

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/services.md>)

1. Service?
2. Docker-compose.yml
3. Scale out & Load Balancing

Prerequisites

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

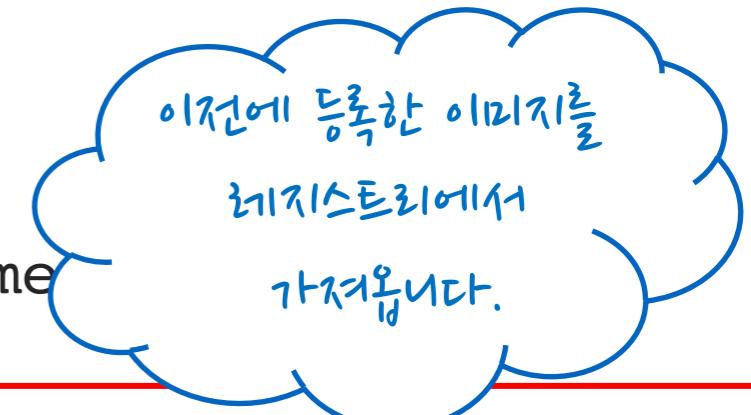
About Services

- Different pieces of the app in a distributed application.
- Just “containers in production.”
- A service only runs one image, but it codifies the way that image runs.
- Scaling a service changes the number of container instances.

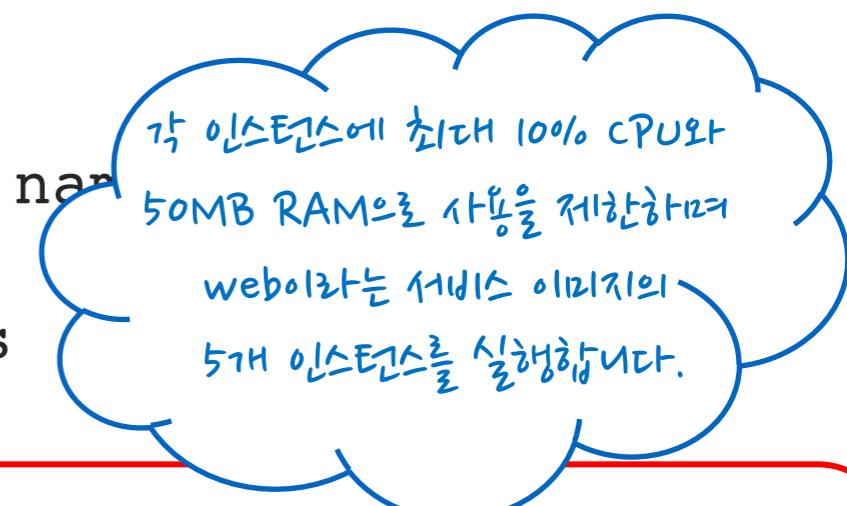
Docker-compose.yml

- A YAML(YAML Ain't Markup Language) file that defines how Docker containers should behave in production.

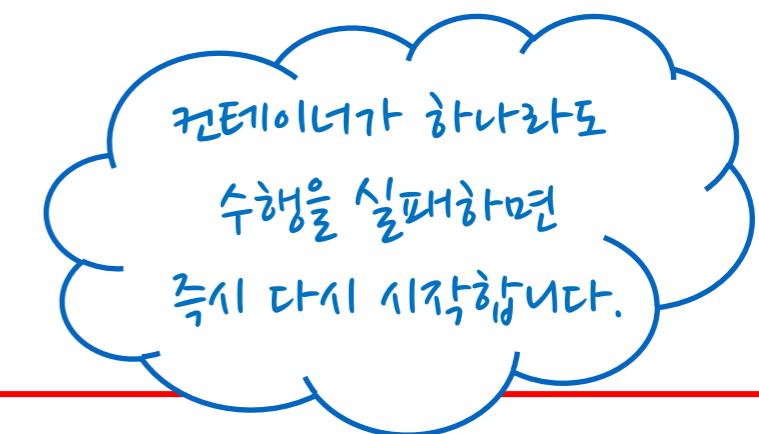
```
version: "3"
services:
  web:
    # replace username/repo:tag with your name
    details
      image: yjbenlee/get-started:containers
  deploy:
    replicas: 5
    resources:
      limits:
        cpus: "0.1"
        memory: 50M
    restart_policy:
      condition: on-failure
  ports:
    - "4000:80"
  networks:
    - webnet
networks:
  webnet:
```



```
version: "3"
services:
  web:
    # replace username/repo:tag with your name and details
    image: yjbenlee/get-started:containers
    deploy:
      replicas: 5
      resources:
        limits:
          cpus: "0.1"
          memory: 50M
      restart_policy:
        condition: on-failure
    ports:
      - "4000:80"
  networks:
    - webnet
networks:
  webnet:
```



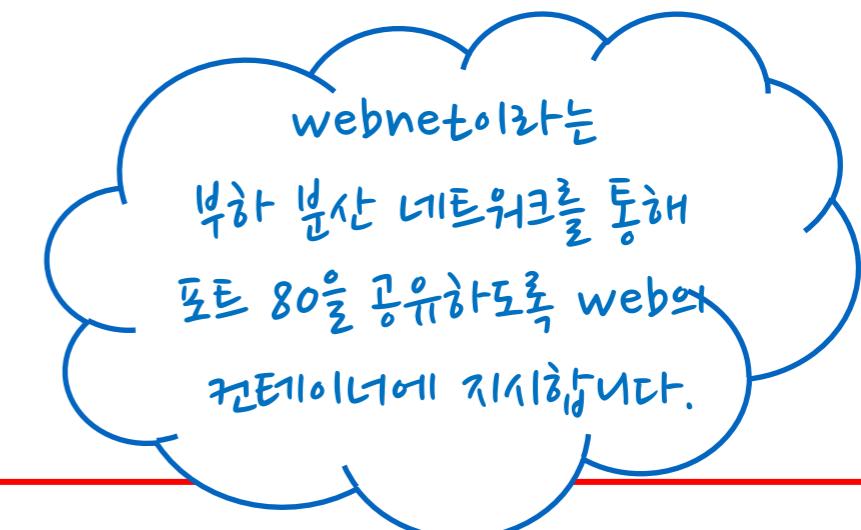
```
version: "3"
services:
  web:
    # replace username/repo:tag with your name and image
    details
    image: yjbenlee/get-started:containers
    deploy:
      replicas: 5
      resources:
        limits:
          cpus: "0.1"
          memory: 50M
      restart_policy:
        condition: on-failure
    ports:
      - "4000:80"
  networks:
    - webnet
networks:
  webnet:
```



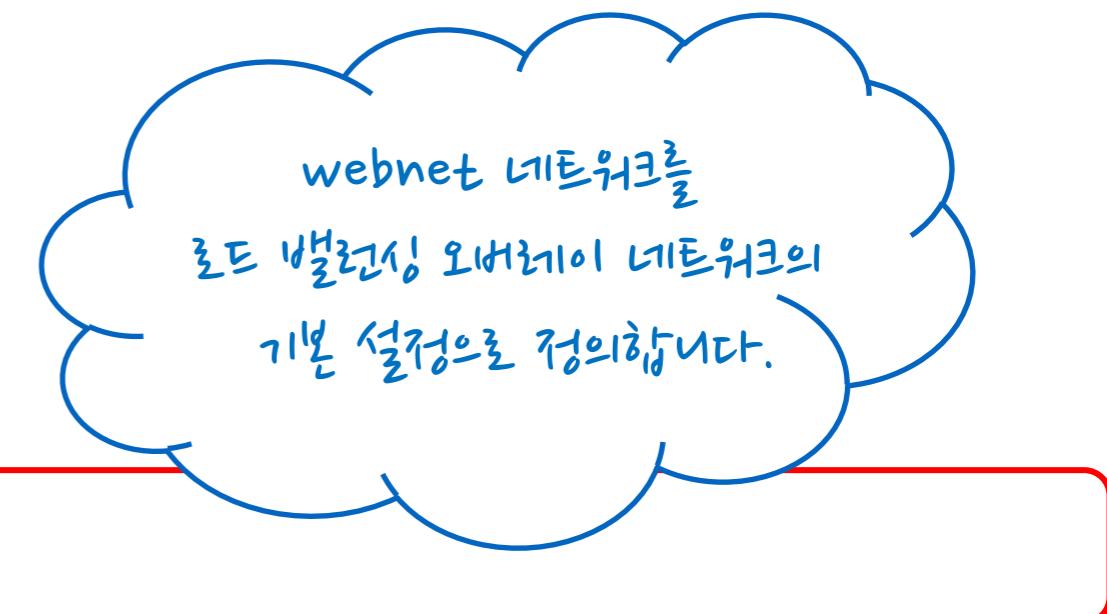
```
version: "3"
services:
  web:
    # replace username/repo:tag with your name and image
    details
    image: yjbenlee/get-started:containers
    deploy:
      replicas: 5
      resources:
        limits:
          cpus: "0.1"
          memory: 50M
      restart_policy:
        condition: on-failure
    ports:
      - "4000:80"
  networks:
    - webnet
networks:
  webnet:
```



```
version: "3"
services:
  web:
    # replace username/repo:tag with your name and image
    details
    image: yjbenlee/get-started:containers
    deploy:
      replicas: 5
      resources:
        limits:
          cpus: "0.1"
          memory: 50M
      restart_policy:
        condition: on-failure
    ports:
      - "4000:80"
  networks:
    - webnet
networks:
  webnet:
```



```
version: "3"
services:
  web:
    # replace username/repo:tag with your name and image
    details
    image: yjbenlee/get-started:containers
    deploy:
      replicas: 5
      resources:
        limits:
          cpus: "0.1"
          memory: 50M
      restart_policy:
        condition: on-failure
    ports:
      - "4000:80"
  networks:
    - webnet
networks:
  webnet:
```



Load Balancing

```
docker swarm init
```

Load Balancing

```
docker swarm init
```

```
docker stack deploy --compose-file=docker-compose.yml  
getstartedlab
```

Load Balancing

```
docker swarm init
```

```
docker stack deploy --compose-file=docker-compose.yml  
getstartedlab
```

```
docker service ls
```

Load Balancing

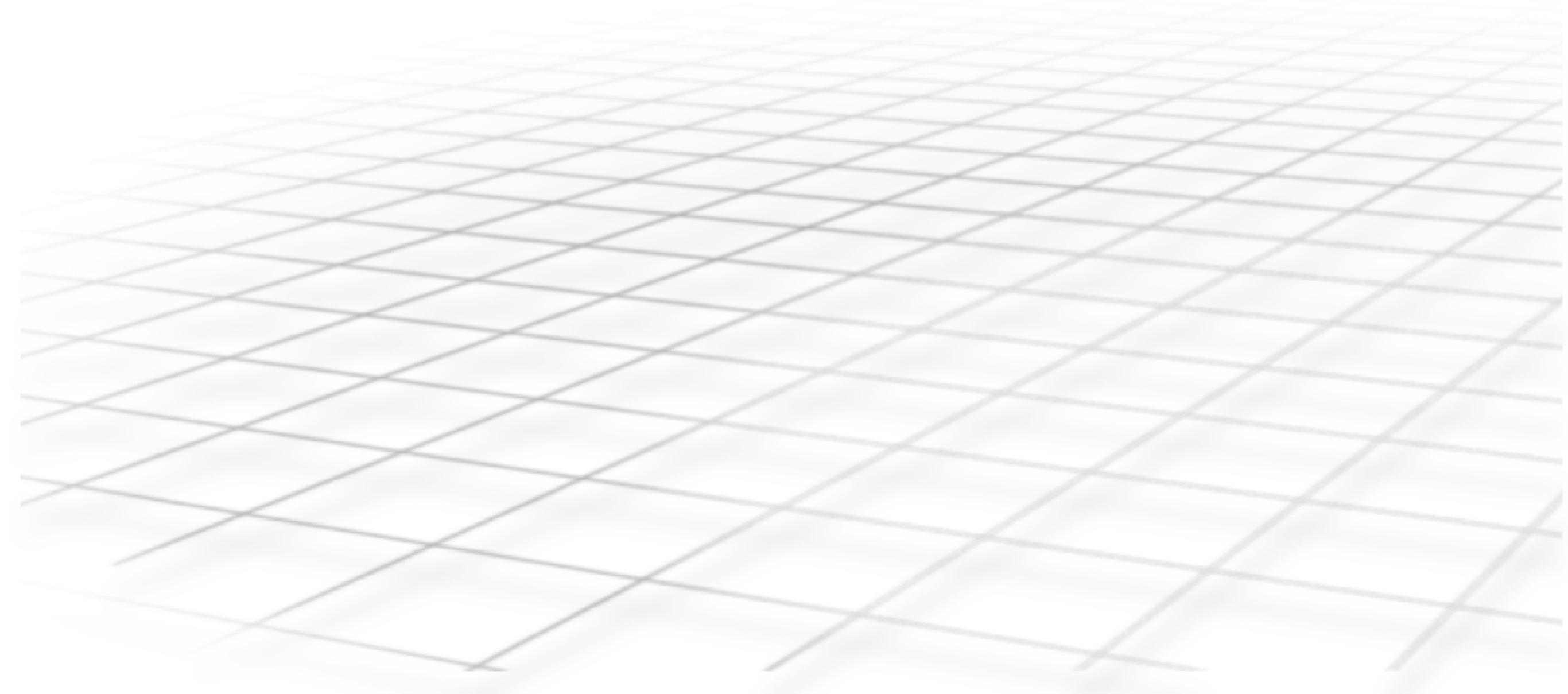
```
docker swarm init
```

```
docker stack deploy --compose-file=docker-compose.yml  
getstartedlab
```

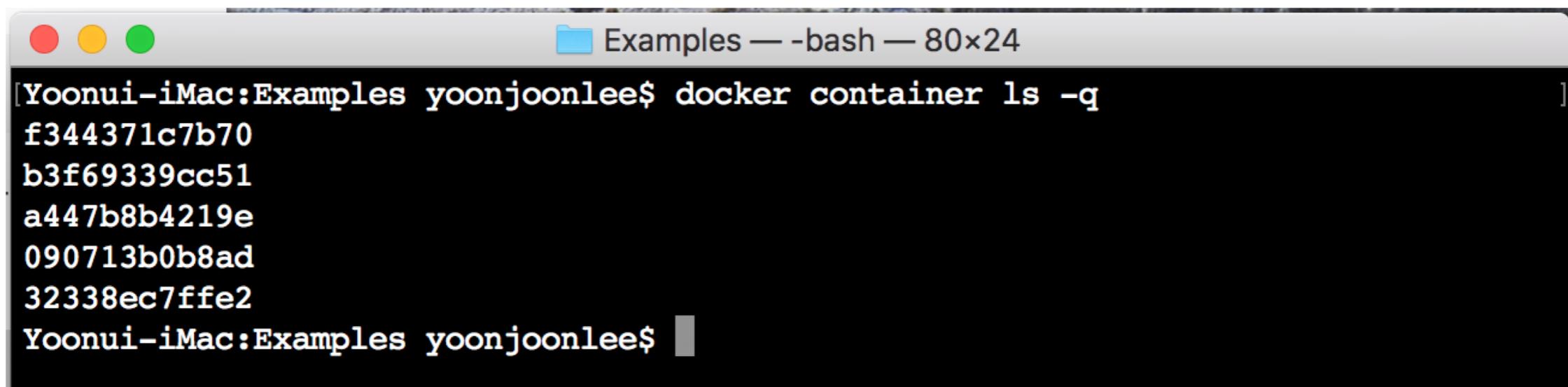
```
docker service ls
```

```
docker service ps getstartedlab_web
```

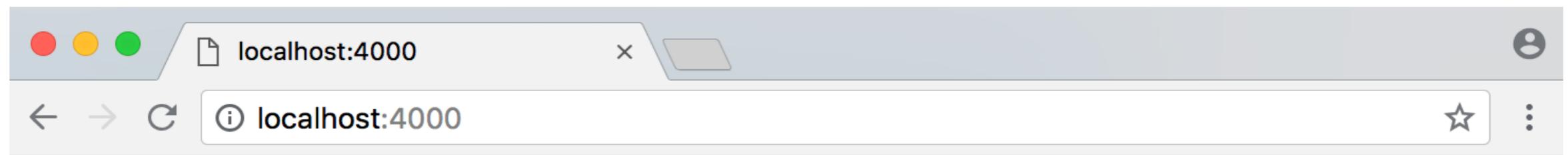
```
docker container ls -q
```



`docker container ls -q`



```
[Yoonui-iMac:Examples yoonjoonlee$ docker container ls -q
f344371c7b70
b3f69339cc51
a447b8b4219e
090713b0b8ad
32338ec7ffe2
Yoonui-iMac:Examples yoonjoonlee$ ]
```



Hello World!

Hostname: 70200c29c569

Visits: *cannot connect to Redis, counter disabled*

Scale out

```
version: "3"
services:
  web:
    # replace username/repo:tag with your name and image details
    image: yjbenlee/get-started:containers
    deploy:
      replicas: 7
    resources:
      limits:
        cpus: "0.1"
        memory: 50M
      restart_policy:
        condition: on-failure
    ports:
      - "4000:80"
  networks:
    - webnet
networks:
  webnet:
```

Scale out

```
version: "3"
services:
  web:
    # replace username/repo:tag with your name and image details
    image: yjbenlee/get-started:containers
    deploy:
      replicas: 7
      resources:
        limits:
          cpus: "0.1"
          memory: 50M
      restart_policy:
        condition: on-failure
    ports:
      - "4000:80"
  networks:
    - webnet
networks:
  webnet:
    driver: bridge

docker stack deploy --compose-file=docker-compose.yml
getstartedlab
```

Scale out

```
version: "3"
services:
  web:
    # replace username/repo:tag with your name and image details
    image: yjbenlee/get-started:containers
    deploy:
      replicas: 7
      resources:
        limits:
          cpus: "0.1"
          memory: 50M
      restart_policy:
        condition: on-failure
    ports:
      - "4000:80"
  networks:
    - webnet
networks:
  webnet:
    driver: bridge

docker stack deploy --compose-file=docker-compose.yml
getstartedlab

docker container ls -q
```

Take down the app

```
docker stack rm getstartedlab
```

Take down the app

```
docker stack rm getstartedlab
```

```
docker container ls -q
```

```
docker service ps getstartedlab_web
```

Take down the app

```
docker stack rm getstartedlab
```

```
docker container ls -q
```

```
docker service ps getstartedlab_web
```

```
docker swarm leave --force
```

Take down the app

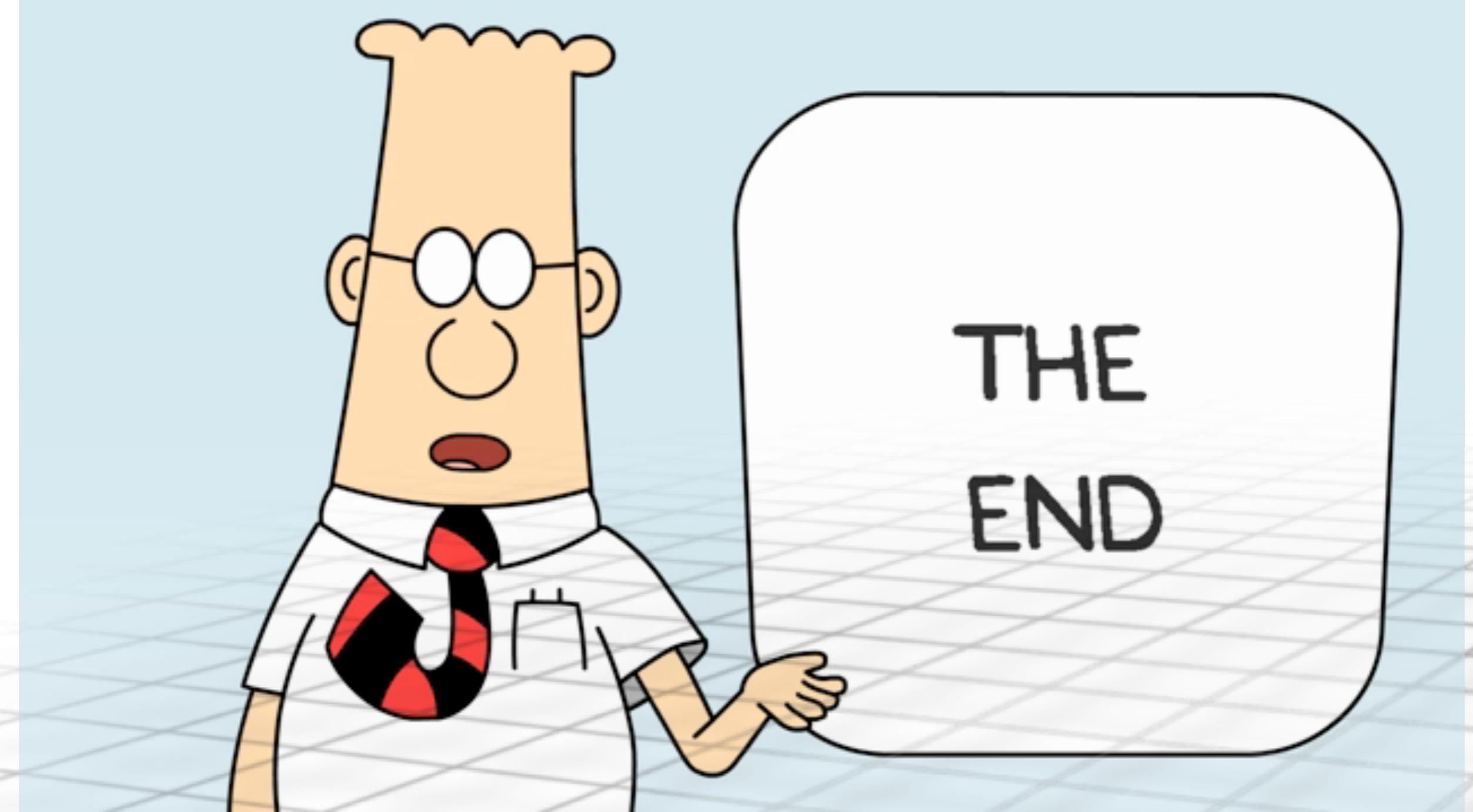
```
docker stack rm getstartedlab
```

```
docker container ls -q
```

```
docker service ps getstartedlab_web
```

```
docker swarm leave --force
```

```
docker service ps getstartedlab_web
```



감사합니다

출처: metachannels.com