

docker

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

(<https://github.com/YoonJoon/DockerTutorial/blob/master/containers.md>)

1. Building an app the Docker way
2. Dockerfile
3. Share Images

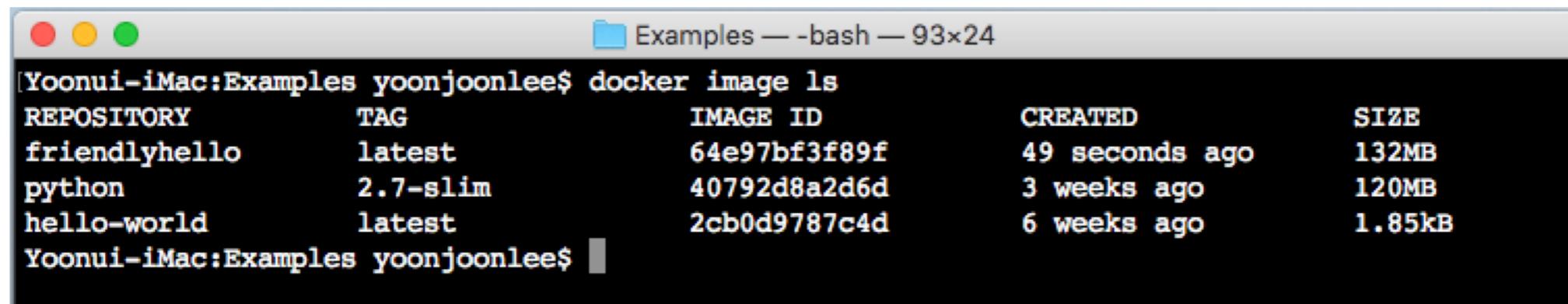
Prerequisites

```
docker run hello-world
```

It All Starts with a Dockerfile

- **Dockerfile** that produces a Docker image.

```
$ docker build -t friendlyhello .
```



The screenshot shows a terminal window with the title "Examples — -bash — 93x24". The command "docker image ls" is run, listing three images:

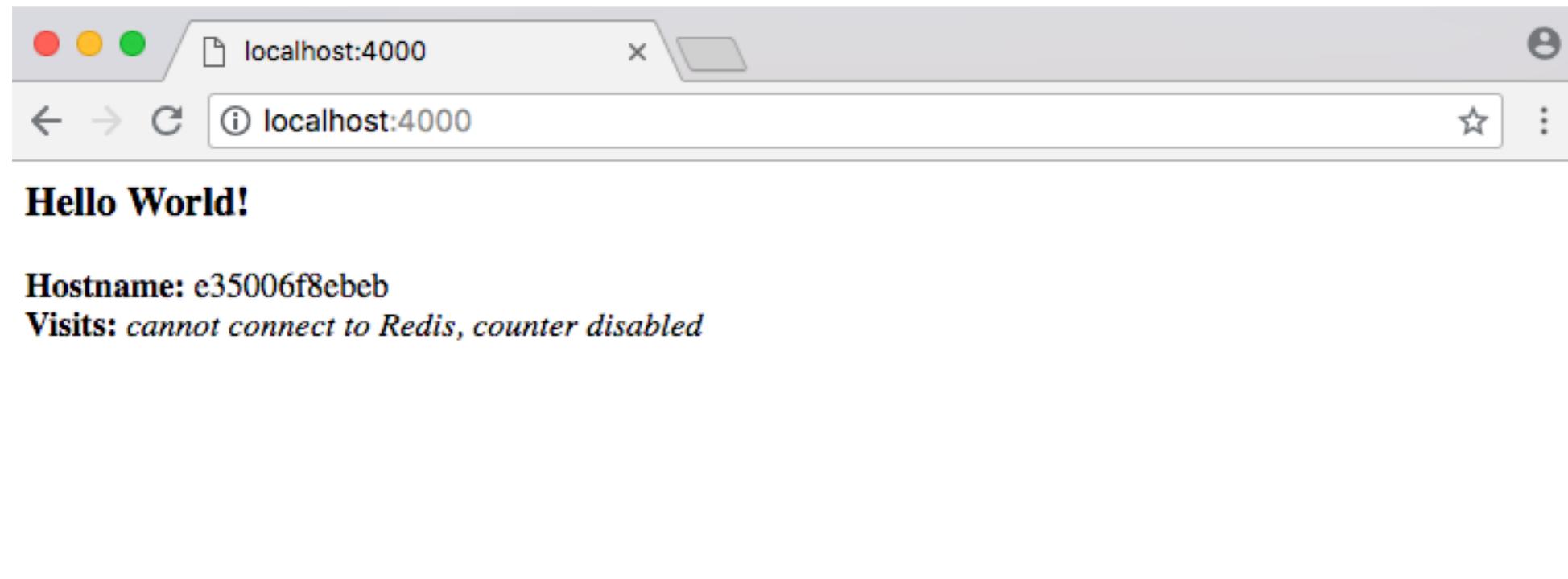
REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
friendlyhello	latest	64e97bf3f89f	49 seconds ago	132MB
python	2.7-slim	40792d8a2d6d	3 weeks ago	120MB
hello-world	latest	2cb0d9787c4d	6 weeks ago	1.85kB

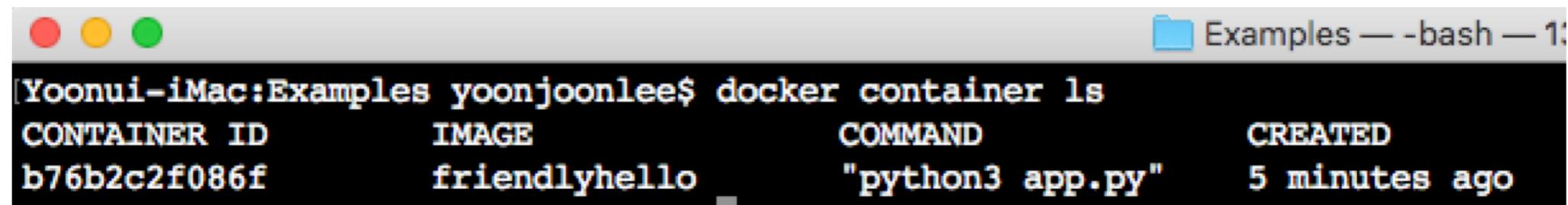
Docker image is considered as a class, where as a Docker container is an instance of that class

If Build It, They will Run (Usually)

- Running a Docker image creates a Docker container, that's it.

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





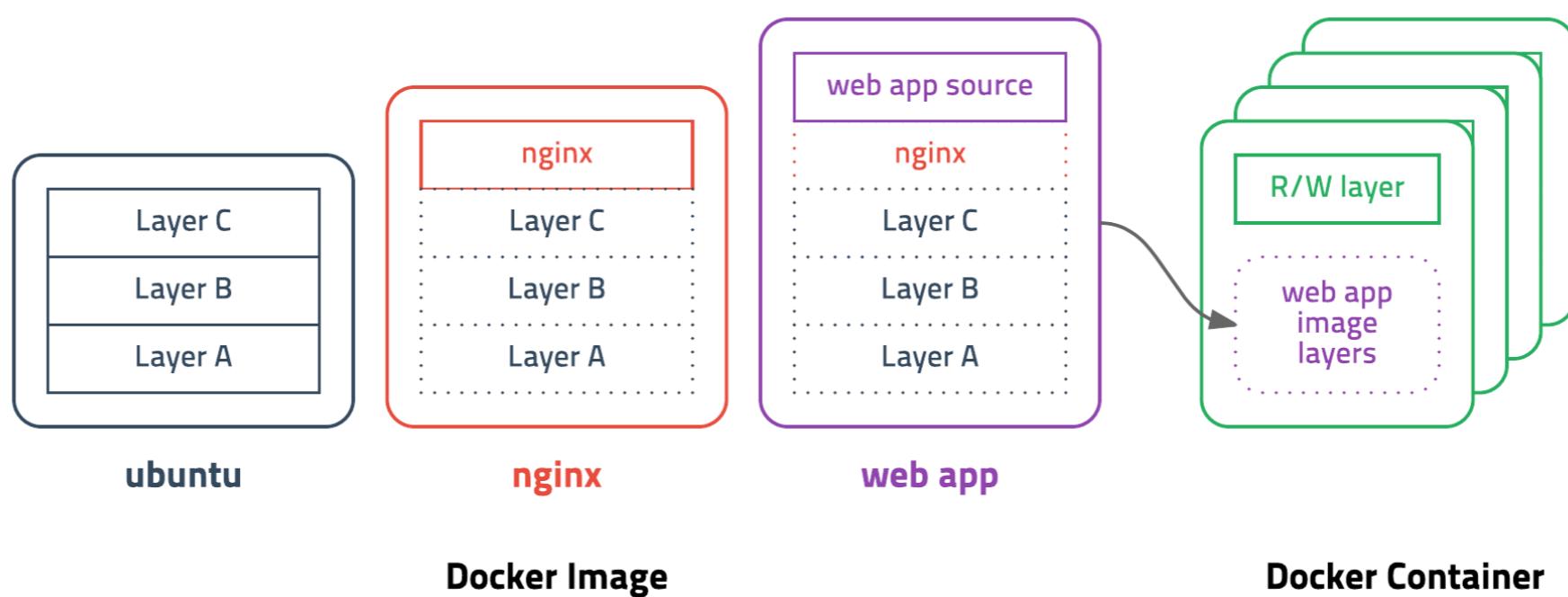
A screenshot of a macOS terminal window titled "Examples — bash — 1". The window shows the command "docker container ls" being run. The output lists one container:

CONTAINER ID	IMAGE	COMMAND	CREATED
b76b2c2f086f	friendlyhello	"python3 app.py"	5 minutes ago

`docker container stop b76b2c2f086f`

Docker Image

- A Docker image consists of read-only layers each of which represents a Dockerfile instruction.
- The layers are stacked and each one is a delta of the changes from the previous layer.



Dockerfile

<https://docs.docker.com/engine/reference/builder/#environment-replacement>

- Environment replacement
- FROM
- RUN
- CMD

Environment Replacement

Environment variables are supported by the following list of instructions in the Dockerfile:

- ADD
- COPY
- ENV
- EXPOSE
- FROM
- LABEL
- STOPSIGNAL
- USER
- VOLUME
- WORKDIR

as well as:

- ONBUILD (when combined with one of the supported instructions above)

Environment Replacement

```
# 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"]
```

FROM

```
# 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"]
```

RUN

```
# 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"]
```

CMD

```
# 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"]
```

Share Images

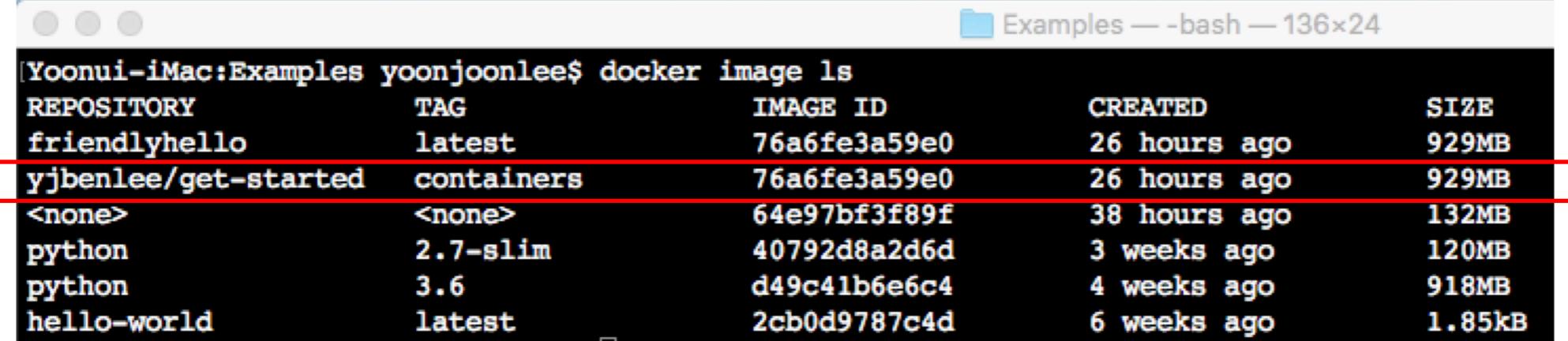
- Log in with Docker ID

```
docker login
```

- Tag the image

```
docker tag image username/repository:tag
```

```
docker tag friendlyhello yjbenlee/  
get-started:part2
```



REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
friendlyhello	latest	76a6fe3a59e0	26 hours ago	929MB
yjbenlee/get-started	containers	76a6fe3a59e0	26 hours ago	929MB
<none>	<none>	64e97bf3f89f	38 hours ago	132MB
python	2.7-slim	40792d8a2d6d	3 weeks ago	120MB
python	3.6	d49c41b6e6c4	4 weeks ago	918MB
hello-world	latest	2cb0d9787c4d	6 weeks ago	1.85kB

- Publish the image

```
docker push username/repository:tag
```

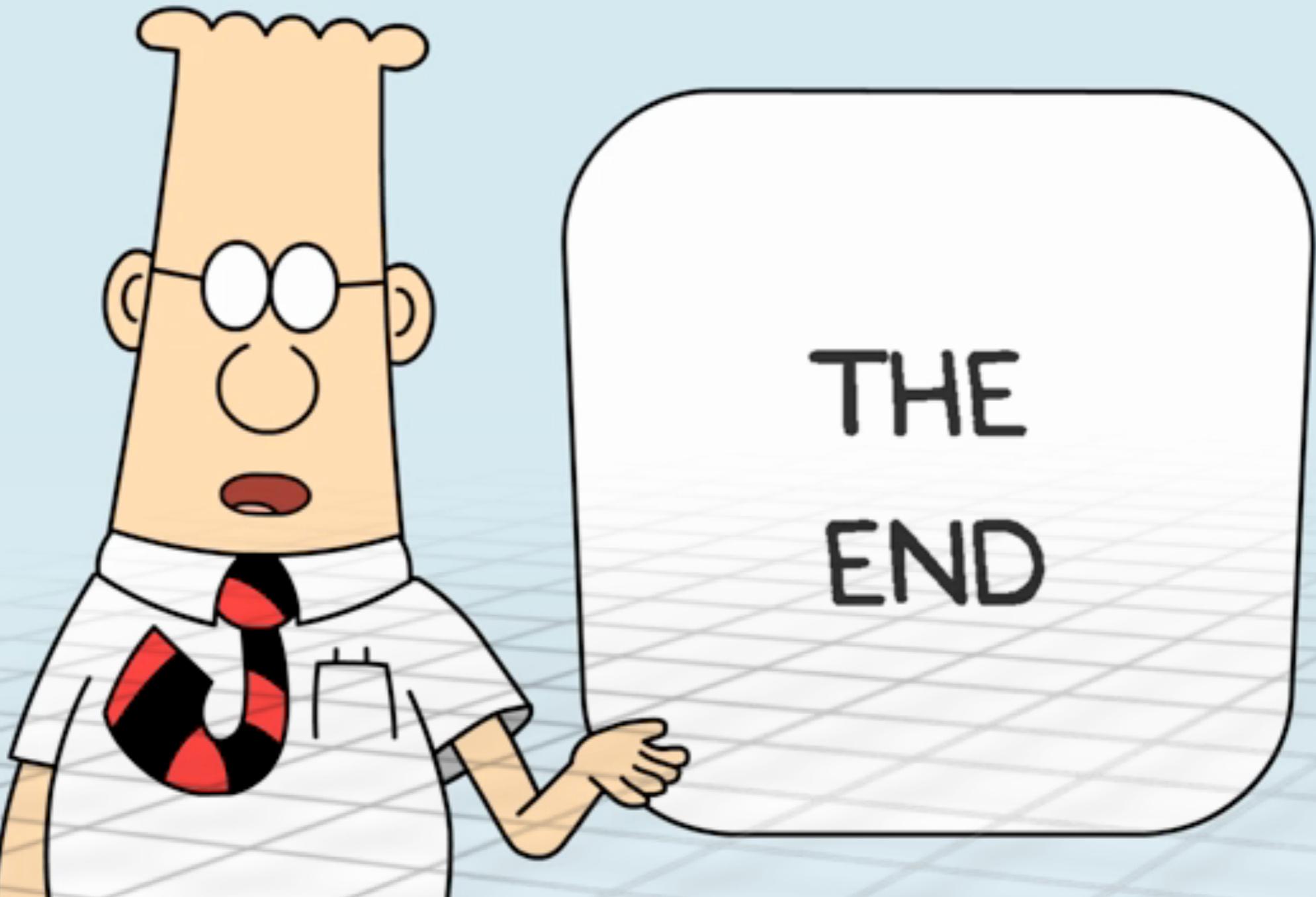
- Pull and run the image from the remote repository

```
docker run -p 4000:80 username/repository:tag
```



A screenshot of a terminal window titled "yjlee — docker run -p 4000:80 yjbenlee/get-started:containers — 91x25". The terminal shows the process of pulling an image and running a container. It starts with a "Last login" message, followed by the command being run. It then shows the image being pulled from a remote repository, with each layer's download progress listed. Finally, it shows the container starting up, including the Flask application configuration and a warning about using a development server in production.

```
Last login: Fri Aug 17 15:43:36 on ttys000
[YoonJoon-Lees-MacBook-Pro:~ yjlee$ docker run -p 4000:80 yjbenlee/get-started:containers
Unable to find image 'yjbenlee/get-started:containers' locally
containers: Pulling from yjbenlee/get-started
55cbf04beb70: Pull complete
1607093a898c: Pull complete
9a8ea045c926: Pull complete
d4eee24d4dac: Pull complete
b59856e9f0ab: Pull complete
b023afffd10b: Pull complete
4d4eb448d315: Pull complete
c4eb58602129: Pull complete
598629fb90fc: Pull complete
0100922d2a4a: Pull complete
4d848cdf405b: Pull complete
4b37415bbf2d: Pull complete
Digest: sha256:6e2e6b550e315d63bd359ff8da6eb01fb4a42a05e80c7c6b124fb4f90e0e9a1a
Status: Downloaded newer image for yjbenlee/get-started:containers
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: Do not use the development server in a production environment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
```



감사합니다

출처: metachannels.com