

YoonJoon Lee
SoC KAIST

What we will take a look in this series

- 1.Docker?
- 2.Image & Container
- 3.Publish & Deploy

What we will take a look today

- 1.Docker?
- 2.Image & Container
- 3.Publish & Deploy

What I will talk about in this section

1. Why Virtualization?

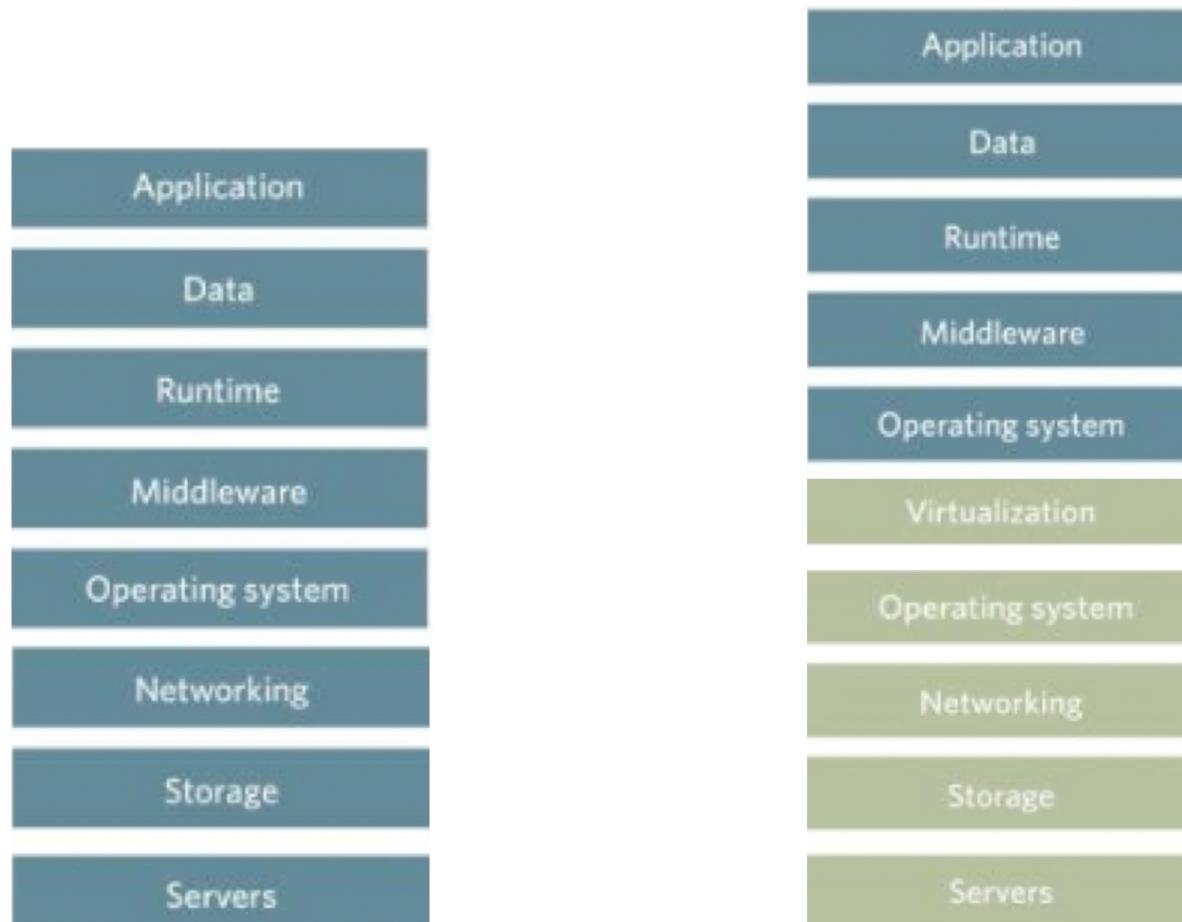
2. What is Docker & Why?

3. IMAGES

4. CONTAINERS

5. SERVICES

Why Virtualization?



What is Docker?



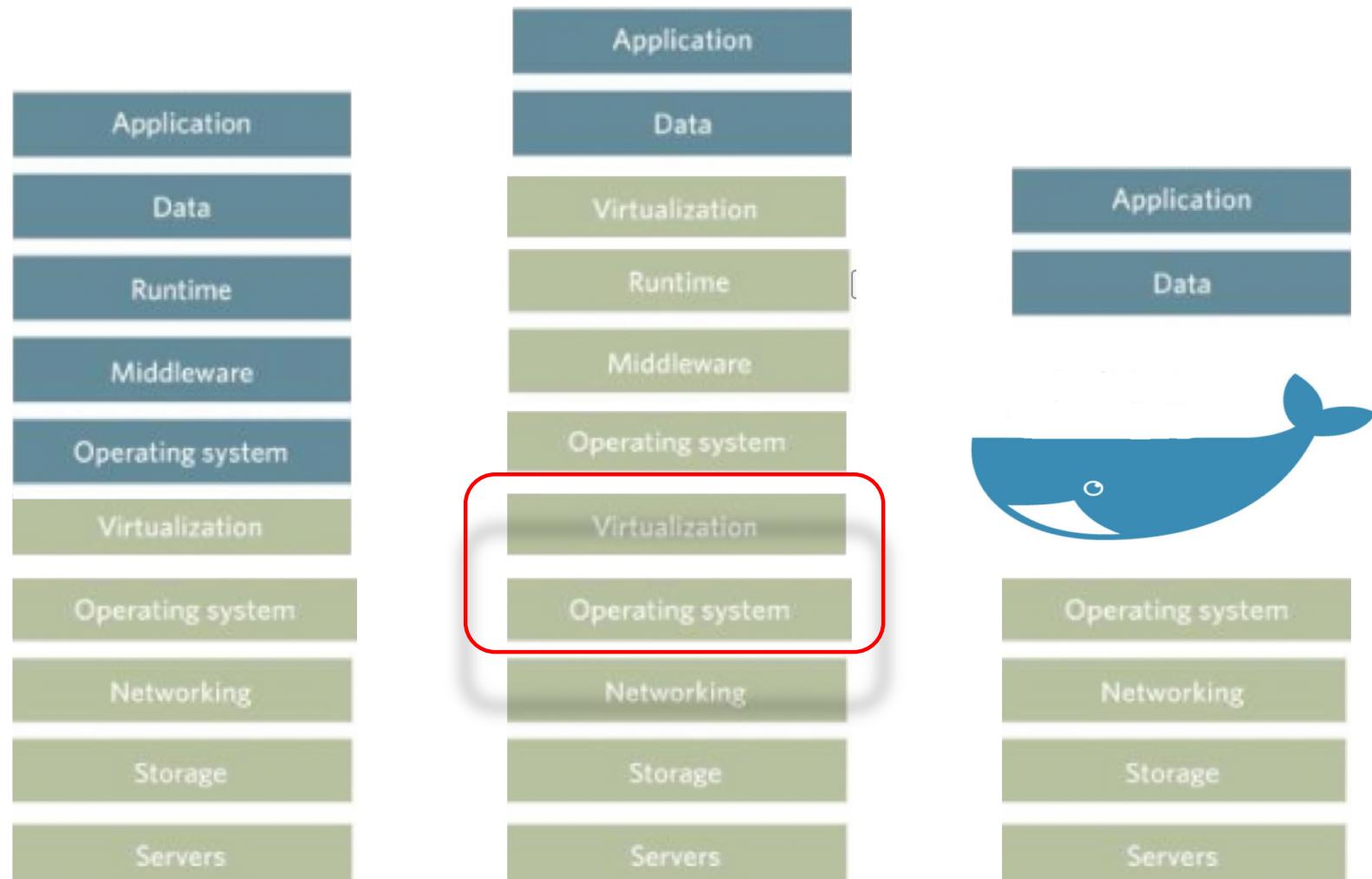
(source: <https://docs.docker.com/get-started/>)

Docker는 개발자 및 시스템 관리자가 컨테이너를 사용하여 응용프로그램의 **개발, 전개 및 실행**을 위한 플랫폼입니다.

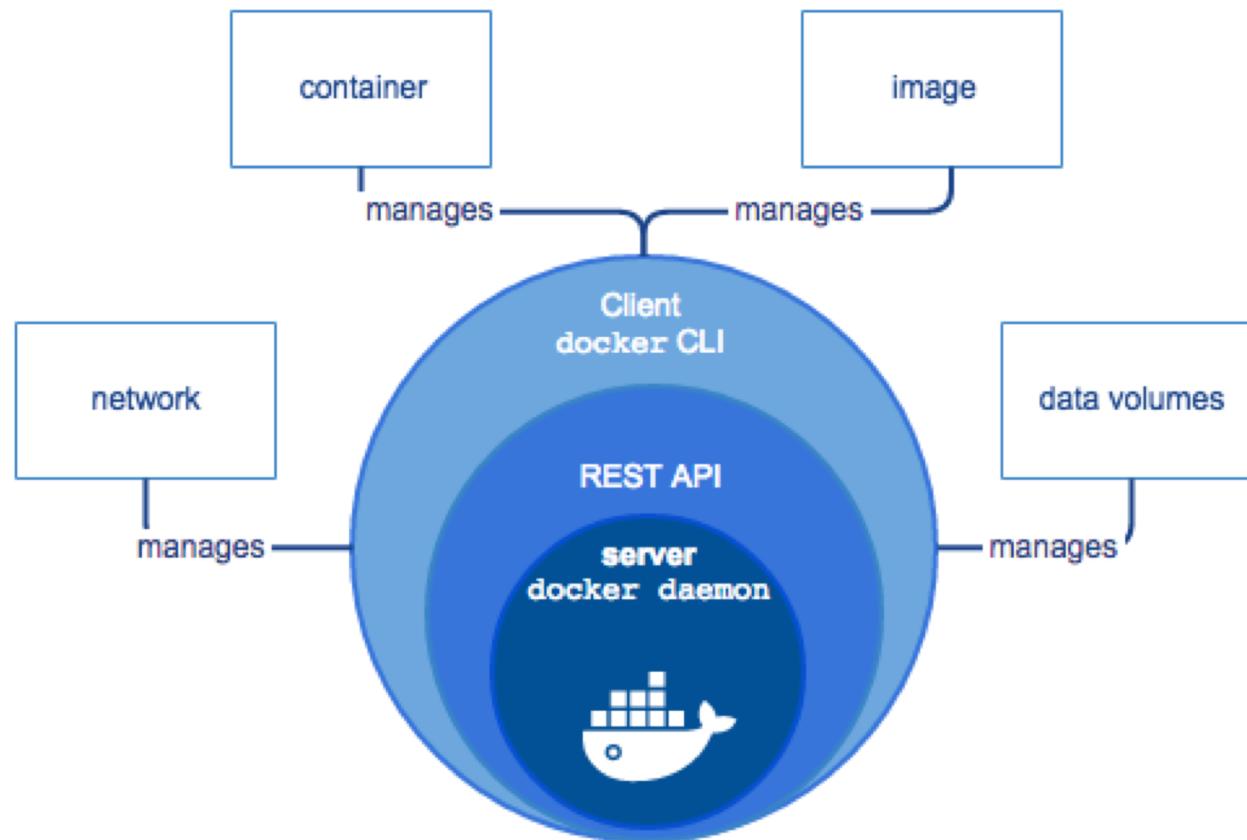
Containerization

- 유연성
- 경량
- 무중단 배포
- 이식성
- 확장성
- 스태킹

Why Docker?

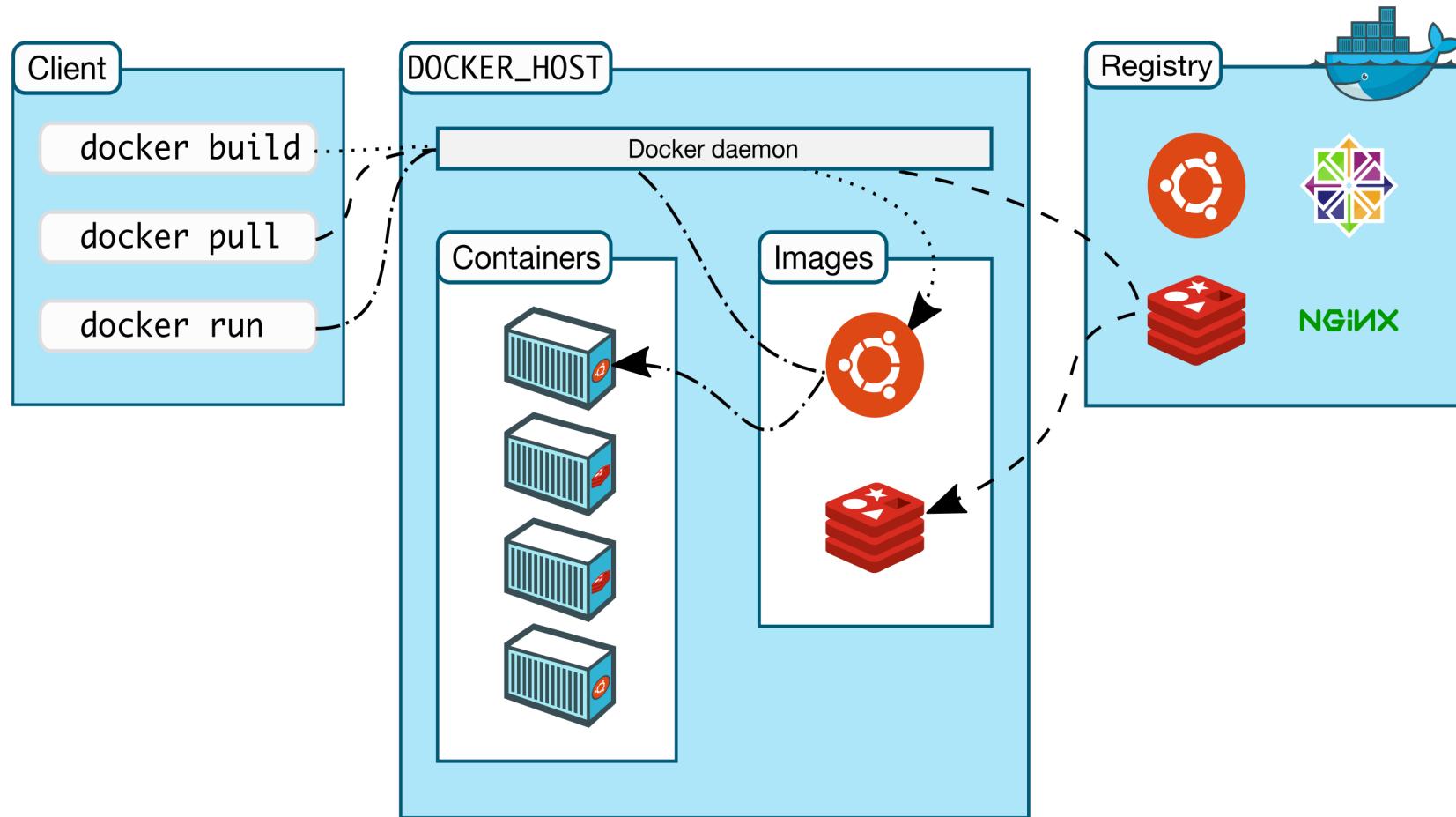


Docker Engine?



(source: <https://docs.docker.com/engine/docker-overview/>)

Docker Architecture?



(source: <https://docs.docker.com/engine/docker-overview/>)

Images

- Docker 컨테이너를 생성하기 위한 지침의 읽기 전용 템플릿
- 이미지를 만들려면 이미지를 만들고 실행하는 데 필요한 단계를 정의하는 간단한 구문의 **Dockerfile** 필요

```
# Use an official Python runtime as a parent image
FROM python:3.6

# Set the working directory to /app
WORKDIR /app

# Copy the current directory contents into the container at /app
ADD . /app

# Install any needed packages specified in requirements.txt
RUN pip install --trusted-host pypi.python.org -r requirements.txt

# Make port 80 available to the world outside this container
EXPOSE 80

# Define environment variable
ENV NAME World

# Run app.py when the container launches
CMD ["python3", "app.py"]
```

requirements.txt

```
Flask
Redis
```

app.py

```
from flask import Flask
from redis import Redis, RedisError
import os
import socket

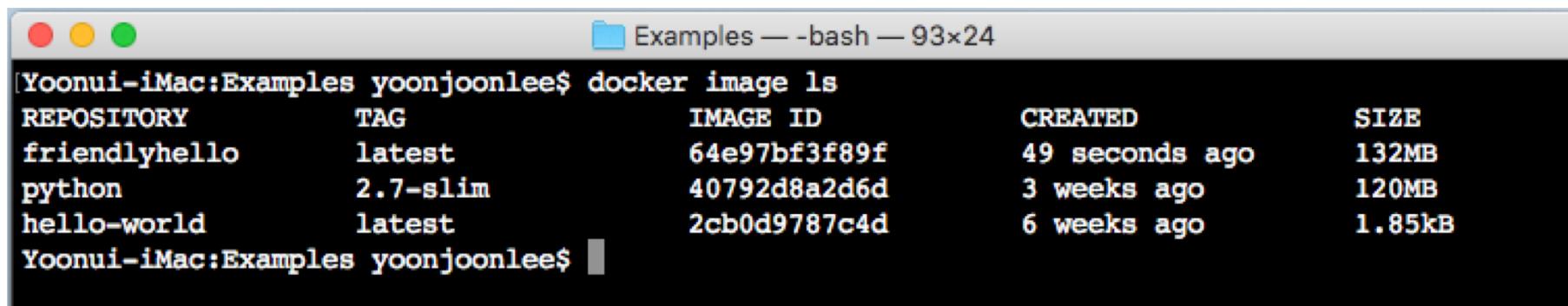
# Connect to Redis
redis = Redis(host="redis", db=0, socket_connect_timeout=2, socket_timeout=2)

app = Flask(__name__)

@app.route("/")
def hello():
    try:
        visits = redis.incr("counter")
    except RedisError:
        visits = "<i>cannot connect to Redis, counter disabled</i>"

    html = "<h3>Hello {name}!</h3> \
            <b>Hostname:</b> {hostname}<br/> \
            <b>Visits:</b> {visits}"
    return html.format(name=os.getenv("NAME", "world"), hostname=socket.gethostname(), visits=visits)

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=80)
```



A screenshot of a macOS terminal window titled "Examples — bash — 93x24". The window shows the command "docker image ls" being run in a directory named "Examples" on a Mac. The output lists four Docker images:

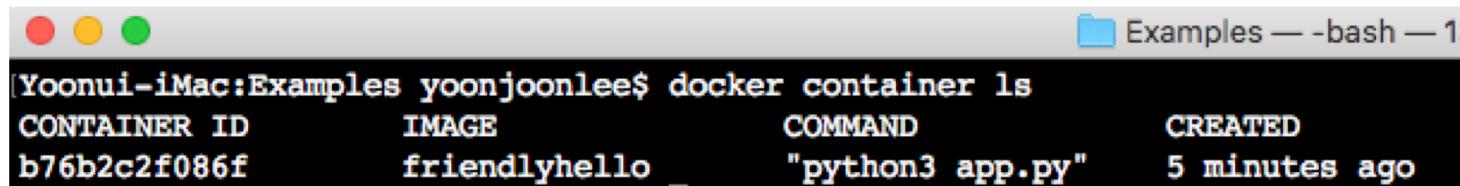
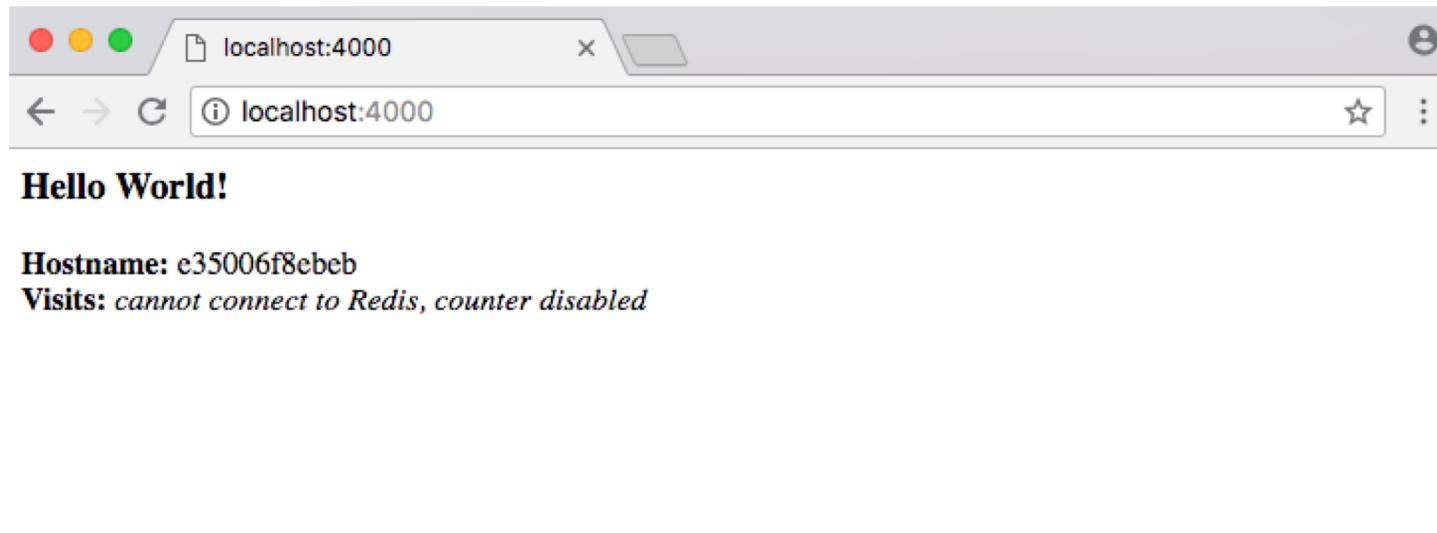
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
friendlyhello	latest	64e97bf3f89f	49 seconds ago	132MB
python	2.7-slim	40792d8a2d6d	3 weeks ago	120MB
hello-world	latest	2cb0d9787c4d	6 weeks ago	1.85kB

The terminal prompt at the bottom is "Yoonui-iMac:Examples yoonjoonlee\$".

Containers

- 이미지를 실행할 수 있는 인스턴스

```
docker run -p 4000:80 friendlyhello
```



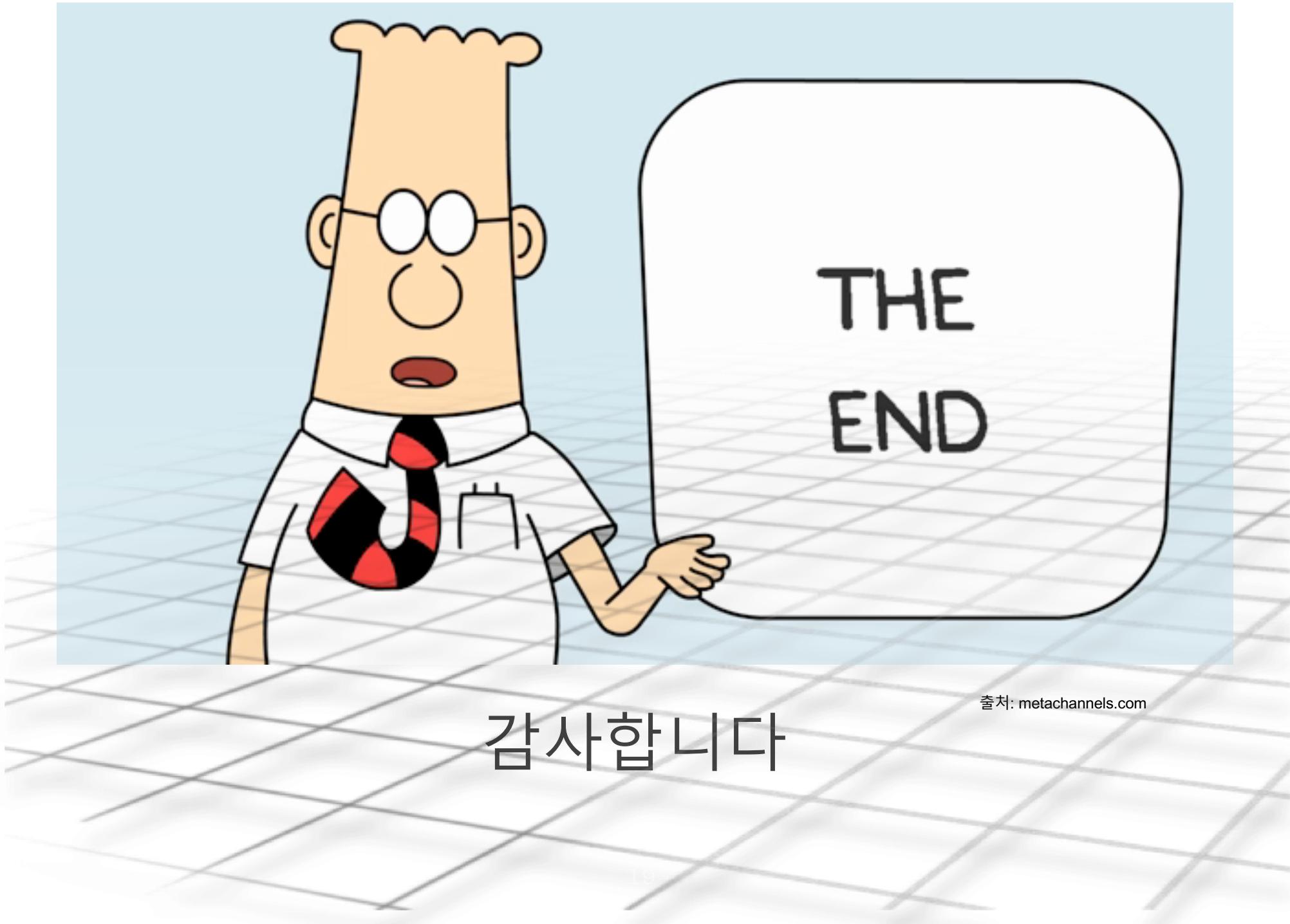
Services

- 서비스를 사용하여 컨테이너를 Docker 여러 데몬으로 확장
- swarm(manager + worker)으로 모든 작업을 함께 실행

What we will take a look next

- 1.Docker?
- 2.Image & Container
- 3.Publish & Deploy

Going beyond the requirements...



출처: metachannels.com

감사합니다