MSA 18주차

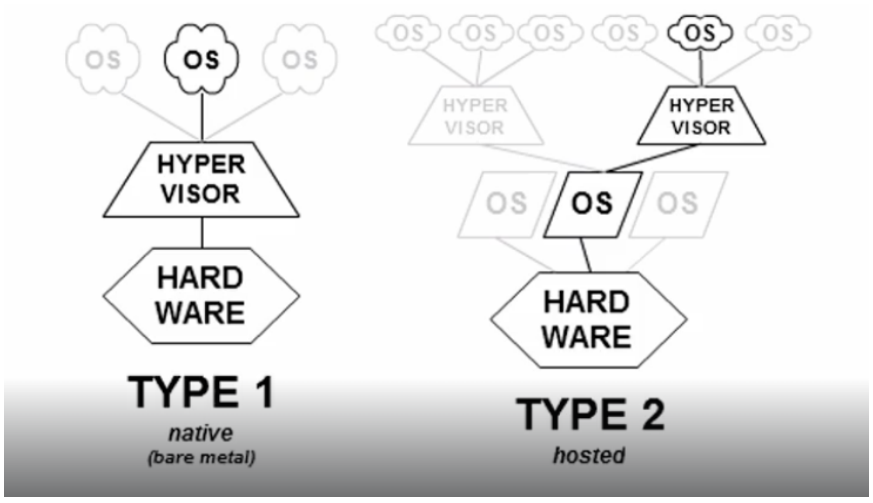
스터디원	소현
Status	Done
URL	https://github.com/SPRING-STUDY-2023/sohyeon-spring-cloud-msa/pull/12
비고	MSA Section15
제출 마감일	@January 10, 2024

애플리케이션 배포를 위한 컨테이너 가상화

컨테이너 가상화

가상화(Virtualization)

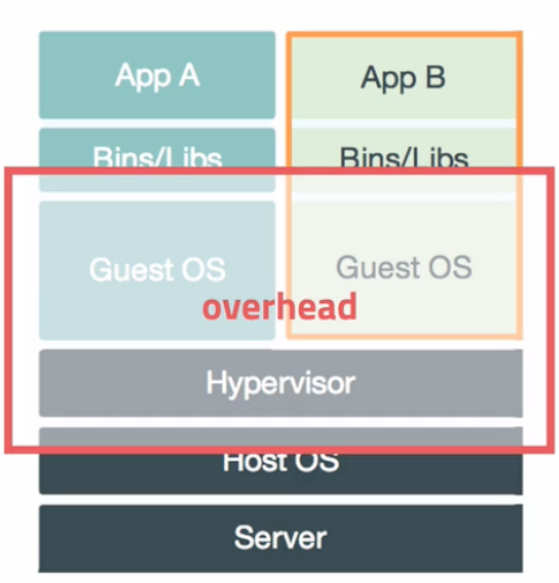
- 물리적인 컴퓨터 리소스를 다른 시스템이나 애플리케이션에서 사용할 수 있도록 제공
  - 플랫폼 가상화
  - 리소스 가상화
- 하이퍼바이저(Hypervisor)
  - Virtual Machine Manager(VMM)
  - 다수의 운영체제를 동시에 실행하기 위한 논리적 플랫폼



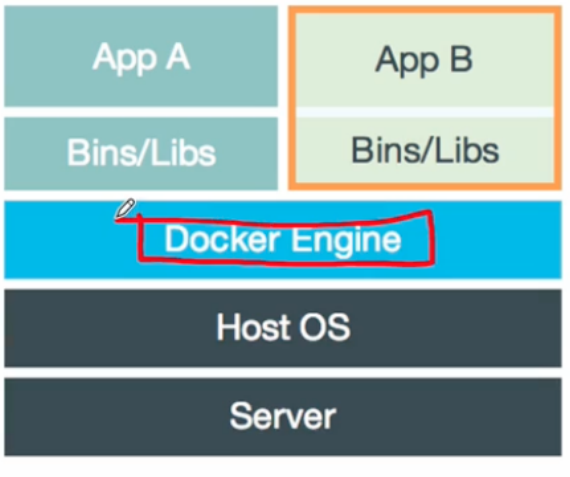
  - Type1: Native or Bare-metal
  - Type2: Hosted

Container Virtualization

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

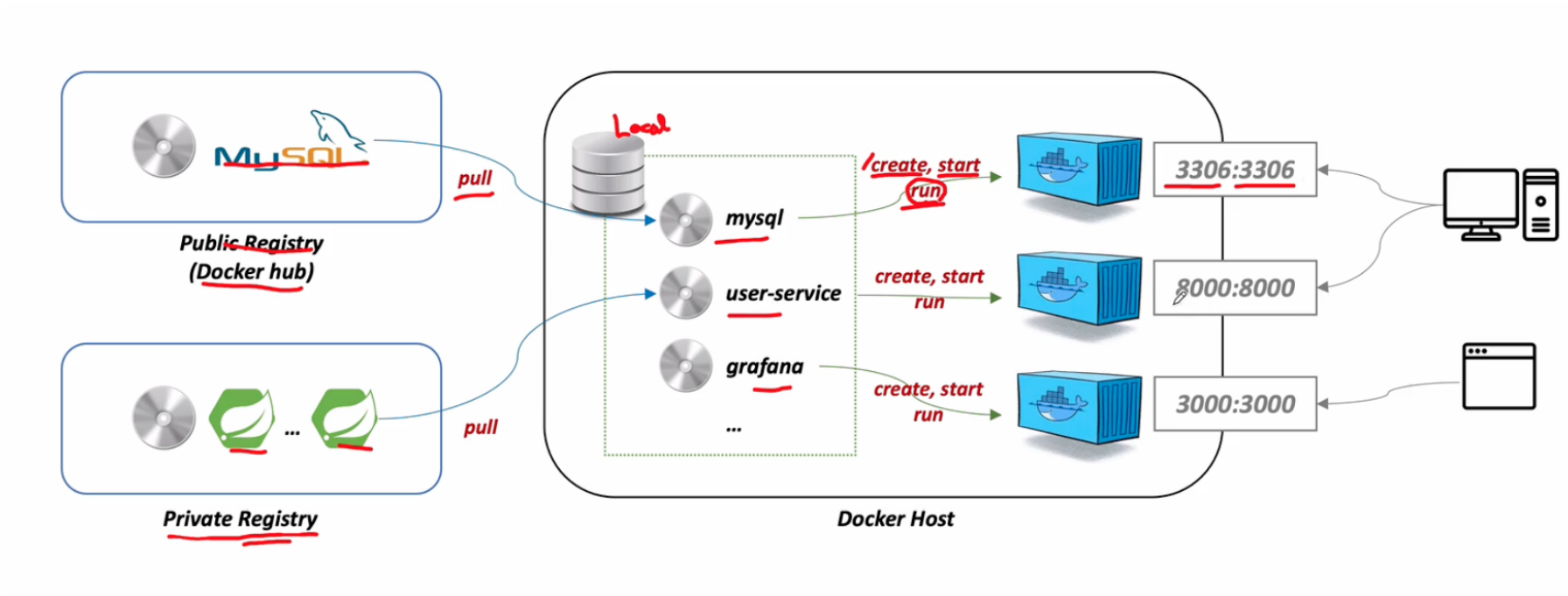


- **Container Virtualization**
  - Host OS가 가진 리소스를 적게 사용하며, 필요한 프로세스 실행
  - 최소한의 라이브러리 도구만 포함
  - Container의 생성 속도 빠름



Container Image

- Container 실행에 필요한 설정 값
  - 상태 값 X, Immutable
- Image를 가지고 실체화 → Container
  - Public Registry(Docker hub)
  - Private Registry



Dockerfile

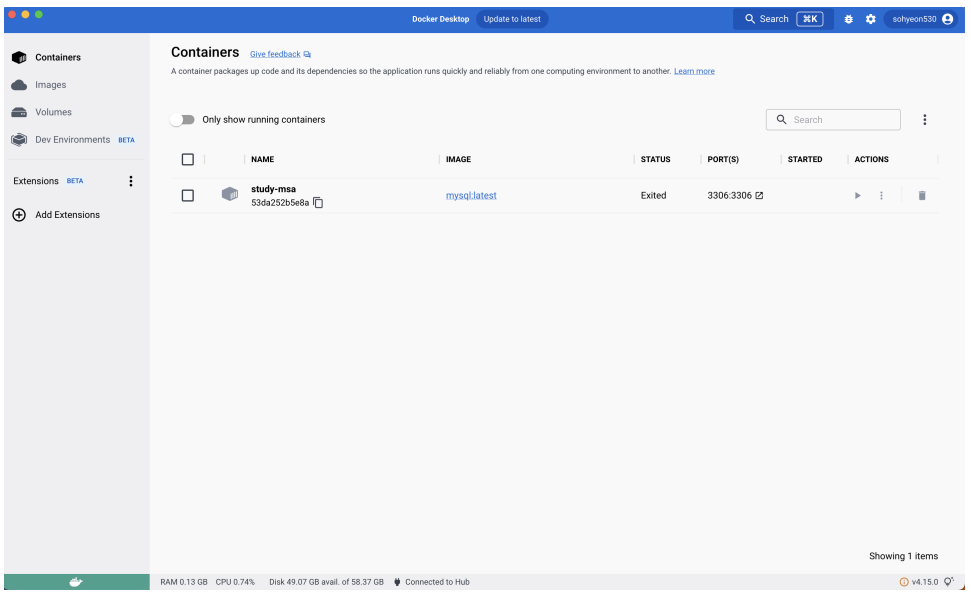
- Docker Image를 생성하기 위한 스크립트 파일
- 자체 DSL(Domain-Specific Language) 언어 사용 → 이미지 생성 과정 기술

```
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 컨테이너

Docker Desktop

- **설치**
  - https://www.docker.com/products/docker-desktop
  - 회원가입, 다운로드 후 Docker 대시보드 접속



- **명령어**

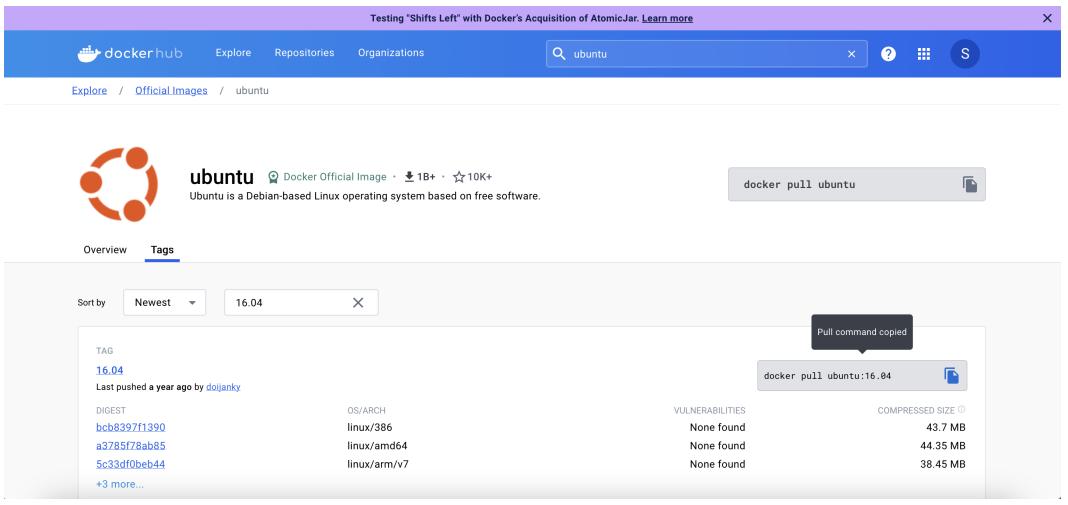
```
$ docker info # 도커 정보 확인
$ docker image ls # 생성된 이미지 확인
$ docker container ls # 실행 중인 컨테이너 확인
```

- **컨테이너 실행**

```
$ docker run [OPTIONS] IMAGE[:TAG|@DIGEST][COMMAND][ARG...] # 실행 포맷
$ docker run ubuntu:16.04 # 실행 예시
```

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 hub**



  - 이미지 검색, 태그 검색, 실행 명령어 복사 가능

- **우분투 이미지 실행 예시**

```
$ docker pull ubuntu:16.04 # 이미지 다운로드
$ docker image | grep ubuntu # 우분투 이미지 확인
$ docker run ubuntu:16.04 # 이미지 실행 (바로 종료)
$ docker container ls -a # 종료된 컨테이너까지 확인
$ docker container rm [CONTAINER ID] # 컨테이너 ID로 컨테이너 삭제
```

컨테이너 생성과 실행

Docker 실행

- **MySQL 이미지를 통해 컨테이너 생성 및 실행**

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

  - `-d` : 백그라운드 실행
  - `-p 3306:3306` : 포트 포워딩

    → {호스트 PC에서 접근하고자 하는 포트}:{컨테이너에서 응답하기 위한 포트}
  - `-e MYSQL_ALLOW_EMPTY_PASSWORD=true` : 루트 비밀번호 설정(비밀번호 없는 것으로 설정)
  - `--name mysql` : 컨테이너 이름 설정
  - `mysql:5.7` : 이미지 이름

- **실행 중인 컨테이너에 접속(exec) 명령어 전달(it)**

```
$ docker exec -it mysql bash # bash shell을 이용해서 컨테이너 실행
```

예시 (Mariadb 이미지 사용)

- **mariadb 컨테이너 실행 및 터미널 접속**

```
→  ~ docker run -d -p 3306:3306 -e MYSQL_ALLOW_EMPTY_PASSWORD=true --name mariadb mariadb
01d039e2e2efdc580340a2d7bc9e3630a34b3c8d28a7e57686292ad4f6578ab4
→  ~ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS                      PORTS                    NAMES
01d039e2e2ef   mariadb        "docker-entrypoint.s…"   57 seconds ago   Up 57 seconds               0.0.0.0:3306->3306/tcp   mariadb
53da252b5e8a   mysql:latest   "docker-entrypoint.s…"   3 weeks ago      Exited (0) 2 weeks ago                               study-msa
→  ~ docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED              STATUS              PORTS                    NAMES
01d039e2e2ef   mariadb   "docker-entrypoint.s…"   About a minute ago   Up About a minute   0.0.0.0:3306->3306/tcp   mariadb
```

컨테이너 실행

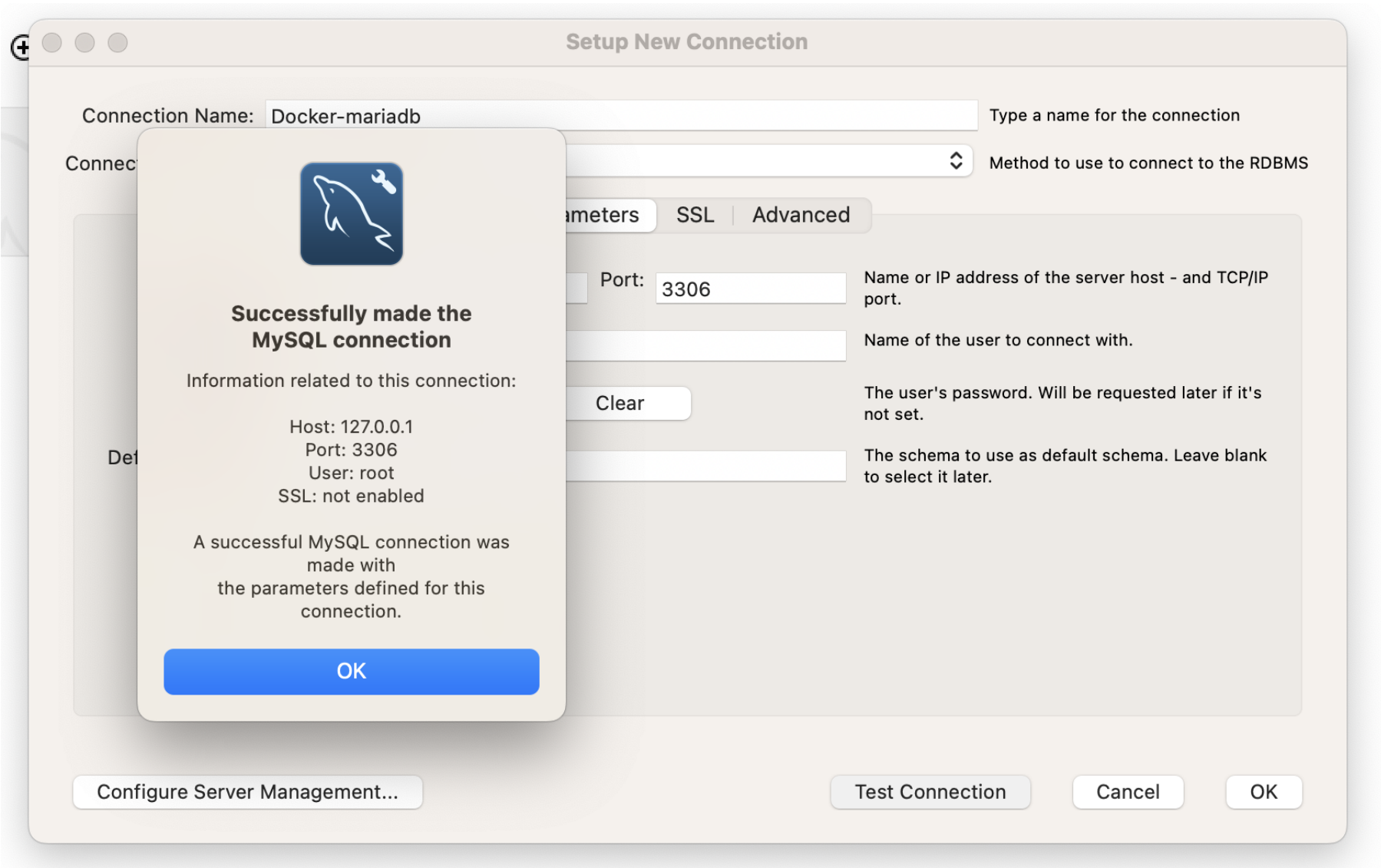
```
➜  ~ docker exec -it mariadb /bin/bash
root@01d039e2e2ef:/# mariadb -uroot -p -h127.0.0.1
Enter password:




MariaDB [(none)]>
```

컨테이너 내 MariaDB 접속

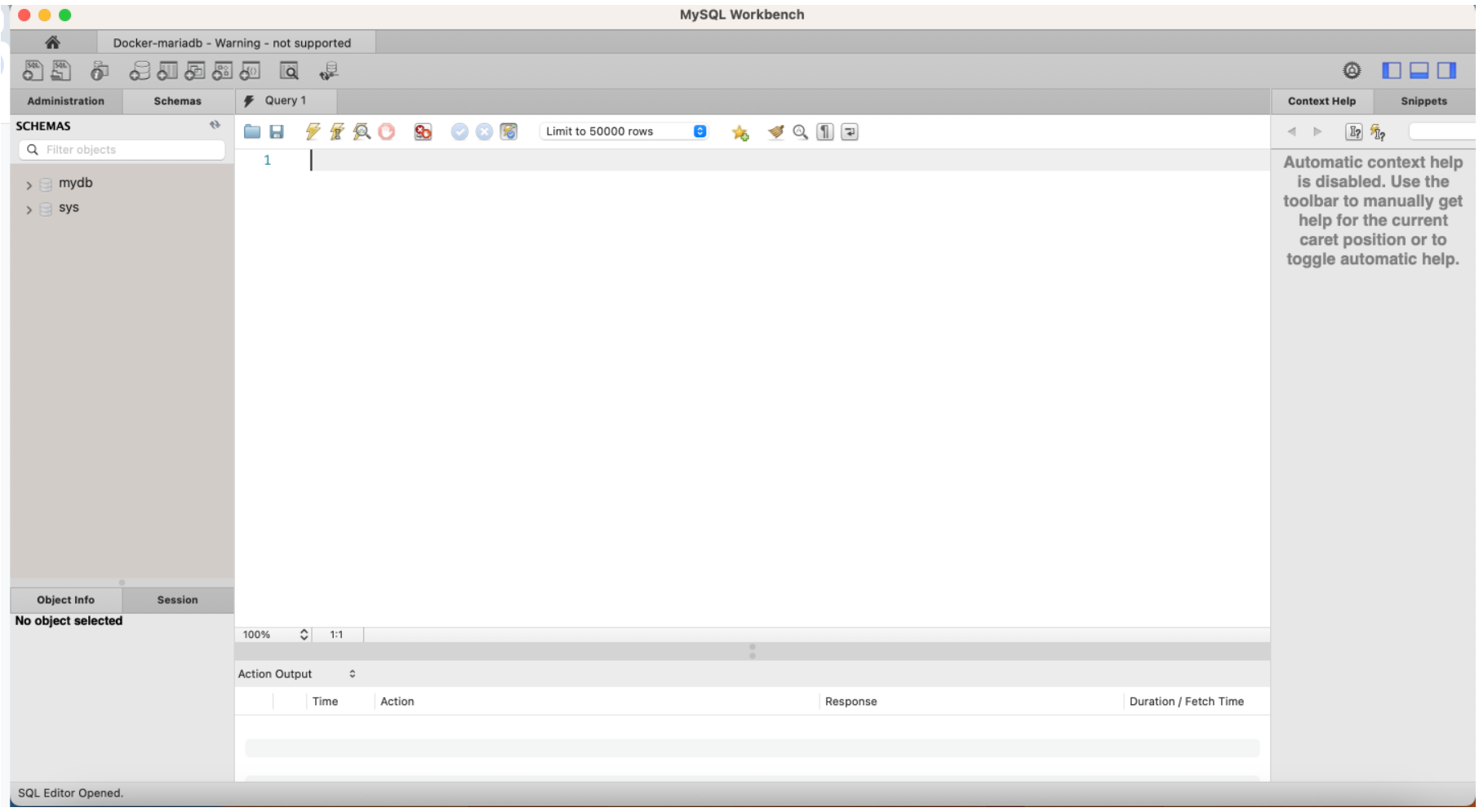
```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+


MariaDB [(none)]> clear
MariaDB [(none)]> create database mydb;


MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mydb               |
| mysql              |
| performance_schema |
| sys                |
+--------------------+


MariaDB [(none)]> exit

root@01d039e2e2ef:/# exit
exit
➜  ~
```

데이터베이스 생성

- **Host PC에서 접속**



워크벤치 접속



connection 접속

- **컨테이너 중지 및 삭제**

Docker 이미지 생성과 Public registry에 Push

Dockerfile for Users Microservice

- **Dockerfile 생성**

```
FROM openjdk:17-ea-jdk-slim
VOLUME /tmp
COPY target/user-service-1.0.jar UserService.jar
ENTRYPOINT ["java", "-jar", "UserService.jar"]
```

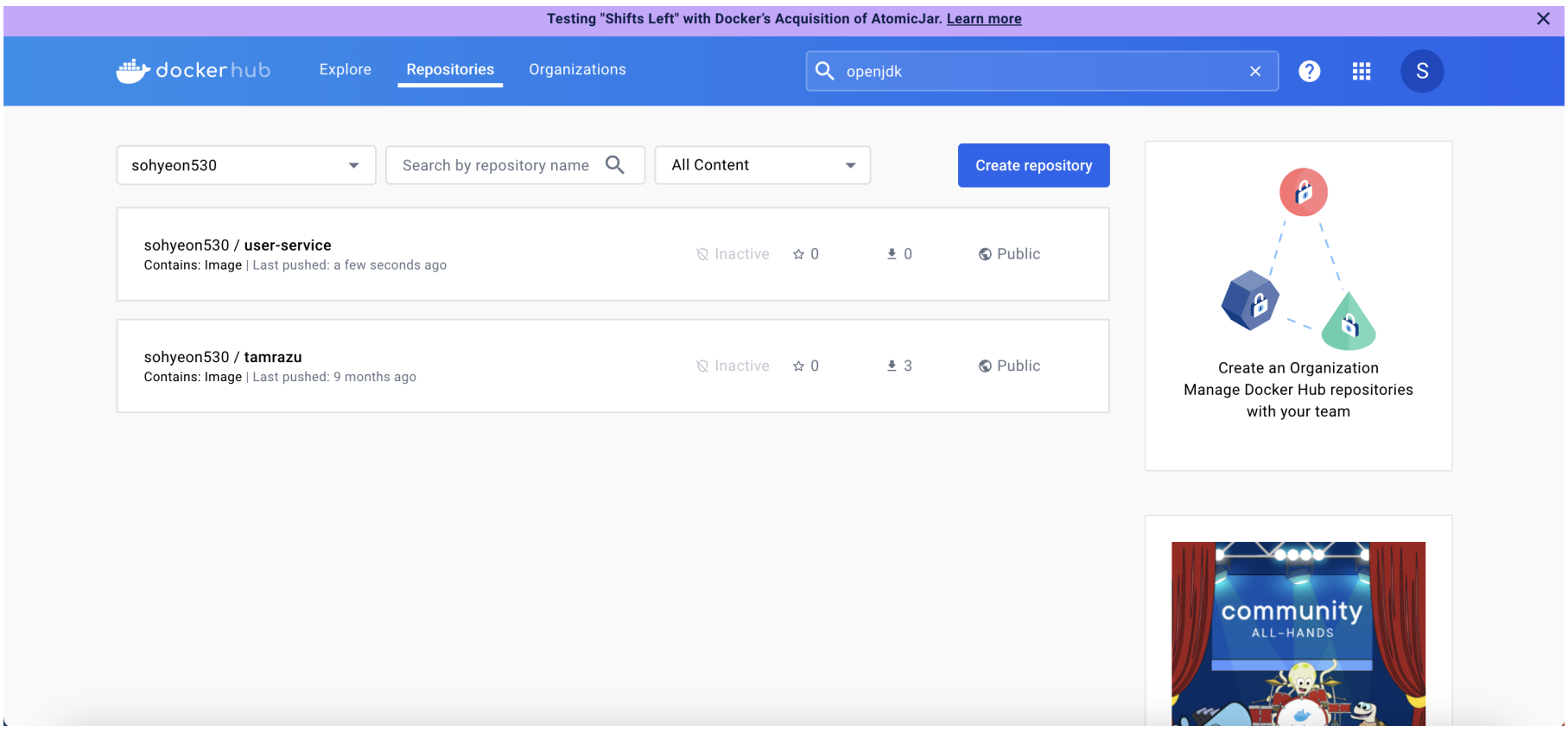
- **Clean, Compile, Build → jar 파일 생성**
- **이미지 생성 및 Docker Hub 업로드**

```
$ docker build --tag sohyeon530/user-service:1.0 . # 이미지 생성
$ docker push sohyeon530/user-service:1.0 # 이미지 업로드
$ docker pull sohyeon530/user-service:1.0 # 이미지 다운로드
$ docker run sohyeon530/user-service:1.0 # 컨테이너 생성 및 실행
```

```
➜  user-service git:(week18) ✗ docker images
REPOSITORY                    TAG             IMAGE ID       CREATED          SIZE
sohyeon530/user-service       1.0             eb6a717c96ba   48 seconds ago   501MB
mysql                         latest          1402b5eee239   3 weeks ago      641MB
mariadb                       latest          46acb2f21cf9   5 weeks ago      388MB
paketobuildpacks/run          base-cnb        f2e5000af0cb   6 months ago     87.1MB
tamrazu                       latest          ce7dccf471e6   9 months ago     443MB
<none>                        <none>          4370def48fd2   9 months ago     443MB
<none>                        <none>          e6d8342dfa1b   9 months ago     402MB
mysql                         8.0.31          7b6f3978ca29   13 months ago    550MB
postgres                      latest          d470761f9551   14 months ago    359MB
mysql                         8               5148355a8b6c   14 months ago    547MB
ubuntu                        16.04           d125c6a1fe22   14 months ago    119MB
mysql/mysql-server            8.0             55af6abb77a6   15 months ago    527MB
openjdk                       17-jdk-slim     8a3a2ffec52a   20 months ago    402MB
docker/getting-started        latest          157095baba98   21 months ago    27.4MB
pack.local/builder/ufzegapgsl latest          ef07cb028cd8   44 years ago     1.31GB
user-service                  1.0             cc3e8ae50132   44 years ago     365MB
user-service                  0.0.1-SNAPSHOT  7dd67b64c787   44 years ago     365MB
paketobuildpacks/builder      base            050ed48532b2   44 years ago     1.31GB
sohyeon530/tamrazu            latest          327d8c45ee54   54 years ago     554MB
➜  user-service git:(week18) ✗
```

user-service 이미지 생성 확인

- **업로드 된 이미지 확인**



Docker Hub에 업로드 된 user-service 이미지

```
➜  user-service git:(week18) ✗ docker pull sohyeon530/user-service:1.0
1.0: Pulling from sohyeon530/user-service
513c6bababb2b: Already exists
e464611cad28: Already exists
861123380381: Already exists
e27ae3388fb1: Already exists
Digest: sha256:860b015d081d474f301c6de23d9ecf99548a1f4feaef1ea50095f431e8552ed1
Status: Downloaded newer image for sohyeon530/user-service:1.0
docker.io/sohyeon530/user-service:1.0
```

이미지 삭제 후, 생성했던 이미지를 다운로드 가능

```
➜  user-service git:(week18) ✗ docker run sohyeon530/user-service:1.0

  .   ____          _            __ _ _
 /\\ / ___'_ __ _ _(_)_ __  __ _ \ \ \ \
( ( )\___ | '_ | '_| | '_ \/ _` | \ \ \ \
 \\/  ___)| |_)| | | | | || (_| |  ) ) ) )
  '  |____| .__|_| |_|_| |_\__, | / / / /
 =========|_|==============|___/=/_/_/_/
 :: Spring Boot ::                (v3.1.3)

2024-01-08T07:55:56.551Z  INFO [user-service,,,] 1 --- [           main] c.c.c.ConfigServicePropertySourceLocator : Fetching config from server at : http://lo
calhost:8888
```

이미지로 컨테이너 생성 및 실행