

# Spring Cloud로 개발하는 マイクロ서비스 アプリケイ션



Microservices

+



Spring  
Cloud

```
class Book {
    def self, title, price, author;
    self.title = title
    self.price = price
    self.author = author
}

public static void main(String[] args)
{
    var fs = require('fs');
    fs.readFile('/JONE.txt' /* 1 */,
        function (err, data) {
            console.log(data); // 3
        });
}

<@interface NextInnovationDelegate : NSObject <UIApplicationDelegate> >

<@implementation NextInnovationDelegate >
<@end>
```



# 목차



- Section 9: 암호화 처리를 위한 Encryption과 Decryption
- Section 10: 마이크로서비스간 통신
- Section 11: 데이터 동기화를 위한 Kafka 활용 ①
- Section 12: 데이터 동기화를 위한 Kafka 활용 ②
- Section 13: 장애 처리와 Microservice 분산 추적
- Section 14: Microservice 모니터링
- **Section 15: 애플리케이션 배포를 위한 컨테이너 가상화**
- Section 16: 애플리케이션 배포 – Docker Container
- Appendix: Microservice 패턴

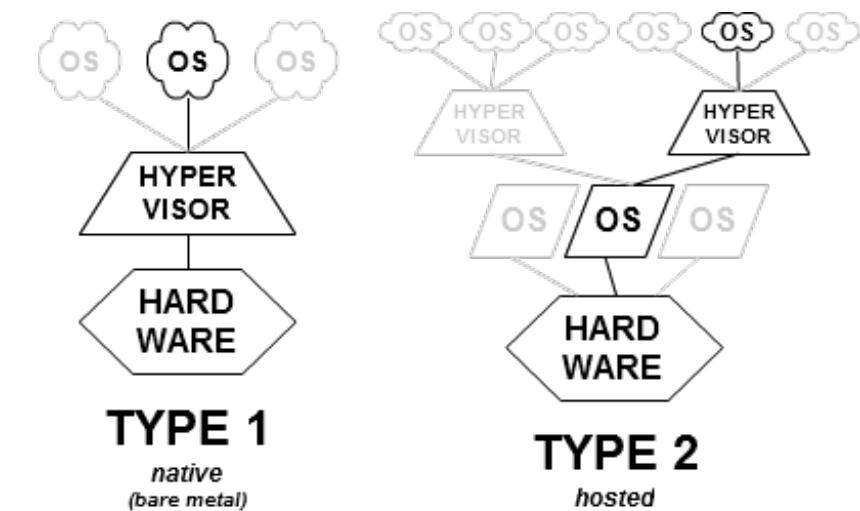
# Section 15.

## 애플리케이션 배포를 위한 컨테이너 가상화

- Container Virtualization
- Docker Desktop
- Docker 실행

# Virtualization

- 물리적인 컴퓨터 리소스를 다른 시스템이나 애플리케이션에서 사용할 수 있도록 제공
  - 플랫폼 가상화
  - 리소스 가상화
- 하이퍼바이저 (Hypervisor)
  - Virtual Machine Manager (VMM)
  - 다수의 운영체제를 동시에 실행하기 위한 논리적 플랫폼
  - Type 1: Native or Bare-metal
  - Type 2: Hosted



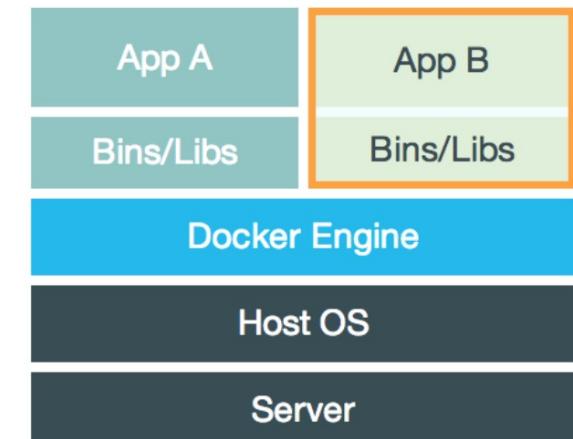
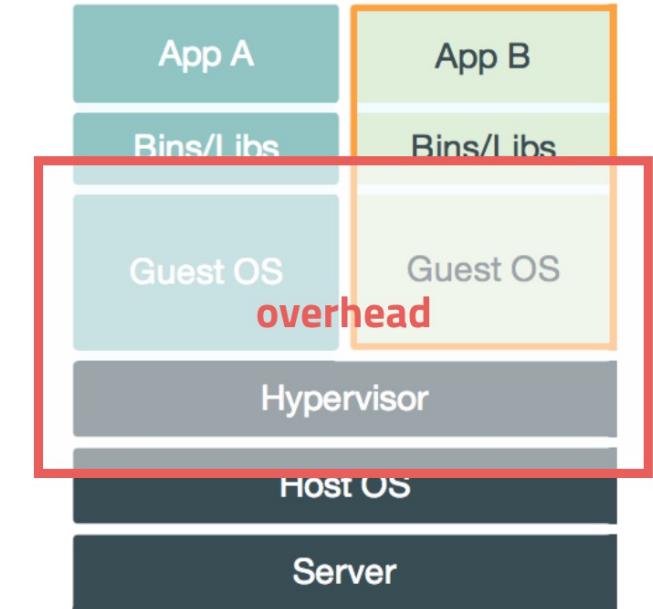
# Container Virtualization

## ■ OS Virtualization

- Host OS 위에 Guest OS 전체를 가상화
- VMWare, VirtualBox
- 자유도가 높으나, 시스템에 부하가 많고 느려짐

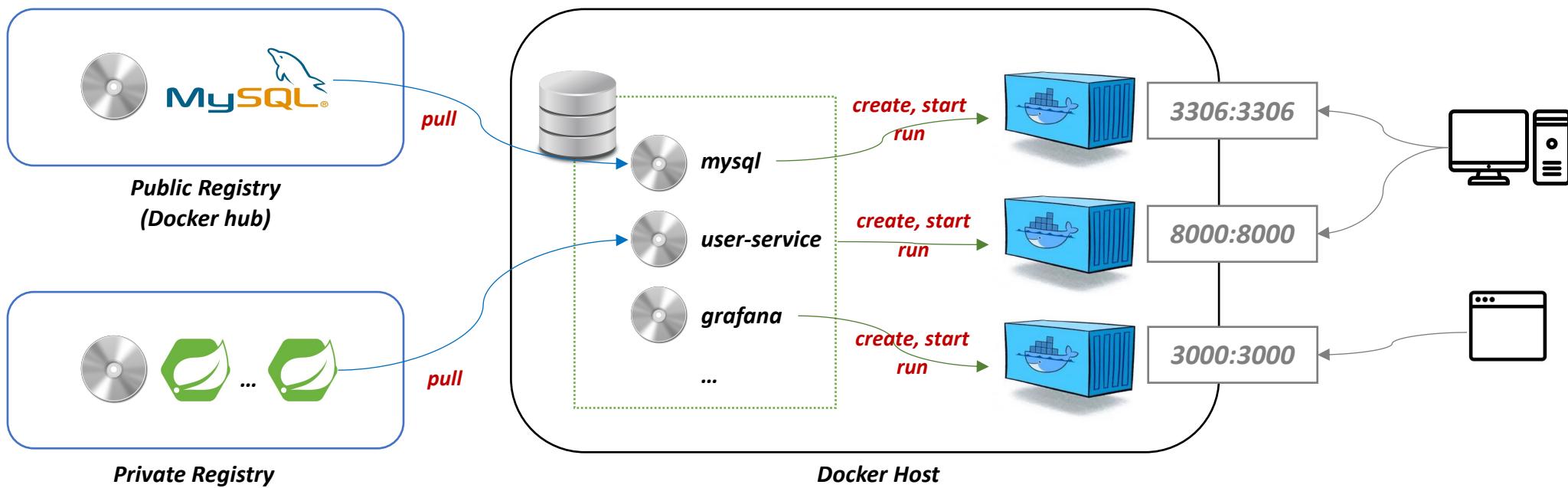
## ■ Container Virtualization

- Host OS가 가진 리소스를 적게 사용하며, 필요한 프로세스 실행
- 최소한의 라이브러리와 도구만 포함
- Container의 생성 속도 빠름



# Container Image

- Container 실행에 필요한 설정 값
  - 상태값, *Immutable*
- Image를 가지고 실체화 → Container





# Dockerfile

- Docker Image를 생성하기 위한 스크립트 파일
- 자체 DSL(Domain-Specific language) 언어 사용 → 이미지 생성과정 기술

```
FROM mysql:5.7

ENV MYSQL_ALLOW_EMPTY_PASSWORD true
ENV MYSQL_DATABASE mydb

ADD ./db_mount /var/lib/mysql

EXPOSE 3306

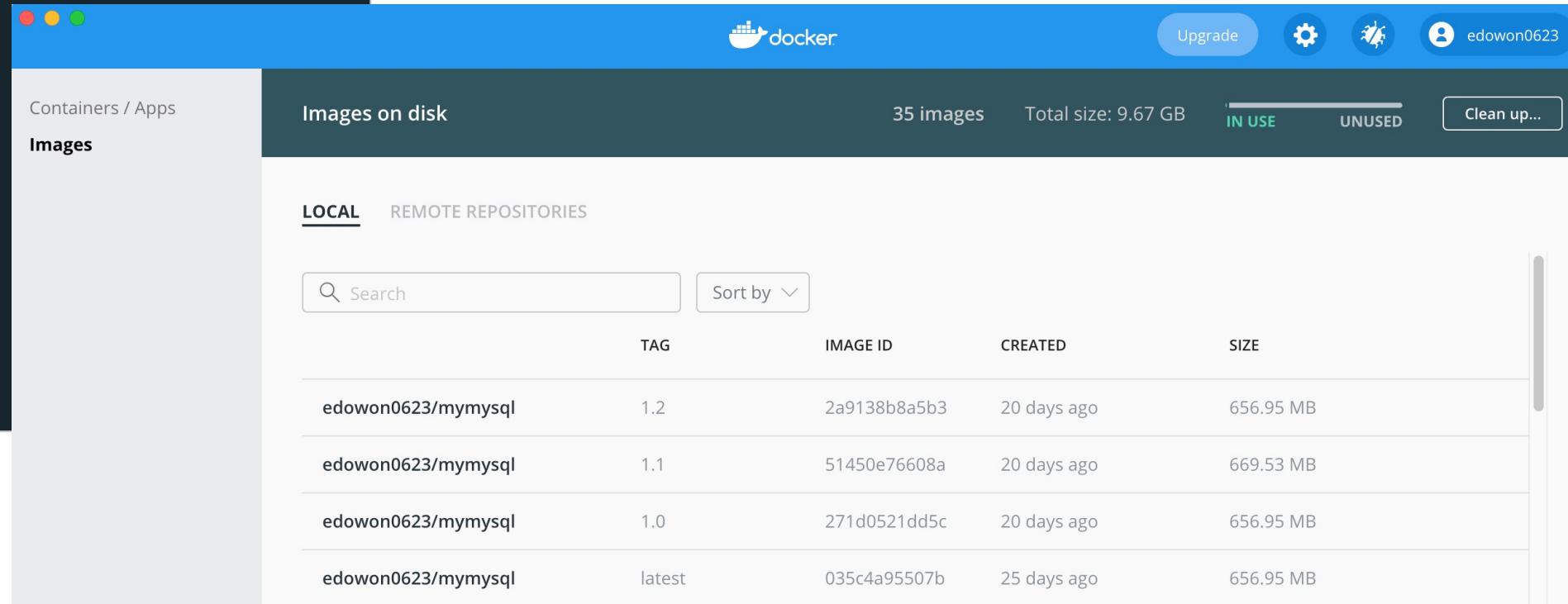
CMD ["mysqld"]
```

# Docker Desktop

- <https://www.docker.com/products/docker-desktop>

```
▶ docker info
Client:
  Context:    default
  Debug Mode: false
  Plugins:
    app: Docker App (Docker Inc., v0.9.1-beta3)
    buildx: Build with BuildKit (Docker Inc., v0.5.1-docker)
    scan: Docker Scan (Docker Inc., v0.5.0)

Server:
  Containers: 1
    Running: 1
    Paused: 0
    Stopped: 0
  Images: 108
  Server Version: 20.10.2
  Storage Driver: overlay2
    Backing Filesystem: extfs
    Supports d_type: true
    Native Overlay Diff: true
  Logging Driver: json-file
  Cgroup Driver: cgroupfs
  Cgroup Version: 1
```



# Docker 실행

## ■ 컨테이너 실행

\$ **docker run [OPTIONS] IMAGE[:TAG]@DIGEST [COMMAND] [ARG...]**

옵션	설명
-d	detached mode 흔히 말하는 백그라운드 모드
-p	호스트와 컨테이너의 포트를 연결 (포워딩)
-v	호스트와 컨테이너의 디렉토리를 연결 (마운트)
-e	컨테이너 내에서 사용할 환경변수 설정
-name	컨테이너 이름 설정
-rm	프로세스 종료시 컨테이너 자동 제거
-it	-i와 -t를 동시에 사용한 것으로 터미널 입력을 위한 옵션
-link	컨테이너 연결 [컨테이너명:별칭]

\$ **docker run ubuntu:16.04**

```
[admin@centos7 ~]$ docker run ubuntu:16.04
Unable to find image 'ubuntu:16.04' locally
16.04: Pulling from library/ubuntu
35b42117c431: Pull complete
ad9c569a8d98: Pull complete
293b44f45162: Pull complete
0c175077525d: Pull complete
Digest: sha256:a4d8e674ee993e5ec88823391de828a5e
Status: Downloaded newer image for ubuntu:16.04
```

# Docker 실행

```
$ docker run -d -p 3306:3306 -e MYSQL_ALLOW_EMPTY_PASSWORD=true --name mysql mysql:5.7
```

```
bcadmin@hlf03:~$ docker run -d -p 3306:3306 -e MYSQL_ALLOW_EMPTY_PASSWORD=true --name mysql mysql:5.7
Unable to find image 'mysql:5.7' locally
5.7: Pulling from library/mysql
f17d81b4b692: Pull complete
c691115e6ae9: Pull complete
41544cb19235: Pull complete
254d04f5f66d: Pull complete
4fe240edfdc9: Pull complete
0cd4fcc94b67: Pull complete
8df36ec4b34a: Pull complete
b8edeb9ec9e2: Pull complete
2b5adb9b92bf: Pull complete
5358eb71259b: Pull complete
e8d149f0c48f: Pull complete
Digest: sha256:42bab37eda993e417c5e7d751f1008b653c3fd8
Status: Downloaded newer image for mysql:5.7
842ff7eb6799b714050615ce505c1f96625343c0589f5d3124e2ec
```

```
bcadmin@hlf03:~$ docker exec -it mysql bash
root@842ff7eb6799:/# mysql -h127.0.0.1 -uroot
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.24 MySQL Community Server (GPL)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

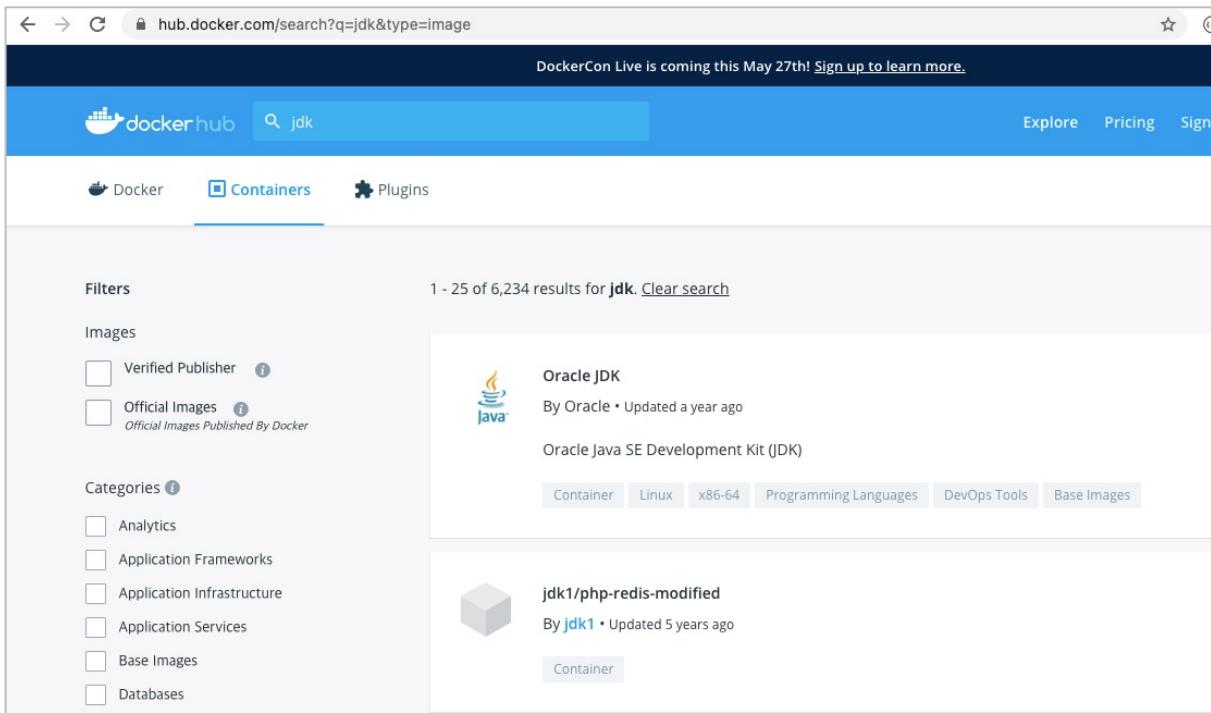
```
$ docker exec -it mysql bash
```

```
mysql> show databases;
+--------------------+
| Database          |
+----+-----+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+----+-----+
4 rows in set (0.00 sec)

mysql>
```

# Dockerfile for Users Microservice

- <https://hub.docker.com>



```
FROM openjdk:8-jdk-alpine

VOLUME /tmp

COPY target/users-ws-0.1.jar users-service.jar

ENTRYPOINT ["java",
"-Djava.security.egd=file:/dev/./urandom",
"-jar",
"users-service.jar"]
```

```
$ docker build -t edowon0623/users-service:1.0 .
```

```
$ docker push edowon0623/user-service:1.0
```

```
$ docker pull edowon0623/user-service:1.0
```