:48:43 2019
OS/Arch: linux/amd64
Experimental: false
containerd:
Version: 1.2.10
GitCommit: b34a5c8af56e510852c35414db4c1f4fa6172339
runc:
Version: 1.0.0-rc8+dev
GitCommit: 3e425f80a8c931f88e6d94a8c831b9d5aa481657
docker-init:
Version: 0.18.0
GitCommit: fec3683
```

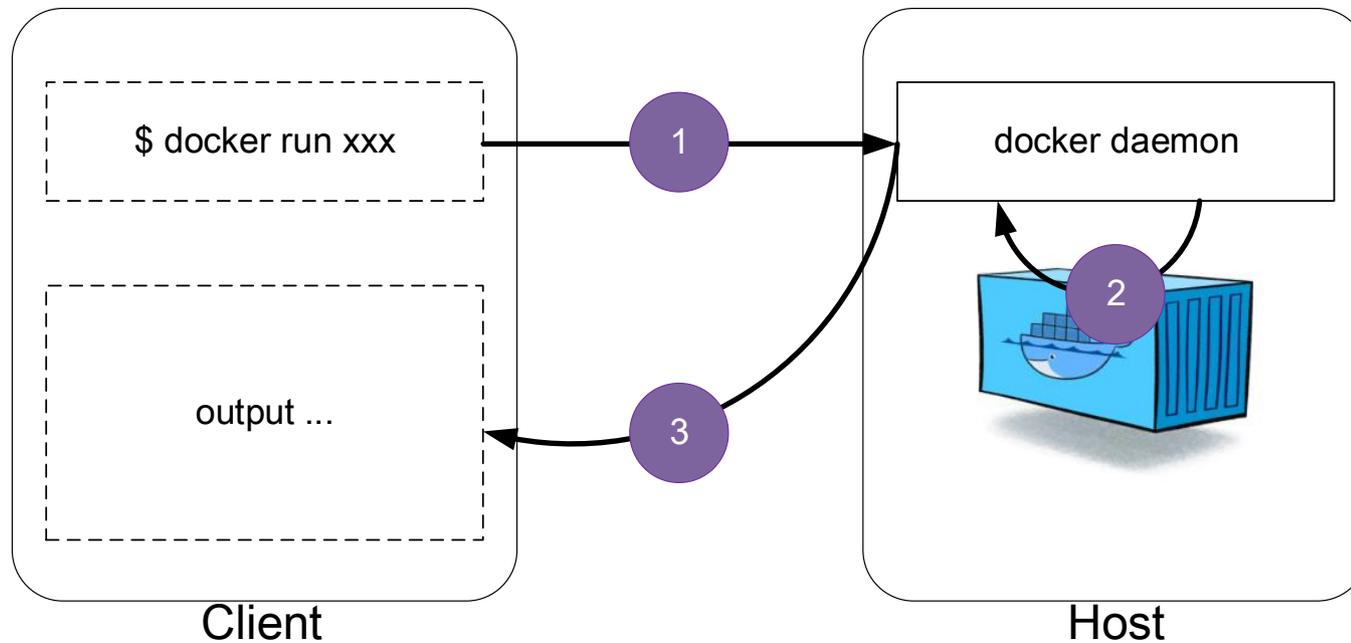
클라이언트와 서버가 모두 출력  
되어야함

# Docker 사용법

- 실습

- 도커 클라이언트와 서버

- 도커는 하나의 실행파일이지만 클라이언트와 서버 역할을 각각 수행



# Docker 명령어

## • 대표적인 명령어

	명령어	설명	사용법
1	run	이미지 실행	docker run [image name]
2	push	이미지 업로드	docker push [image name]
3	pull	이미지 다운로드	docker pull [image name]
4	rmi	이미지 삭제	docker rmi [image name]
5	rm	컨테이너 삭제	docker rm [container ID]
6	build	Dockerfile로 이미지 생성	docker build [dockerfile name]
7	commit	현재 컨테이너를 이미지로 저장	docker commit [container name]
8	ps	실행 중인 컨테이너 리스트 출력	docker ps
9	images	로컬에 저장된 이미지 출력	docker images
10	start	컨테이너 시작	docker start [container name]
11	stop	컨테이너 중지	docker stop [container name]
12	attach	시작한 컨테이너 접속	docker attach [container name]
13	inspect	이미지 정보 출력	docker inspect [image name]

# Docker 사용법

- 실습

- 도커 이미지 검색

- 다운로드 받고자 하는 도커 이미지에 대해 검색을 수행
  - `$ docker search <image:tag>`

```
tester@ubuntu:~$ docker search ubuntu
```

NAME	DESCRIPTION	STARS	OFFICIAL	AUTOMATED
ubuntu	Ubuntu is a Debian-based Linux operating sys...	10214	[OK]	
dorowu/ubuntu-desktop-lxde-vnc	Docker image to provide HTML5 VNC interface ...	366		[OK]
rastasheep/ubuntu-sshd	Dockerized SSH service, built on top of offi...	236		[OK]
consol/ubuntu-xfce-vnc	Ubuntu container with "headless" VNC session...	197		[OK]
ubuntu-upstart	Upstart is an event-based replacement for th...	101	[OK]	
ansible/ubuntu14.04-ansible	Ubuntu 14.04 LTS with ansible	98		[OK]
1and1internet/ubuntu-16-nginx-php-phpmyadmin-mysql-5	ubuntu-16-nginx-php-phpmyadmin-mysql-5	50		[OK]
ubuntu-debootstrap	debootstrap --variant=minbase --components=m...	40	[OK]	
nuagebec/ubuntu	Simple always updated Ubuntu docker images w...	24		[OK]
i386/ubuntu	Ubuntu is a Debian-based Linux operating sys...	18		
1and1internet/ubuntu-16-apache-php-5.6	ubuntu-16-apache-php-5.6	14		[OK]
1and1internet/ubuntu-16-apache-php-7.0	ubuntu-16-apache-php-7.0	13		[OK]
eclipse/ubuntu_jdk8	Ubuntu, JDK8, Maven 3, git, curl, nmap, mc, ...	12		[OK]
1and1internet/ubuntu-16-nginx-php-phpmyadmin-mariadb-10	ubuntu-16-nginx-php-phpmyadmin-mariadb-10	11		[OK]
1and1internet/ubuntu-16-nginx-php-5.6	ubuntu-16-nginx-php-5.6	8		[OK]
1and1internet/ubuntu-16-nginx-php-5.6-wordpress-4	ubuntu-16-nginx-php-5.6-wordpress-4	7		[OK]
1and1internet/ubuntu-16-apache-php-7.1	ubuntu-16-apache-php-7.1	6		[OK]
darksheer/ubuntu	Base Ubuntu Image -- Updated hourly	5		[OK]
1and1internet/ubuntu-16-nginx-php-7.0	ubuntu-16-nginx-php-7.0	4		[OK]
pivotaldata/ubuntu	A quick freshening-up of the base Ubuntu doc...	2		
pivotaldata/ubuntu-gpdb-dev	Ubuntu images for GPDB development	1		
1and1internet/ubuntu-16-php-7.1	ubuntu-16-php-7.1	1		[OK]
smartentry/ubuntu	ubuntu with smartentry	1		[OK]
1and1internet/ubuntu-16-sshd	ubuntu-16-sshd	1		[OK]
1and1internet/ubuntu-16-rspec	ubuntu-16-rspec	0		[OK]

# Docker 사용법

---

- 실습

- 도커 이미지 다운로드

- \$ docker search 명령을 통해 확인한 도커 이미지에 대해 다운로드 수행
  - \$ docker pull <image:tag>

```
tester@ubuntu:~$ docker pull ubuntu:latest
latest: Pulling from library/ubuntu
7ddbc47eeb70: Pull complete
c1bbdc448b72: Pull complete
8c3b70e39044: Pull complete
45d437916d57: Pull complete
Digest: sha256:6e9f67fa63b0323e9a1e587fd71c561ba48a034504fb804fd26fd8800039835d
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

# Docker 사용법

---

- 실습

- 다운로드 받은 도커 이미지 확인
  - \$ docker images 커맨드를 통해 확인 가능

```
tester@ubuntu:~$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	latest	775349758637	4 weeks ago	64.2MB

# Docker 사용법

- 실습

- 다운로드 받은 도커 이미지 확인

- \$ docker images 커맨드를 통해 확인 가능

```
tester@ubuntu:~$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	latest	775349758637	4 weeks ago	64.2MB

- 도커 이미지 삭제

- \$ docker rmi <image> 커맨트를 통해 삭제 가능

```
tester@ubuntu:~$ docker rmi ubuntu:latest
```

```
Untagged: ubuntu:latest
Untagged: ubuntu@sha256:6e9f67fa63b0323e9a1e587fd71c561ba48a034504fb804fd26fd8800039835d
Deleted: sha256:775349758637aff77bf85e2ff0597e86e3e859183ef0baba8b3e8fc8d3cba51c
Deleted: sha256:4fc26b0b0c6903db3b4fe96856034a1bd9411ed963a96c1bc8f03f18ee92ac2a
Deleted: sha256:b53837dafdd21f67e607ae642ce49d326b0c30b39734b6710c682a50a9f932bf
Deleted: sha256:565879c6effe6a013e0b2e492f182b40049f1c083fc582ef61e49a98dca23f7e
Deleted: sha256:cc967c529ced563b7746b663d98248bc571afdb3c012019d7f54d6c092793b8b
```

# Docker 사용법

- 실습

- 다운로드 받은 이미지를 통해 컨테이너 생성 및 실행
  - `$ docker run -i -t --name <container_name> -p 80:80 <image:tag>`

```
tester@ubuntu:~$ docker run -i -t --name test-ubuntu -p 80:80 ubuntu:latest
root@c0d3e8187623:/#
```

Option	Description
<code>--name</code>	컨테이너 이름 지정
<code>-i</code>	Interactive, 컨테이너의 입력 및 출력등을 상호작용
<code>-t</code>	Pseudo-tty로 터미널과 같은 환경 사용하겠다는 옵션
<code>-p</code>	Host의 포트를 컨테이너의 포트로 오픈 <code>-p &lt;host-port&gt;:&lt;container-port&gt;</code> Host의 포트를 지정하지 않으면 임의의 포트로 할당

# Docker 사용법

- 실습

- 실행중인 컨테이너 목록 확인

- \$ docker ps 명령을 통해 실행중인 컨테이너 목록 확인 가능

```
tester@ubuntu:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS          NAMES
c0d3e8187623   ubuntu:latest  "/bin/bash"            11 minutes ago  Up 11 minutes  0.0.0.0:80->80/tcp  test-ubuntu
```

- 생성된 모든 컨테이너 목록 확인

- \$ docker ps -a 명령을 통해 생성된 모든 컨테이너 목록 확인 가능

```
tester@ubuntu:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS          NAMES
5fac7a6687cd   centos:latest  "/bin/bash"            About a minute ago  Exited (0) About a minute ago
c0d3e8187623   ubuntu:latest  "/bin/bash"            12 minutes ago    Up 12 minutes   0.0.0.0:80->80/tcp  test-ubuntu
test-centos
```

# Docker 사용법

- 실습

- 생성한 컨테이너를 구동 및 종료

- 구동

- \$ docker start <container\_name> 명령을 통해 컨테이너 구동 가능

```
tester@ubuntu:~$ docker start test-ubuntu
```

```
test-ubuntu
```

```
tester@ubuntu:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
c0d3e8187623	ubuntu:latest	"/bin/bash"	16 minutes ago	Up 5 seconds	0.0.0.0:80->80/tcp	test-ubuntu

- 종료

- \$ docker stop <container\_name> 명령을 통해 컨테이너 종료 가능

```
tester@ubuntu:~$ docker stop test-ubuntu
```

```
test-ubuntu
```

```
tester@ubuntu:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

# Docker 사용법

## • 실습

- 생성한 컨테이너에 대해 추후 내부 접근하는 방법
  - \$ docker attach -i -t <container name>으로 접근 가능
    - \$ docker run을 한번 수행하면 이후에는 docker attach를 통해 컨테이너에 접근해야함

```
tester@ubuntu:~$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
99dcfe0063f0       tnsghd1004/webtest:testing  "/bin/bash"        About an hour ago   Exited (0) 50 minutes ago                web-test
tester@ubuntu:~$ docker run -i -t --name web-test -p 80:80 tnsghd1004/webtest:testing
docker: Error response from daemon: Conflict. The container name "/web-test" is already in use by container "99dcfe0063f0d74678e978e80b347fcf85377cf86f6522f22d3298b474f5b881". You have to remove (or rename) that container to be able to reuse that name.
```



```
tester@ubuntu:~$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
99dcfe0063f0       tnsghd1004/webtest:testing  "/bin/bash"        About an hour ago   Exited (0) 6 seconds ago                web-test
tester@ubuntu:~$ docker start web-test
web-test
tester@ubuntu:~$ docker attach web-test
root@99dcfe0063f0:/#
```

# Docker 사용법

---

- 실습

- 컨테이너 내부에서 명령어 실행
  - Bash shell모드로 컨테이너 내부로 들어갈 수 있음
  - `$ docker exec -it test-ubuntu bash`
    - `$ cat /etc/issue` 명령을 통해 OS 정보 확인 가능

```
tester@ubuntu:~$ docker exec -it test-ubuntu bash
root@c0d3e8187623:/# cat /etc/issue
Ubuntu 18.04.3 LTS \n \l
```

# Docker 사용법

---

- 실습

- 도커 이미지 생성하기

- 우분투 이미지를 다운로드하여 변경사항을 만들고, 이미지 생성

- 변경사항

- 레포지토리 업데이트
- 웹서버 띄우기

# Docker 사용법

- 실습
  - 우분투 컨테이너 레포지토리 업데이트
    - # apt-get update를 통해 레포지토리 업데이트 가능

```
root@cd3e8187623:/# apt-get update
Get:1 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic InRelease [242 kB]
Get:3 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [16.8 kB]
Get:4 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [6508 B]
Get:5 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [789 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:7 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [739 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:9 http://archive.ubuntu.com/ubuntu bionic/multiverse amd64 Packages [186 kB]
Get:10 http://archive.ubuntu.com/ubuntu bionic/main amd64 Packages [1344 kB]
Get:11 http://archive.ubuntu.com/ubuntu bionic/universe amd64 Packages [11.3 MB]
Get:12 http://archive.ubuntu.com/ubuntu bionic/restricted amd64 Packages [13.5 kB]
Get:13 http://archive.ubuntu.com/ubuntu bionic-updates/universe amd64 Packages [1316 kB]
Get:14 http://archive.ubuntu.com/ubuntu bionic-updates/restricted amd64 Packages [29.9 kB]
Get:15 http://archive.ubuntu.com/ubuntu bionic-updates/multiverse amd64 Packages [9549 B]
Get:16 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 Packages [1035 kB]
Get:17 http://archive.ubuntu.com/ubuntu bionic-backports/main amd64 Packages [2496 B]
Get:18 http://archive.ubuntu.com/ubuntu bionic-backports/universe amd64 Packages [4235 B]
Fetched 17.3 MB in 7s (2564 kB/s)
Reading package lists... Done
```

# Docker 사용법

- 실습

- 웹서버 구축을 위한 패키지 다운로드

1. vim 패키지 다운로드

- # apt-get install vim

```
root@c0d3e8187623:/# apt-get install vim
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  file libexpat1 libgpm2 libmagic-mgc libmagic1 libmpdec2 libpython3.6 |
  xxd xz-utils
Suggested packages:
  gpm readline-doc ctags vim-doc vim-scripts
The following NEW packages will be installed:
  file libexpat1 libgpm2 libmagic-mgc libmagic1 libmpdec2 libpython3.6 |
  vim-runtime xxd xz-utils
0 upgraded, 19 newly installed, 0 to remove and 2 not upgraded.
Need to get 12.9 MB of archives.
After this operation, 61.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

# Docker 사용법

- 실습

- 웹서버 구축을 위한 패키지 다운로드

2. Apache 패키지 다운로드

- # apt-get install apache2

```
root@c0d3e8187623:/# apt-get install apache2
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dev
  libheimbase1-heimdal libheimntlm0-heimdal libhx509-5-heimdal libicu52
  libsasl2-modules libsasl2-modules-db libwind0-heimdal libxml2 netbase
Suggested packages:
  www-browser apache2-doc apache2-suexec-pristine | apache2-suexec-common
  libsasl2-modules-sql ca-certificates perl-doc libterm-readline-gnu
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
  libheimbase1-heimdal libheimntlm0-heimdal libhx509-5-heimdal libicu52
  libsasl2-modules libsasl2-modules-db libwind0-heimdal libxml2 netbase
0 upgraded, 34 newly installed, 0 to remove and 2 not upgraded.
Need to get 18.8 MB of archives.
After this operation, 87.8 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

# Docker 사용법

- 실습

- 웹서버 구축을 위한 패키지 다운로드

3. php 패키지 다운로드

- # apt-get install libapache2-mod-php

```
root@c0d3e8187623:/# apt-get install libapache2-mod-php
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libapache2-mod-php7.2 libargon2-0 libedit2 libsodium23 php-common |
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php libapache2-mod-php7.2 libargon2-0 libedit2 libs
0 upgraded, 14 newly installed, 0 to remove and 2 not upgraded.
Need to get 4391 kB of archives.
After this operation, 21.3 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

# Docker 사용법

- 실습

- 웹서버 구축을 위한 패키지 다운로드

4. mysql 패키지 다운로드

- # apt-get install mysql-server mysql-client

```
root@c0d3e8187623:/# apt-get install mysql-server mysql-client
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libencode-locale-perl lib
  libio-html-perl liblwp-mediatypes-perl libnuma1 libtimedate-perl
Suggested packages:
  libdata-dump-perl libipc-sharedcache-perl libwww-perl mailx tinc
The following NEW packages will be installed:
  libaio1 libcgi-fast-perl libcgi-pm-perl libencode-locale-perl lib
  libio-html-perl liblwp-mediatypes-perl libnuma1 libtimedate-perl
  mysql-server-core-5.7
0 upgraded, 24 newly installed, 0 to remove and 2 not upgraded.
Need to get 19.8 MB of archives.
After this operation, 157 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

# Docker 사용법

- 실습

- 웹서버 구축을 위한 패키지 다운로드

5. phpmyadmin 패키지 다운로드

- # apt-get install phpmyadmin

```
root@c0d3e8187623:/# apt-get install phpmyadmin
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  ca-certificates dbconfig-common dbconfig-mysql fontconfig-config fc
  libjpeg-turbo8 libjpeg8 libjs-jquery libjs-sphinxdoc libjs-undersco
  libxcb1 libxdmcp6 libxpm4 libxslt1.1 libzip4 multiarch-support php
  php7.2-curl php7.2-gd php7.2-mbstring php7.2-mysql php7.2-xml php7.
Suggested packages:
  libgd-tools krb5-doc krb5-user php-libsodium php-mcrypt php-gmp php
The following NEW packages will be installed:
  ca-certificates dbconfig-common dbconfig-mysql fontconfig-config fc
  libjpeg-turbo8 libjpeg8 libjs-jquery libjs-sphinxdoc libjs-undersco
  libxcb1 libxdmcp6 libxpm4 libxslt1.1 libzip4 multiarch-support php
  php7.2-curl php7.2-gd php7.2-mbstring php7.2-mysql php7.2-xml php7.
0 upgraded, 59 newly installed, 0 to remove and 2 not upgraded.
Need to get 18.3 MB of archives.
After this operation, 69.7 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

# Docker 사용법

- 실습

- 웹서버 구동을 위한 설정

- `/etc/apache2/apache2.conf` 코드 추가를 통한 phpmyadmin 구동

```
#Enable phpmyadmin  
include /etc/phpmyadmin/apache.conf
```

- 웹서버 구동

- `# /etc/init.d/apache2 restart` 명령을 통해 웹서버 구동 가능

```
root@c0d3e8187623:/# /etc/init.d/apache2 restart  
* Restarting Apache httpd web server apache2  
[Sun Dec 01 21:35:26.544734 2019] [alias:warn] [pid 14561] AH00671: The Alias directive in /etc/phpmyadmin/apache.conf at line 3 will probably never match because it overlaps an earlier Alias.  
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
```

[ OK ]

# Docker 사용법

---

- 실습
  - 샘플 웹 코드 작성
    - /var/www/html 디렉토리에 index.php 코드 작성

```
1 <?php
2     echo "Success Web Establishment";
3 ?>
```

# Docker 사용법

- 실습

- 웹서버가 성공적으로 동작하는지 확인
  - Firefox에 IP주소/index.php를 입력하여 확인 가능
    - IP 주소의 경우, apache2 restart 시, 출력되는 것으로 이용

```
root@c0d3e8187623:/# /etc/init.d/apache2 restart
* Restarting Apache httpd web server apache2
[Sun Dec 01 21:35:26.544734 2019] [alias:warn] [pid 14561] AH00671: The Alias directive in /etc/phpmyadmin/
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2.
```



# Docker 사용법

---

- 실습

- 웹 도커 컨테이너를 도커 이미지로 저장

- # docker commit -a "<editor name>" -m "<message>"  
<containername> <image name>:<version>

- e.g., docker commit -a "Soonhong Kwon  
<soonhong@pel.smuc.ac.kr>" -m "webtest" test testingweb:latest

```
tester@ubuntu:~$ docker commit -a "Soonhong Kwon <soonhong@pel.smuc.ac.kr>" -m "webtest" test testingweb:latest  
sha256:d8cad11259f6500962099563215b59728e0a1302bce66e87e4443995d164221d
```

# Docker 사용법

- 실습
  - 생성한 이미지 확인
    - `$ docker images` 명령으로 확인

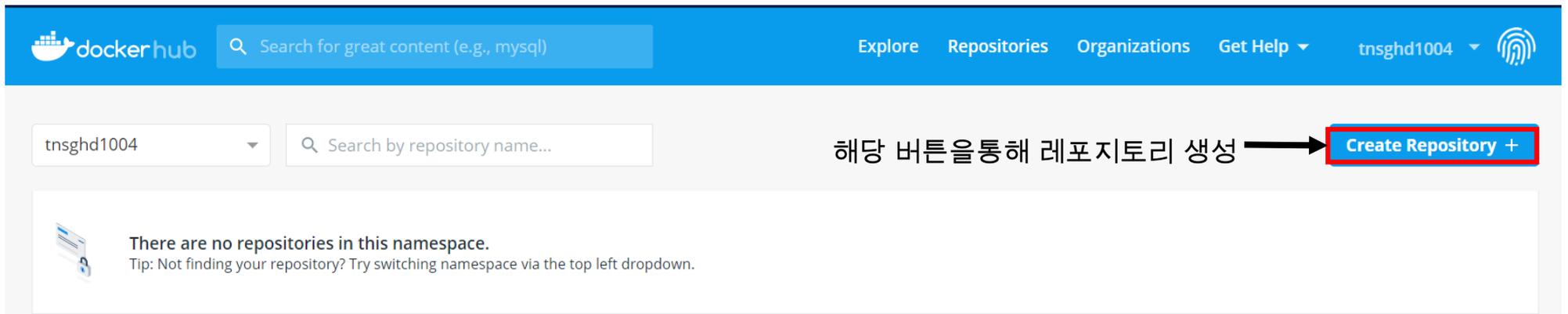
```
tester@ubuntu:~$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
testingweb	latest	d8cad11259f6	About a minute ago	600MB
webtesting	latest	2c542da77139	18 minutes ago	600MB
ubuntu	latest	775349758637	4 weeks ago	64.2MB
centos	latest	0f3e07c0138f	2 months ago	220MB

# Docker 사용법

- 실습

- Docker hub 레포지토리 생성(1/2)



The screenshot shows the Docker Hub user interface. At the top, there is a blue navigation bar with the Docker Hub logo, a search bar, and links for 'Explore', 'Repositories', 'Organizations', and 'Get Help'. The user's profile 'tnsghd1004' is visible in the top right. Below the navigation bar, there is a search bar for repository names and a dropdown menu showing the current namespace 'tnsghd1004'. A red box highlights the 'Create Repository +' button, with an arrow pointing to it from the Korean text '해당 버튼을 통해 레포지토리 생성' (Create repository through this button).

해당 버튼을 통해 레포지토리 생성 → **Create Repository +**

There are no repositories in this namespace.  
Tip: Not finding your repository? Try switching namespace via the top left dropdown.

# Docker 사용법

- 실습

- Docker hub 레포지토리 생성(2/2)

dockerhub Search for great content (e.g., mysql) Explore Repositories Organizations Get Help tnsghd1004

Repositories Create Using 0 of 1 private repositories. [Get more](#)

### Create Repository

tnsghd1004 Name Repository name is required

Description

Visibility

Using 0 of 1 private repositories. [Get more](#)

**Public** Public repositories appear in Docker Hub search results  **Private** Only you can view private repositories

**Build Settings** (optional)

Autobuild triggers a new build with every **git push** to your source code repository. [Learn More](#).

**Please re-link a GitHub or Bitbucket account**

We've updated how Docker Hub connects to GitHub and Bitbucket. You'll need to re-link a GitHub or Bitbucket account to create new automated builds. [Learn More](#)

Disconnected Disconnected

[Cancel](#) [Create](#) [Create & Build](#)

**Pro tip**

You can push a new image to this repository using the CLI

```
docker tag local-image:tagname new-repo:tagname
docker push new-repo:tagname
```

Make sure to change *tagname* with your desired image repository tag.

# Docker 사용법

- 실습

- 앞서 생성한 도커 이미지 레포지토리에 업로드하기
  - \$ docker tag 명령을 사용하여 이미지의 이름을 형식에 맞게 변경 수행
    - e.g., docker tag testingweb:latest tnsghd1004/webtest:testing
  - \$ docker push repo:tagname 명령시 업로드 가능
    - \$ docker push tnsghd1004:webtest:testing

```
tester@ubuntu:~$ docker push tnsghd1004/webtest:testing
The push refers to repository [docker.io/tnsghd1004/webtest]
b4e3e6fcf980: Pushed
e0b3afb09dc3: Mounted from library/ubuntu
6c01b5a53aac: Mounted from library/ubuntu
2c6ac8e5063e: Mounted from library/ubuntu
cc967c529ced: Mounted from library/ubuntu
testing: digest: sha256:f5444c439553bca0e835023c43d5357bd68f941ba033803817db0ab4343843e6 size: 1365
```

# Docker 사용법

- 실습
- 업로드한 이미지 확인

The screenshot displays the Docker Hub interface for the repository `tnsghd1004/webtest`. The repository is described as an `apache2 + php + mysql web container` and was last pushed 3 minutes ago. A single tag named `testing` is listed, which is highlighted with a red rectangular box. The Docker commands section provides the command `docker push tnsghd1004/webtest:tagname` for pushing a new tag. The 'Recent builds' section is currently empty, with a link to 'Link a source provider and run a build to see build results here.' The 'Readme' section is also empty, with a link to 'Click here to edit.'

# Docker 사용법

- 실습
  - 업로드한 이미지를 다운로드

```
tester@ubuntu:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              latest             775349758637       4 weeks ago        64.2MB
centos              latest             0f3e07c0138f       2 months ago        220MB
```

↓

```
tester@ubuntu:~$ docker pull tnsghd1004/webtest:testing
testing: Pulling from tnsghd1004/webtest
7ddbc47eeb70: Already exists
c1bbdc448b72: Already exists
8c3b70e39044: Already exists
45d437916d57: Already exists
f6433996abbf: Pull complete
Digest: sha256:f5444c439553bca0e835023c43d5357bd68f941ba033803817db0ab4343843e6
Status: Downloaded newer image for tnsghd1004/webtest:testing
docker.io/tnsghd1004/webtest:testing
tester@ubuntu:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
tnsghd1004/webtest  testing            2c542da77139       36 minutes ago     600MB
ubuntu              latest             775349758637       4 weeks ago        64.2MB
centos              latest             0f3e07c0138f       2 months ago        220MB
```

# Docker 사용법

- 실습

- 다운로드한 이미지에 대해 웹 구동 확인

```
tester@ubuntu:~$ docker run -i -t --name web-test -p 80:80 tnsghd1004/webtest:testing
root@99dcfe0063f0:/# /etc/init.d/apache2 restart
* Restarting Apache httpd web server apache2
[Sun Dec 01 22:28:07.183859 2019] [alias:warn] [pid 33] AH00671: The Alias directive in /etc/phpmyadmin/ap
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2.
```



# Docker 사용법

- 실습

- 자신이 만든 도커 이미지를 다른 환경에서 사용하기

1. 도커 이미지 다운로드

- `$ docker pull tnsghd1004/webtest:testing`

```
user@ubuntu:~$ docker pull tnsghd1004/webtest:testing
testing: Pulling from tnsghd1004/webtest
7ddbc47eeb70: Pull complete
c1bbdc448b72: Pull complete
8c3b70e39044: Pull complete
45d437916d57: Pull complete
f6433996abbf: Pull complete
Digest: sha256:f5444c439553bca0e835023c43d5357bd68f941ba033803817db0ab4343843e6
Status: Downloaded newer image for tnsghd1004/webtest:testing
docker.io/tnsghd1004/webtest:testing
user@ubuntu:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
tnsghd1004/webtest  testing            2c542da77139       2 hours ago        600MB
```

# Docker 사용법

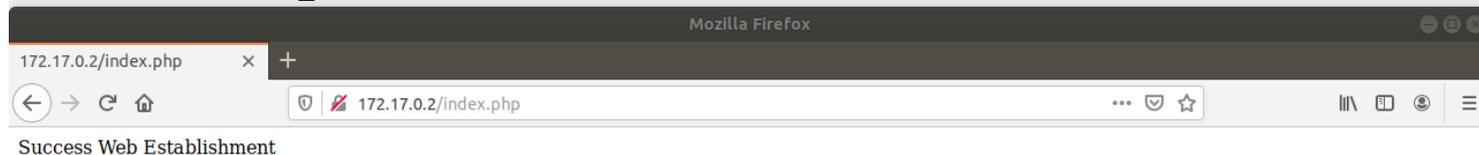
- 실습

- 자신이 만든 도커 이미지를 다른 환경에서 사용하기
  2. 다운로드 받은 이미지에 대한 컨테이너 구동 및 확인

```
user@ubuntu:~$ docker run -i -t --name webtest-other -p 80:80 tnsghd1004/webtest:testing
root@cb6976cf4bcc:/# /etc/init.d/apache2 restart
* Restarting Apache httpd web server apache2
[Mon Dec 02 00:04:57.230208 2019] [alias:warn] [pid 33] AH00671: The Alias directive in /etc/phpmyadmin/apache.conf at line 3 will probably never match because it overlaps an earlier Alias.
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 172.17.0.2. Set the 'ServerName' directive globally to suppress this message
```

[ OK ]

```
root@cb6976cf4bcc:/#
```



---

# Thanks!

권 순 홍 ([soonhong@pel.smuc.ac.kr](mailto:soonhong@pel.smuc.ac.kr))