

# Docker

- VM 위에서 실행 -> 환경 지옥에서 탈출,
  - “ Java를 생각하라! ”
- Process

## 용어

- image : class
- container : instance

# Docker 실행

```
% docker run hello-world
```

```
% docker ps -a
```

```
# docker run --help
```

```
% docker run -it ubuntu bash
```

```
# tomcat
```

```
% docker run -d -p 8080:8080 tomcat:8.0
```

```
# MySQL
```

```
% docker run -e MYSQL_ROOT_PASSWORD='Passw0wd' \  
--name mydb -d mysql:5.6
```

```
# Nginx, with conf file
```

```
% docker run -d -p 80:80 --name myweb \  
-v /Users/1001028/nginx.conf:/etc/nginx/nginx.conf nginx
```

# Docker Hub

```
# Docker Hub  
% docker search whalesay  
% docker pull docker/whalesay  
% docker images  
  
# 실행  
% docker run docker/whalesay cowsay boo  
% docker ps -a
```

# Docker 깨끗이 정리(삭제)

```
# 실행중인 docker 중지
% docker stop $(docker ps -a -q)

# 모든 컨테이너 삭제
% docker rm $(docker ps -a -q)

# 모든 이미지 삭제
% docker rmi $(docker images -q)
```

# Docker 이미지 제작

```
% cat Dockerfile
FROM docker/whalesay:latest
RUN apt-get -y update && apt-get install -y fortune
CMD /usr/games/fortune -a | cowsay

# build
% docker build -t docker-whale .

# run
% docker run docker-whale
```

- [jenkinsci/jenkins: Jenkins automation server](#)

# Docker 로그/정보

```
% docker logs myweb -f  
% docker inspect myweb
```

# docker-compose

```
% cat my-docker-compose.yml
myweb:
  image:nginx:latest
  ports:
    -80:80
  volumes:
    -/Users/1001028/nginx.conf:/etc/nginx/nginx.conf
...
mywas:
  image:tomcat:8.0
...
mydb:
  image:mysql:5.6
...
% docker-compose -f my-docker-compose.yml up -d
```

# Docker Hub #2

```
% docker build -t docker-whale .  
% docker login -e 이메일 -u 아이디 -p 암호  
% docker tag docker-whale mollra/docker-whale:${TAG}  
% docker push mollra/docker-whale:${TAG}
```

# Docker Swarm

# Kubernetes

- [Q: Docker Swarm, Kubernetes 및 Amazon ECS의 차이점은 무엇입니까?](#)
- [튜토리얼 - Kubernetes](#)