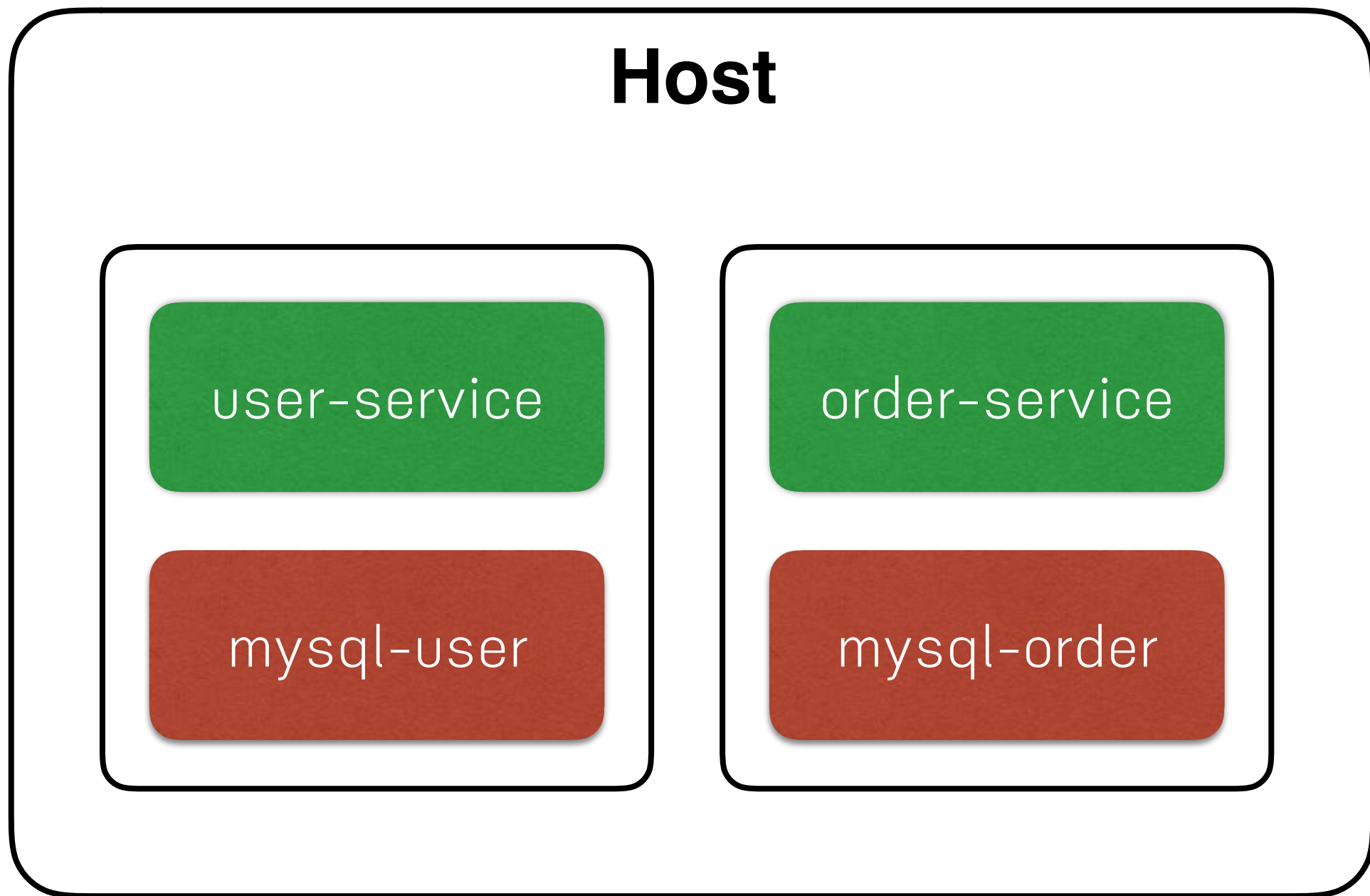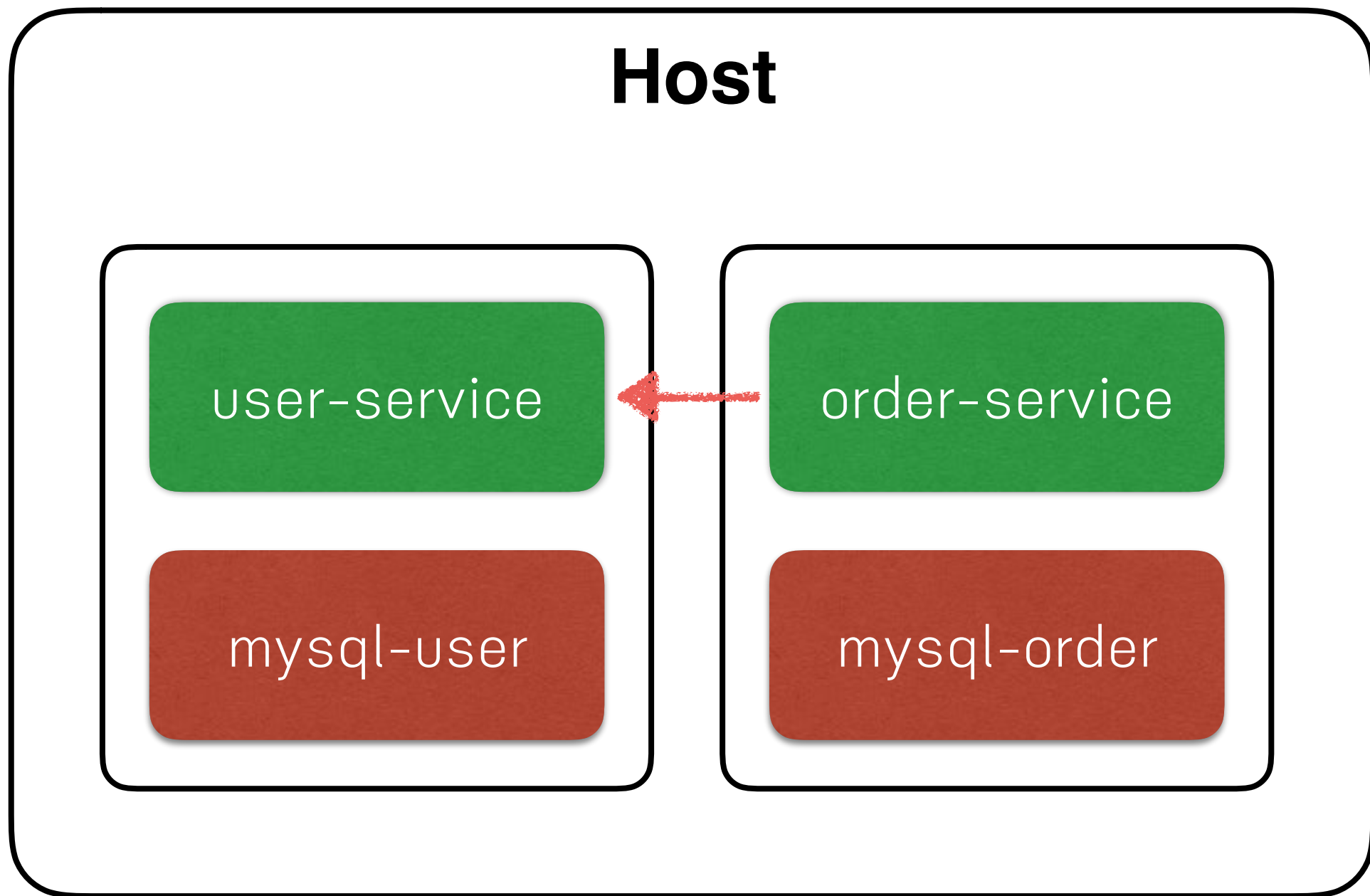# Deploy Spring boot with Docker

Spring boot application

Spring boot application

User service

1. Create database container

database = mysql-user

```
$docker container run  -d --name mysql-user  \
   -e MYSQL_ROOT_PASSWORD=123456 \
   -e MYSQL_DATABASE=user  mysql:latest
```

2. Create user–service image

Dockerfile

```
FROM frolvlad/alpine-oraclejdk8:slim
ADD  ./target/user-service.jar app.jar
ENTRYPOINT [ "sh", "-c", "java -jar app.jar" ]
```

2. Create user–service image

$docker image build –t user–service .

```
FROM frolvlad/alpine-oraclejdk8:slim
ADD  ./target/user-service.jar app.jar
ENTRYPOINT [ "sh", "-c", "java -jar app.jar" ]
```

3. Create user-service container

Link to **mysql-user** container

```
$docker container run  -d --name user-service \
    --link mysql-user \
    -p 9001:9001 user-service
```

4. Testing

http://localhost:9001/user/1


```
{
    id: 1,
    firstname: "somkiat",
    lastname: "pui"
}
```

Order service

1. Create database container

database = mysql-order

```
$docker container run  -d --name mysql-order  \
  -e MYSQL_ROOT_PASSWORD=123456 \
  -e MYSQL_DATABASE=user  mysql:latest
```

2. Create order–service image

Dockerfile

```
FROM frolvlad/alpine-oraclejdk8:slim
ADD  ./target/order-service.jar app.jar
ENTRYPOINT [ "sh", "-c", "java -jar app.jar" ]
```

2. Create order–service image

$docker image build –t order–service .

```
FROM frolvlad/alpine-oraclejdk8:slim
ADD  ./target/user-service.jar app.jar
ENTRYPOINT [ "sh", "-c", "java -jar app.jar" ]
```
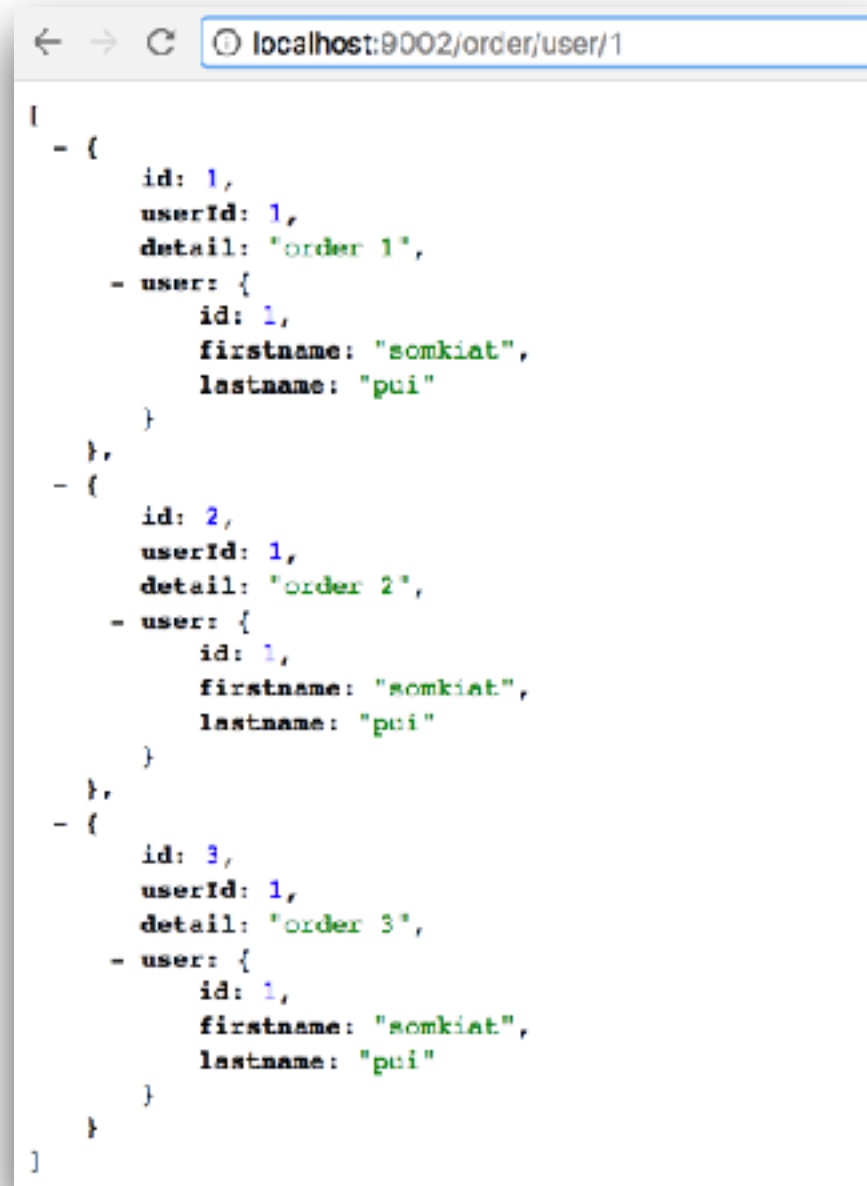
3. Create order-service container

Link to **mysql-order** and **user-service**

container

```
$docker container run  -d --name order-service \
    --link mysql-user \
    --link user-service \
    -p 9002:9001 order-service
```

4. Testing

http://localhost:9002/order/user/1