



Operating System (OS)

Lec13 : Docker

충북대학교

강병호 (지능로봇공학과)

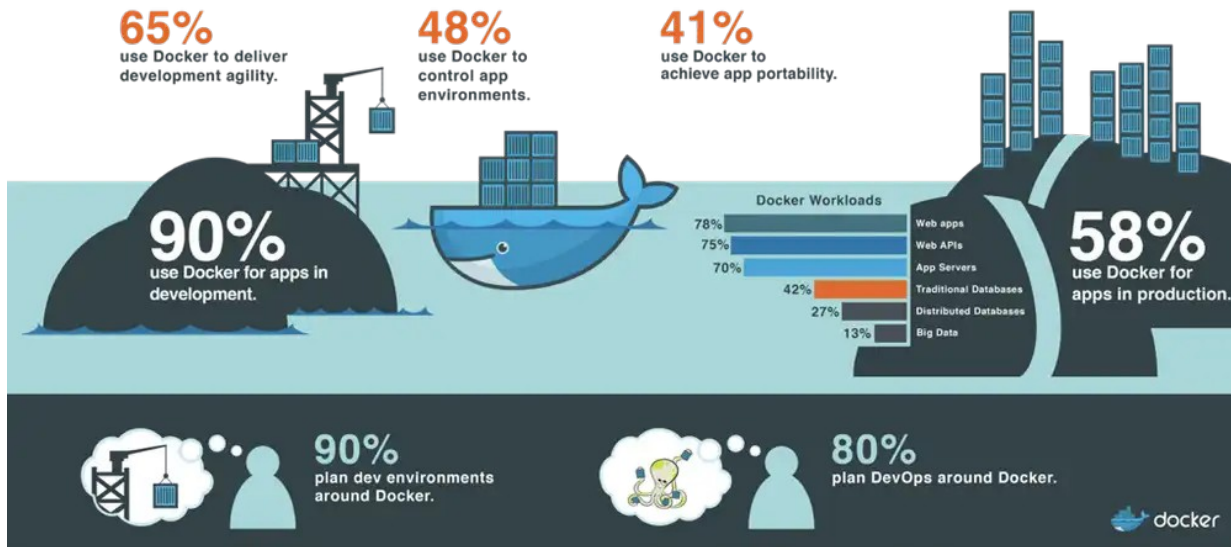
kang6283@chungbuk.ac.kr

01

docker

History

- 2013 년 3 월 Pycon Conference 에서 Solomon Hykes Docker 소개

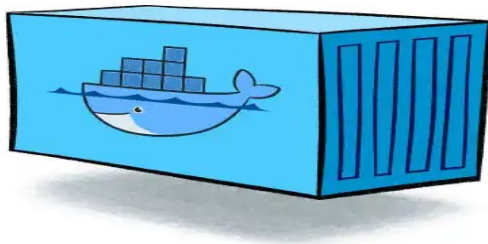


Docker 란

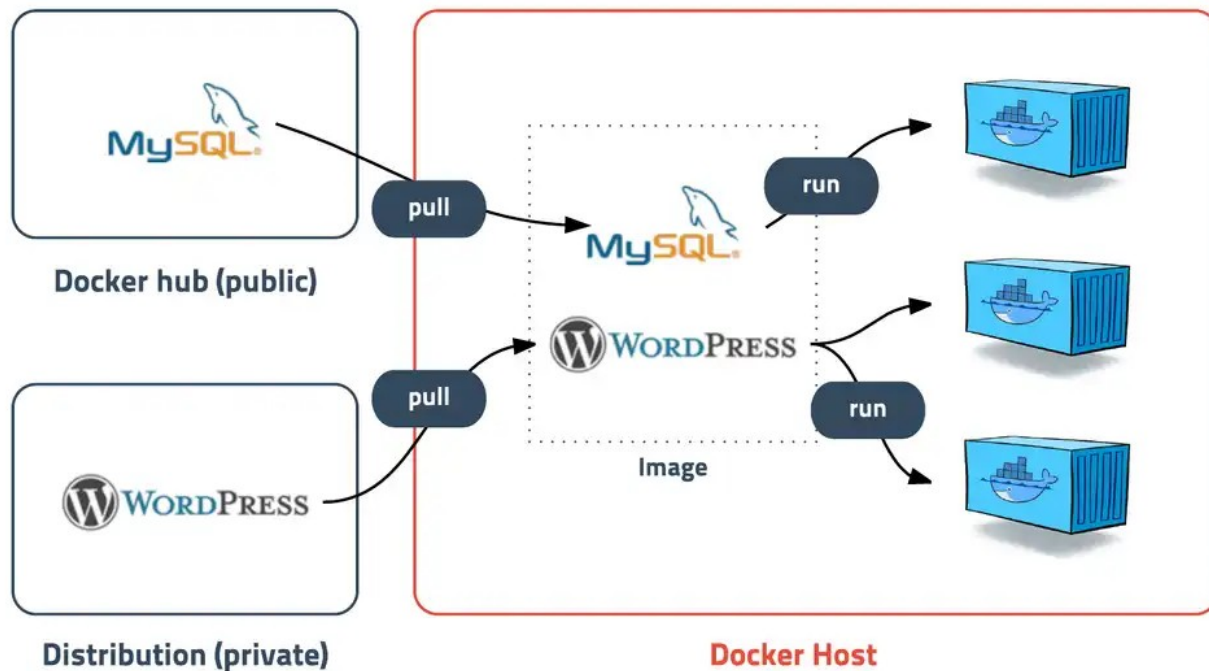
- 컨테이너

격리된 공간에서 프로세스가 동작하는 기술

컨테이너 기반의 오픈소스 가상화 플랫폼

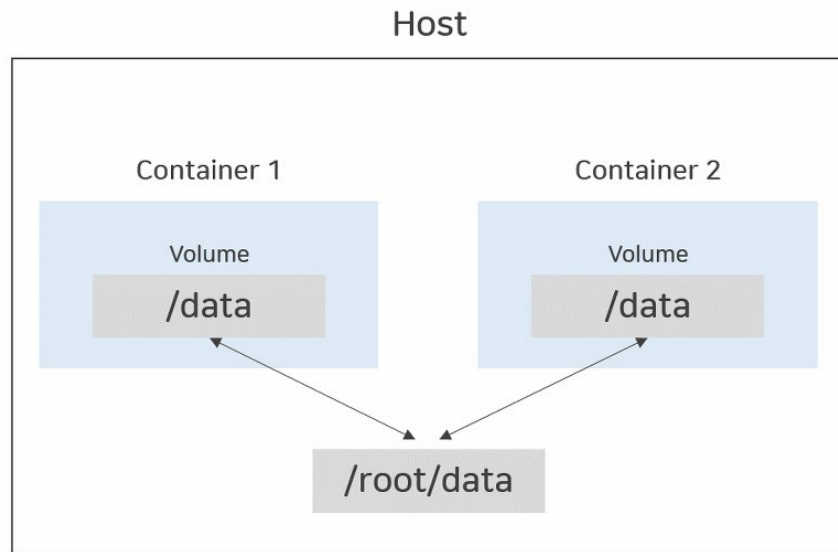


Docker Image



Docker 특징

- Volume



02

실습

Install docker

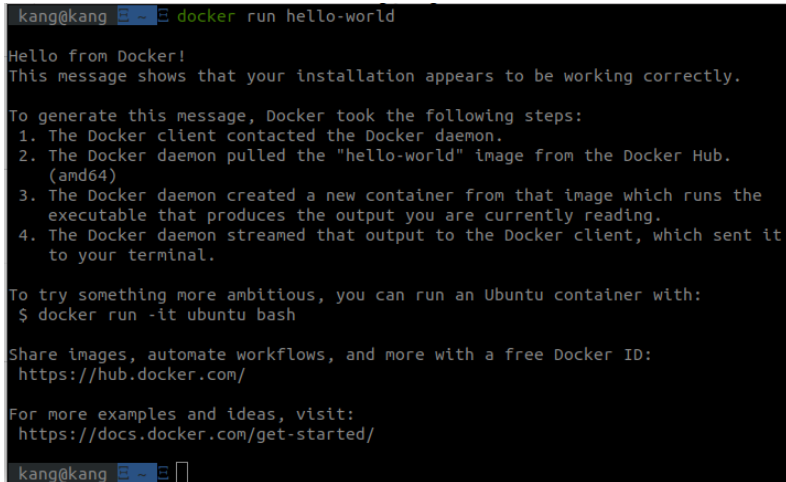
```
# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
    "deb [arch=$(dpkg --print-architecture)
signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
    $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
    sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```


Install docker

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

```
sudo docker run hello-world
```

A terminal window with a dark background. The prompt is 'kang@kang'. The command 'docker run hello-world' has been executed. The output is a multi-line message from Docker, including a greeting, a confirmation of successful installation, a list of four steps taken by Docker to generate the message, and links to Docker Hub and documentation. The prompt 'kang@kang' is visible at the bottom of the terminal.

```
kang@kang ~$ docker run hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
   (amd64)
3. The Docker daemon created a new container from that image which runs the
   executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it
   to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

kang@kang ~$
```

sudo 없이 도커 사용하기



```
sudo groupadd docker
```

```
#option 1
```

```
sudo usermod -aG docker ${user}
```

```
#option 2
```

```
sudo gpasswd -a $USER docker
```

```
sudo service docker restart
```

```
logout
```

```
docker run hello-world
```

nvidia - docker

```
distribution=$(. /etc/os-release;echo $ID$VERSION_ID) \  
    && curl -s -L https://nvidia.github.io/nvidia-docker/gpgkey | sudo apt-key  
add - \  
    && curl -s -L https://nvidia.github.io/nvidia-docker/$distribution/nvidia-  
docker.list | sudo tee /etc/apt/sources.list.d/nvidia-docker.list  
  
sudo apt-get update  
sudo apt-get install -y nvidia-docker2  
  
sudo systemctl restart docker
```

docker

- pull : dockerhub 에서 이미지를 다운받을 때

```
docker (image) pull [OPTIONS] NAME[:TAG|@DIGEST]
```

- push: dockerhub 에 이미지를 업로드 할 때

```
docker (image) push [OPTIONS] NAME[:TAG]
```

- commit: 현재 container 상태를 이미지로 저장할 때

```
docker (container) commit [OPTIONS] CONTAINER [REPOSITORY[:TAG]]
```

docker

- stop : container 를 정지시킬때

```
docker (container) stop [OPTIONS] CONTAINER [CONTAINER...]
```

- rm : container 를 삭제시킬때

```
docker (container) rm [OPTIONS] CONTAINER [CONTAINER...]
```

- rmi : 이미지를 삭제시킬때

```
docker image rm [OPTIONS] IMAGE [IMAGE...]  
docker rmi IMAGE [IMAGE...]
```

docker

- start : 중지된 컨테이너를 실행시킬 때

```
docker (container) start [OPTIONS] CONTAINER [CONTAINER...]
```

- run : 도커 이미지를 컨테이너로 만들 때

```
docker (container) run [OPTIONS] IMAGE [COMMAND] [ARG...]
```

- exec : 도커 컨테이너에 명령을 실행 할 때

```
docker (container) exec [OPTIONS] CONTAINER COMMAND [ARG...]
```

Dockerhub

The screenshot shows the Docker Hub interface for the `osrf/ros` repository. The page is viewed in a web browser with the URL `hub.docker.com/r/osrf/ros/tags`. The repository is sponsored by the Open Source Robotics Foundation and has 170 stars. It describes the Robot Operating System (ROS) as an open source project for building robot applications. The page is divided into 'Overview' and 'Tags' sections. The 'Tags' section is active, showing a list of Docker images. The first tag is `noetic-desktop-full`, pushed 8 hours ago by `osrfbot`. It has a digest of `f2063147b3a7` and a compressed size of 995.73 MB. The second tag is `humble-desktop-full`, also pushed 8 hours ago by `osrfbot`, with a digest of `9e71c0d1327f` and a compressed size of 1.1 GB. The third tag is `humble-desktop-full-jammy`, pushed 8 hours ago by `osrfbot`. Each tag entry includes a 'Copy' button for the pull command.

osrf/ros Tags | Docker Hub

hub.docker.com/r/osrf/ros/tags

dockerhub

osrf/ros Sponsored OSS ☆170

By Open Source Robotics Foundation · Updated about 8 hours ago

The Robot Operating System (ROS) is an open source project for building robot applications.

LANGUAGES & FRAMEWORKS INTEGRATION & DELIVERY INTERNET OF THINGS

Overview Tags

Sort by Newest Filter Tags

TAG noetic-desktop-full Last pushed 8 hours ago by osrfbot	Digest f2063147b3a7	OS/ARCH linux/amd64	<code>docker pull osrf/ros:noetic-desktop-full</code> Copy	Compressed Size ⓘ 995.73 MB
TAG humble-desktop-full Last pushed 8 hours ago by osrfbot	Digest 9e71c0d1327f	OS/ARCH linux/amd64	<code>docker pull osrf/ros:humble-desktop-full</code> Copy	Compressed Size ⓘ 1.1 GB
TAG humble-desktop-full-jammy Last pushed 8 hours ago by osrfbot			<code>docker pull osrf/ros:humble-desktop-full-jammy</code> Copy	

Docker

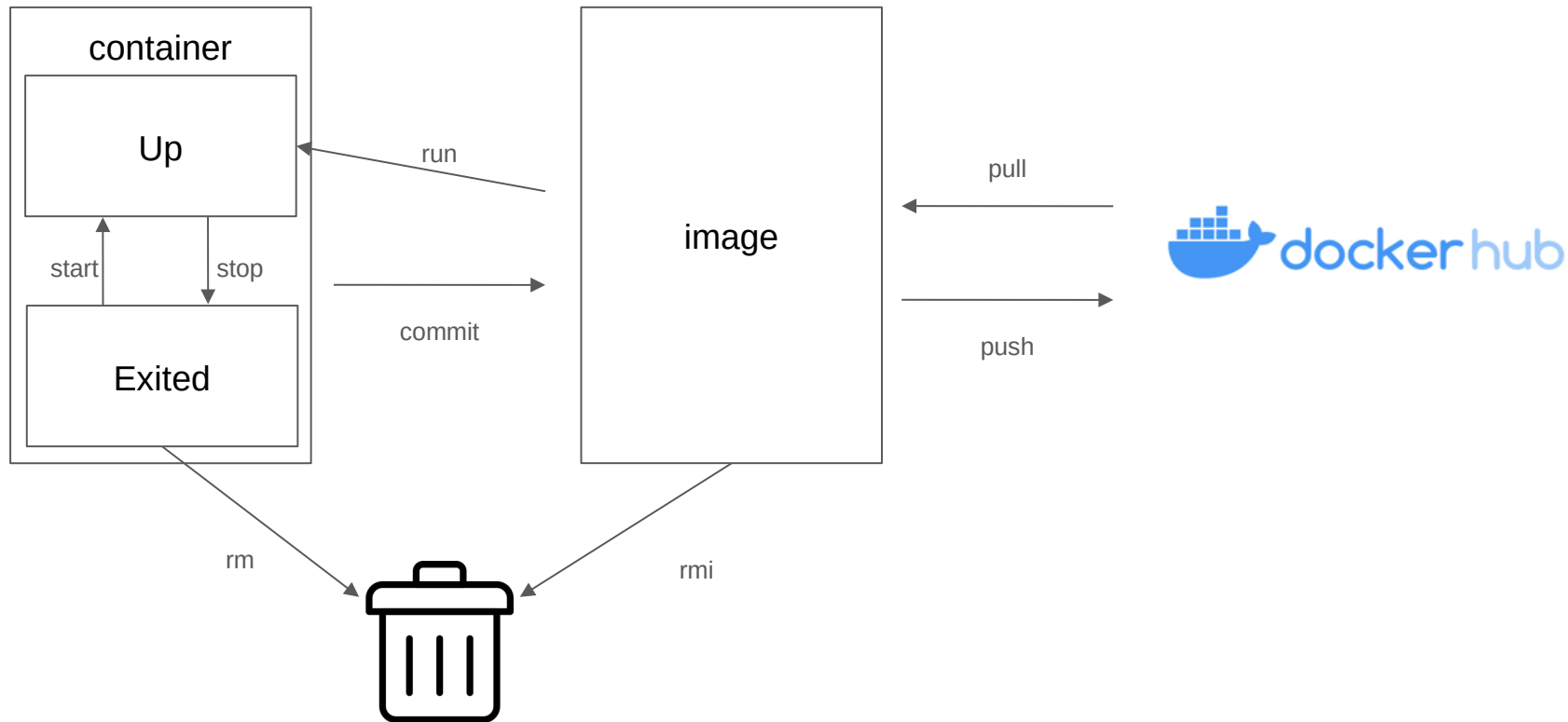
```
docker pull osrf/ros:humble-desktop-jammy
```

```
docker run -it --name □□ osrf/ros:humble-desktop-jammy
```


Docker script

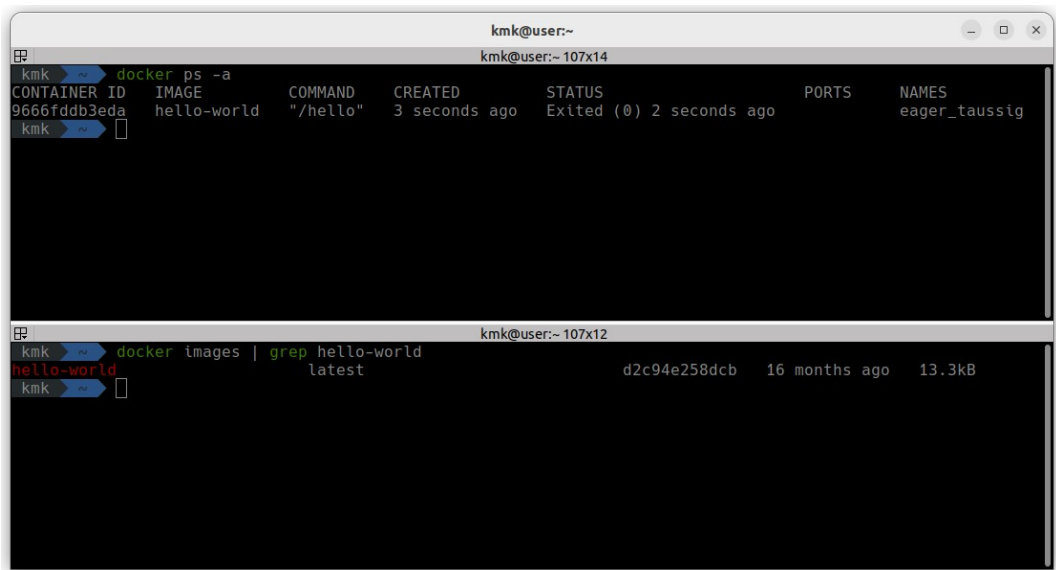
```
docker run --gpus all -it --privileged \  
-e DISPLAY=$DISPLAY \  
--env="QT_X11_NO_MITSHM=1" \  
-e NVIDIA_DRIVER_CAPABILITIES=all \  
-v /tmp/.X11-unix:/tmp/.X11-unix \  
-v /dev:/dev:rw \  
-u user \  
-w /home/user \  
--hostname $(hostname) \  
--network host \  
--name container_name image_name:tag command
```

Docker



Docker

- `ps -a` : 컨테이너 목록을 확인
- `images` : 이미지 목록을 확인



The image shows two terminal windows. The top window displays the output of the `docker ps -a` command, showing a single container with ID `9666fdbb3eda` based on the `hello-world` image, which has exited. The bottom window displays the output of the `docker images | grep hello-world` command, showing the `hello-world` image with ID `d2c94e258dcb`, created 16 months ago, with a size of 13.3kB.

```
kmk@user:~  
kmk@user:~ 107x14  
kmk ➜ ~ docker ps -a  
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES  
9666fdbb3eda   hello-world  "/hello"   3 seconds ago    Exited (0) 2 seconds ago           eager_taussig  
kmk ➜ ~  
  
kmk@user:~ 107x12  
kmk ➜ ~ docker images | grep hello-world  
hello-world    latest    d2c94e258dcb  16 months ago  13.3kB  
kmk ➜ ~
```

Docker user

```
useradd --shell /bin/bash -u 1001 -c "" -m user && usermod -a -G dialout user
```