MySQL operations with Docker

A quick guide for the uninitiated

Giuseppe Maxia

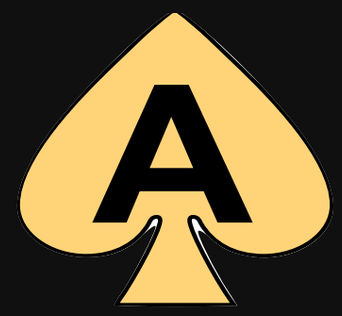
Quality Assurance Architect at VMware

@datacharmer

About me

Who’s this guy?

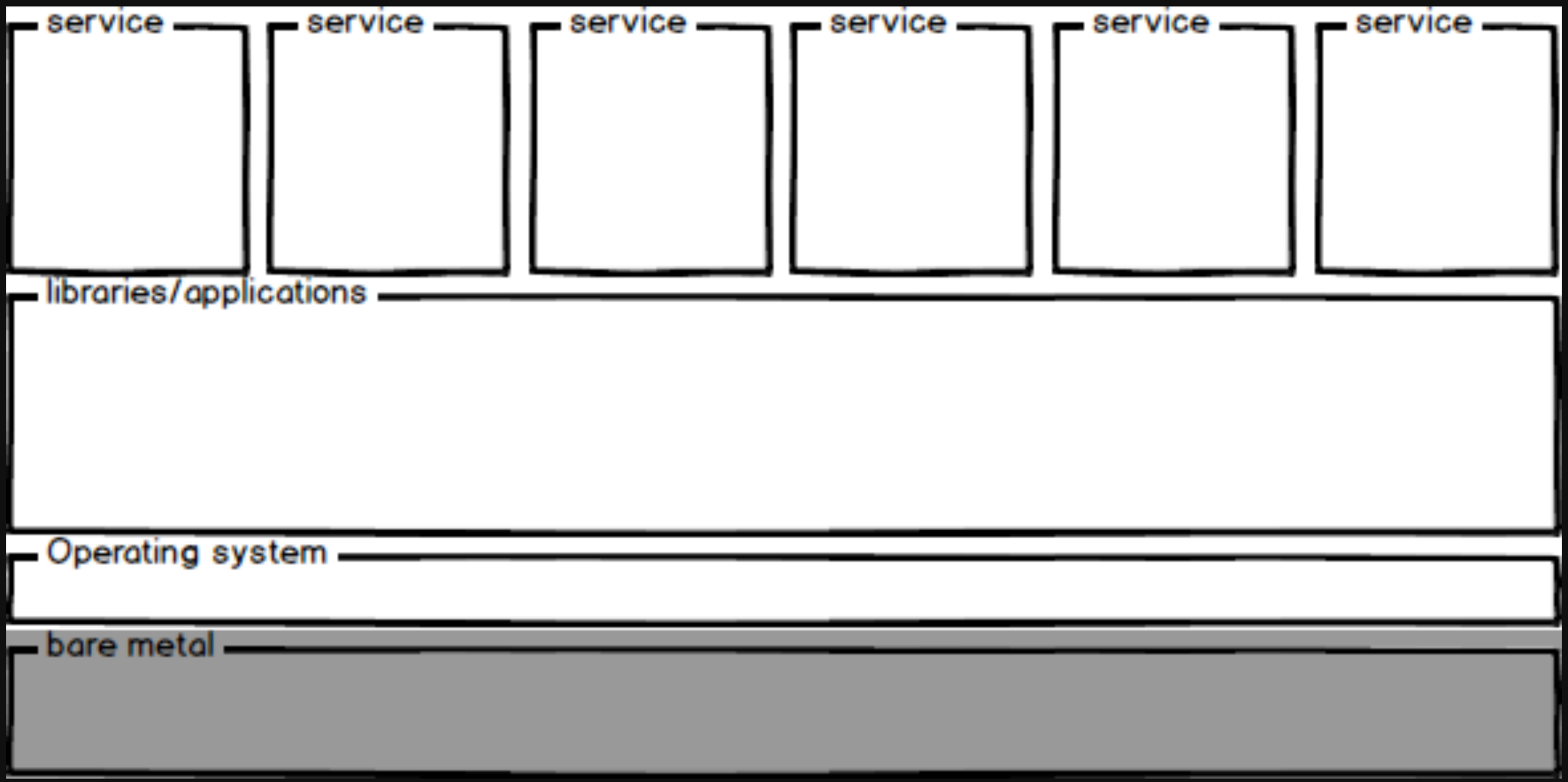
- Giuseppe Maxia, a.k.a. "The Data Charmer"
  - QA Architect at VMware
  - 25+ years development and DB experience
  - Long timer MySQL community member.
  - Oracle ACE Director
  - Blog: http://datacharmer.blogspot.com
  - Twitter: @datacharmer



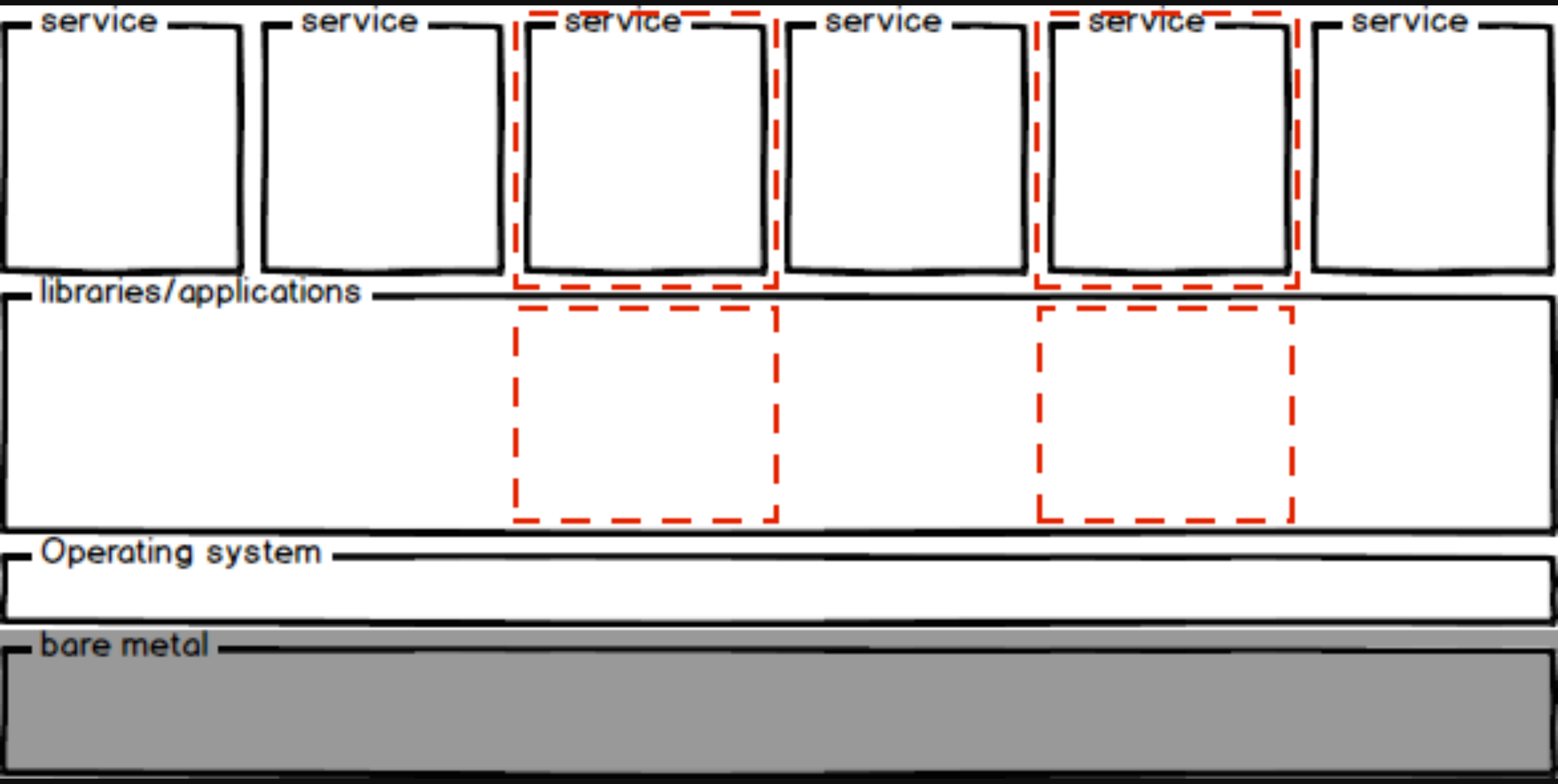
I speak for myself, not my company

- All I say here is **my** opinion
- It does not have to be the opinion of my company
- My company does not influence what I say here

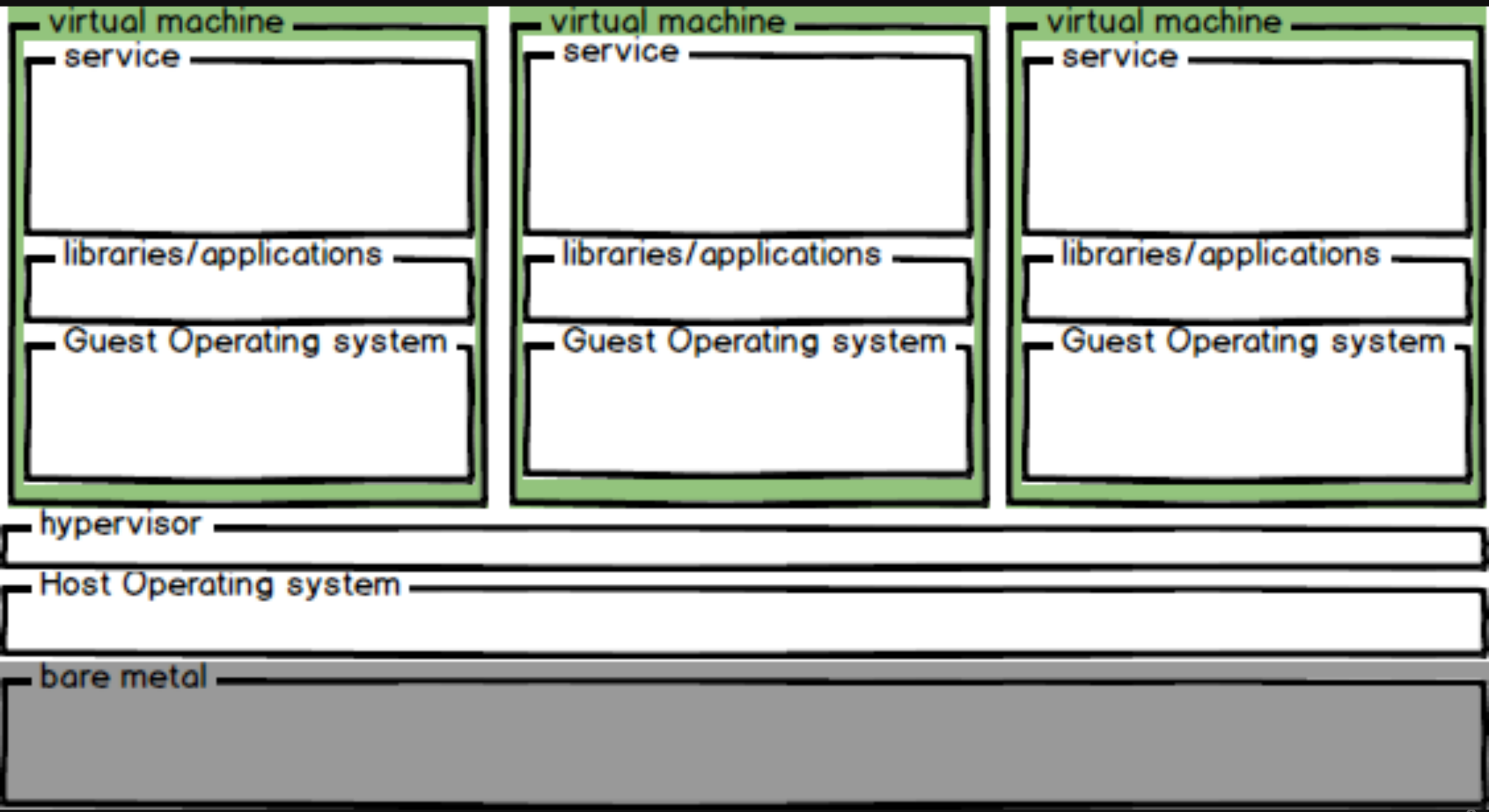
Standalone apps - standalone server



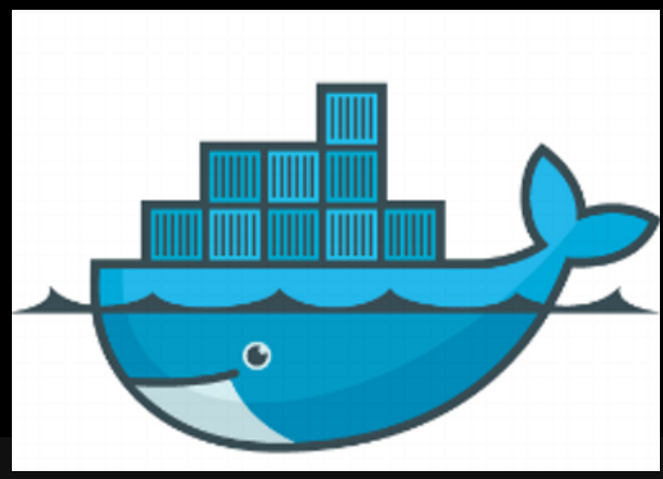
Sandboxes - apps on shared resources

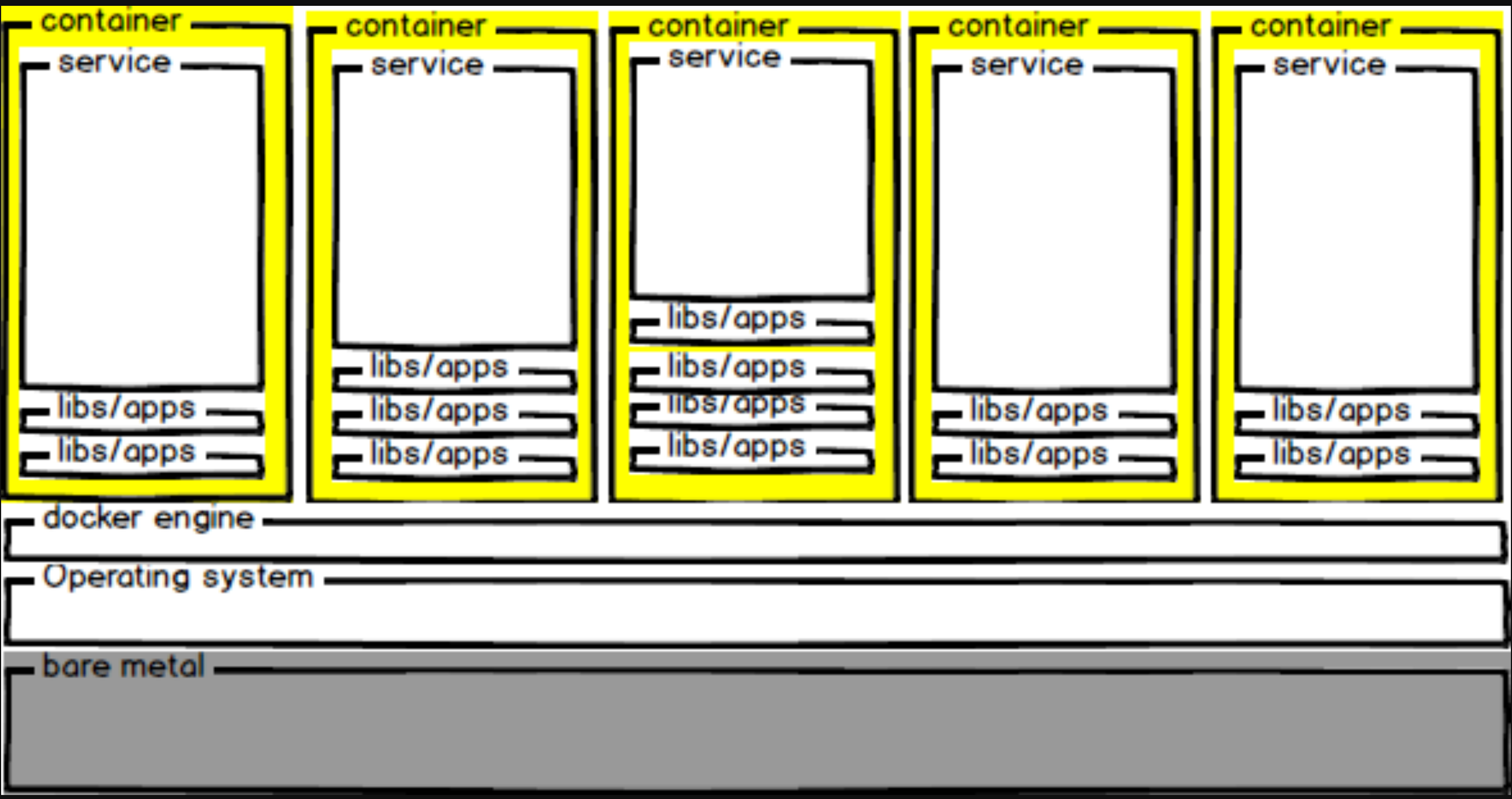


Virtual machines

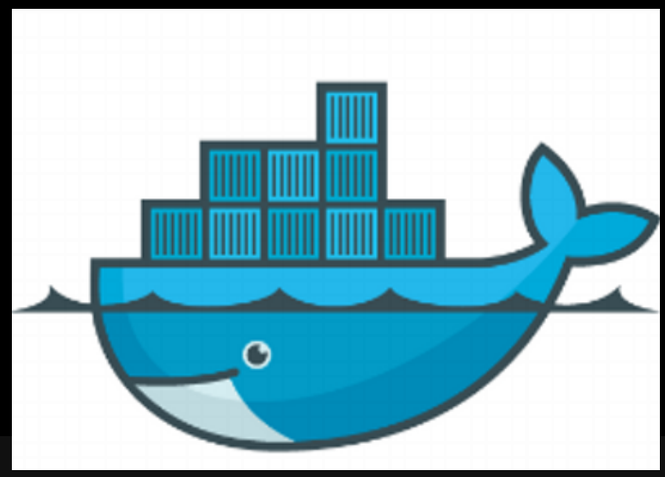


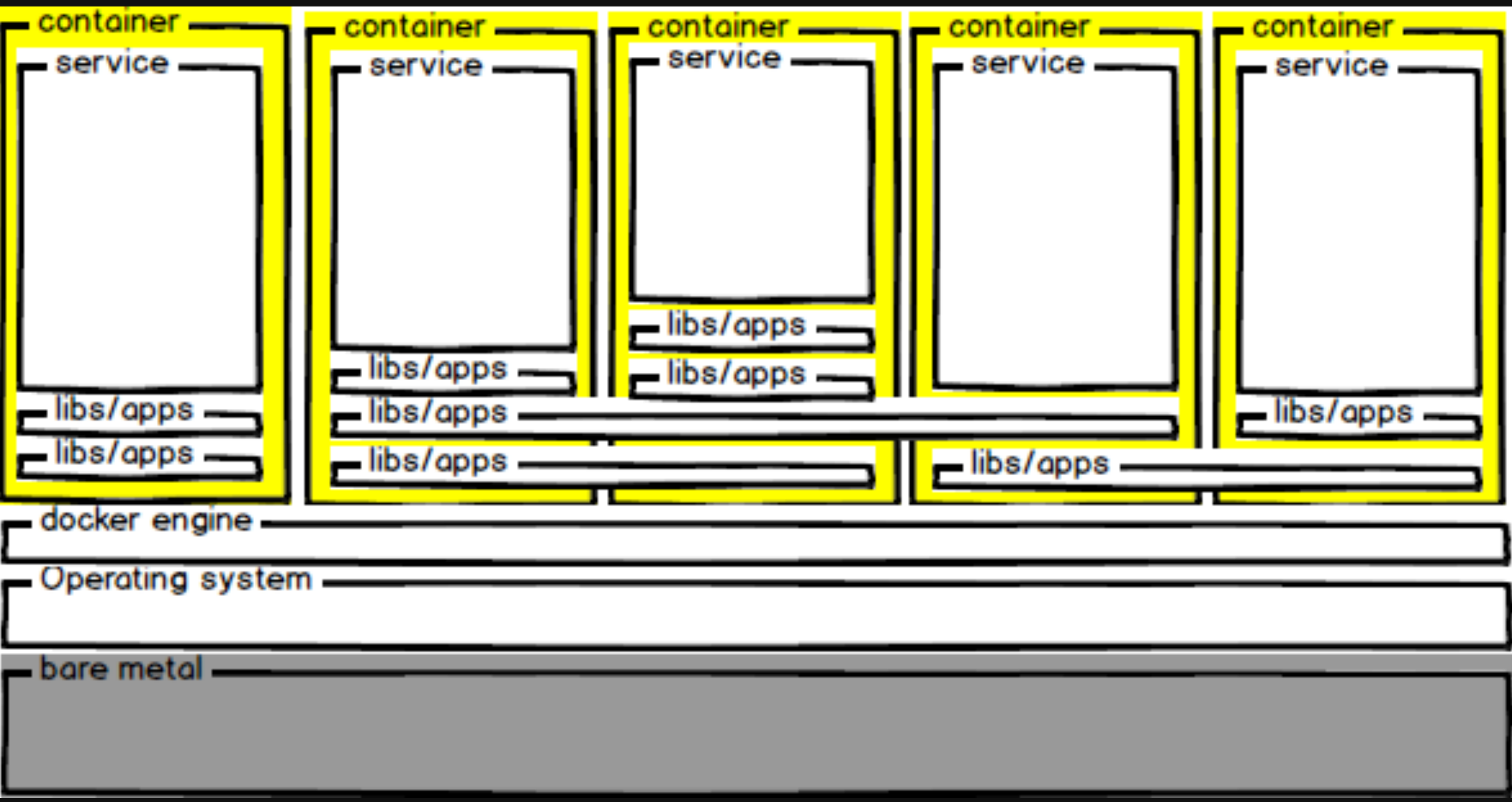
Containers




container
service
libs/apps
libs/apps
container
service
libs/apps
libs/apps
libs/apps
container
service
libs/apps
libs/apps
libs/apps
container
service
libs/apps
libs/apps
container
service
libs/apps
libs/apps
docker engine
Operating system
bare metal

Containers




container
service
libs/apps
libs/apps
container
service
libs/apps
libs/apps
container
service
libs/apps
libs/apps
container
service
libs/apps
container
service
libs/apps
docker engine
Operating system
bare metal

What is a container?

Software distribution blocks

- Virtualization system
- NOT a virtual machine
- Works close to the host operating system
- Can only run in the same O.S. as the host
- Less secure than VM
- Extremely fast to deploy
- Almost as efficient as a standalone server

But really, what is a container?

You won't understand it until you try it.

- A new way of thinking.
- Similar feeling as the first time I used a scripting language instead of C to run system tasks

Performance note:

You need to tune your expectations

- On OSX and Windows, docker runs inside a virtual machine.
- Its performance is not even remotely comparable with containers on native Linux.

Before you start

You need to install Docker. Yeah

- **On Linux**
  - `apt-get install docker`
  - or
  - `yum install docker`
  - or
  - `curl -sSL https://get.docker.com/ | sh`
- **On Mac/Windows**
  - get the Docker Machine from

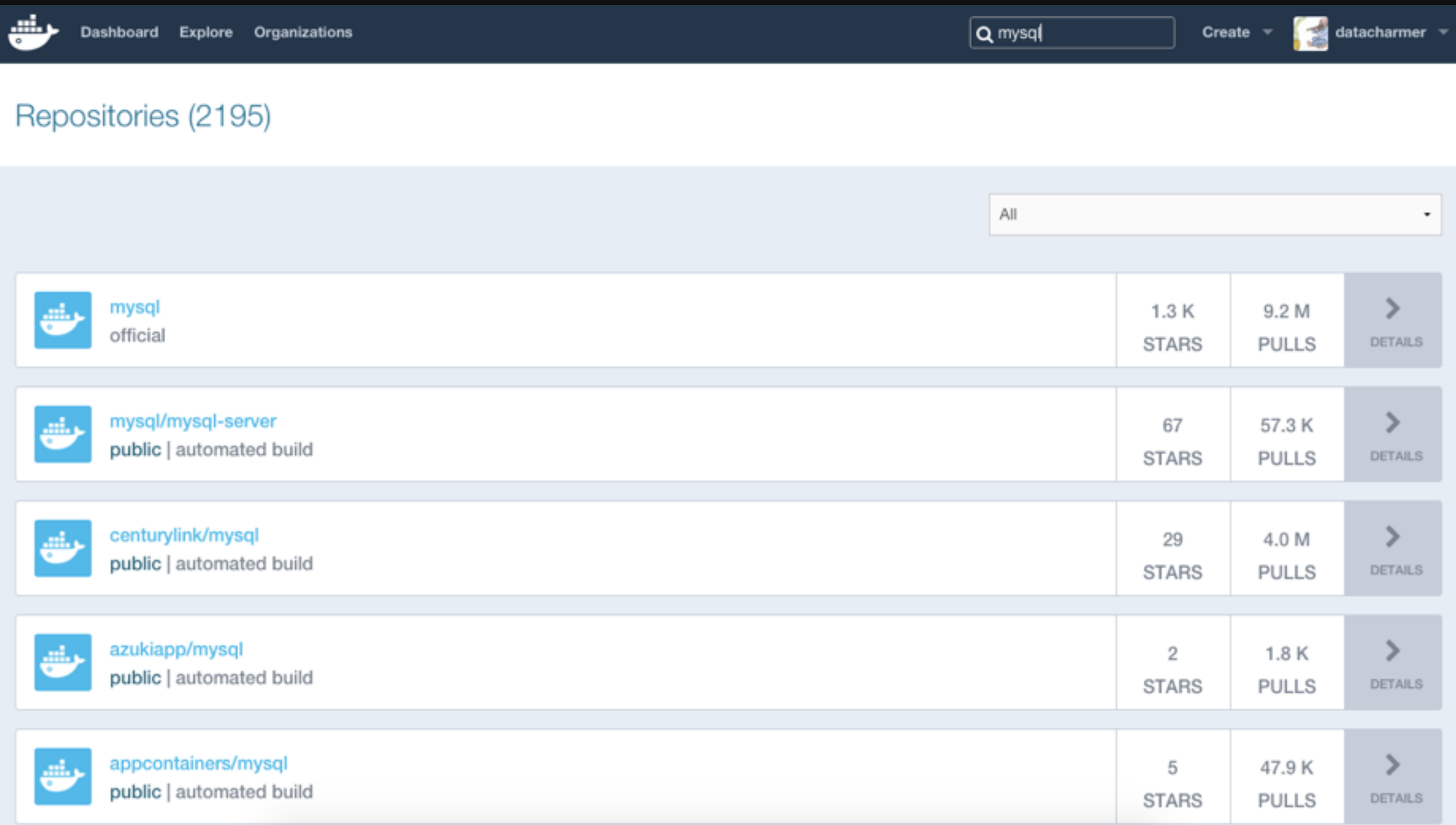
    docs.docker.com/engine/installation/

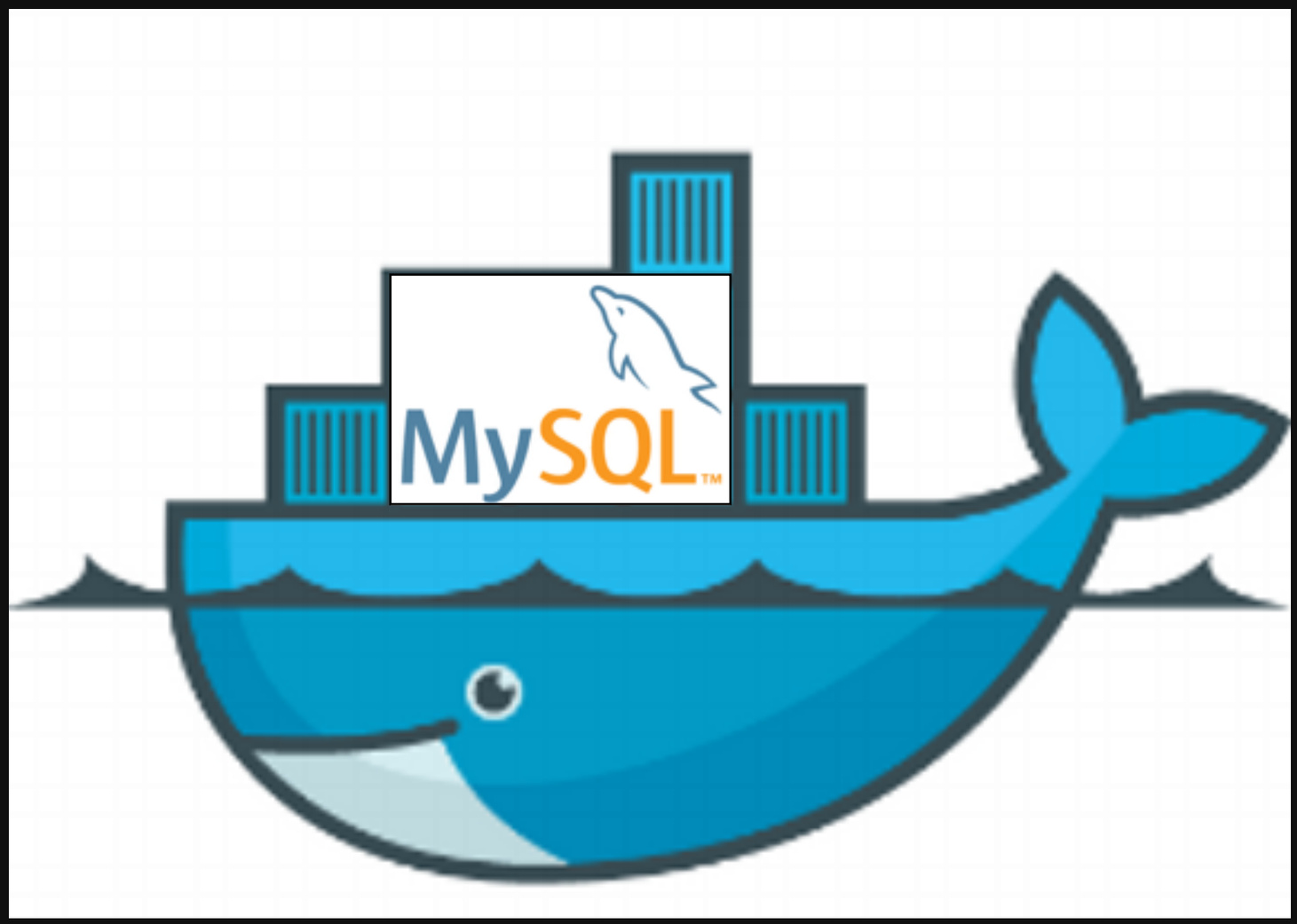
First step

You need to download the image

```
$ docker pull mysql/mysql-server
Using default tag: latest
latest: Pulling from mysql/mysql-server
5e7ce0e805ba: Pull complete
4b9736773855: Pull complete
4fb9c55b1f85: Pull complete
19452ca711c2: Pull complete
0c660f33db44: Pull complete
b8e96405c8c7: Pull complete
9bdbb574fa66: Pull complete
24c2343e048d: Pull complete
Digest:
sha256:da24eddcb99cbeba979d7da8170991278ca59838401a7c7bf17c66307acc641c
Status: Downloaded newer image for mysql/mysql-server:latest
```

hub.docker.com





Seven key points

#1 - containers are full systems

It may be called "a micro-service" but ...

- Every container contains an executable layer
- It's a full O.S.

#2 containers are always Linux

So far, there is no way around it

- Linux flavor in the container is independent from the host
- But they have a compatible kernel
- That's why containers are faster than VMs

#3 - the O.S. in the container is minimal

Don't expect frills

- The container has only the bare minimum to run the service

#4 - installation requires a password

You need to decide your security strategy beforehand

- **You can either pass a password on the command line**
- **or**
- **Tell the container to generate a random password (don't!)**

#5 - Containers are isolated

Don't expect the server to be accessible

- A MySQL server inside a container can only be accessed by other containers
- but
- You can expose ports to the outside

#6 - Data storage can be tricky

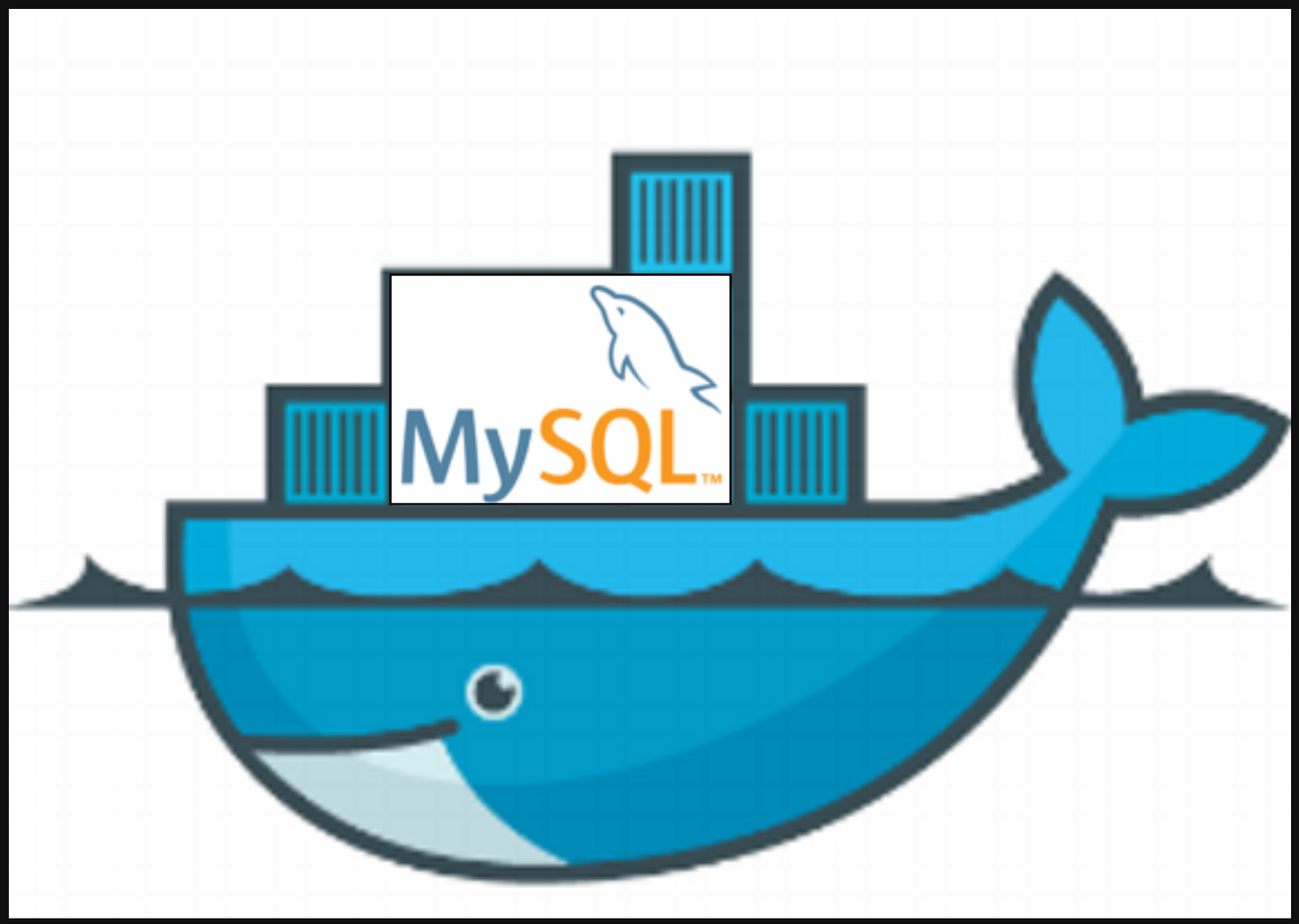
The data directory is inside a container

- By default, the data directory is inside the container
- Containers are volatile by default
- You can use an external "volume" for persistence

#7 configuration is sone by adding files

Forget puppet and Chef. Your container comes ready to use

- **For Virtual machines:**
  - modifying with automated tools is good practice
  - It takes time, but this is the way the game works
- **For Containers**
  - NEVER modify a deployed container
  - Always deploy with ready made components
  - It gives you instantaneous containers!



Examples

Simple MySQL deployment

Here's the minimum command to deploy a MySQL container

```
docker run \
     --name mybox  \
     -e MYSQL_ROOT_PASSWORD=secret \
     -d \
     mysql/mysql-server

06d339254350045 5adcfe6cba36736739c46688de80286 47d
38109c0ec3cf250


# WILL RUN WITH DEFAULT VALUES
```

MySQL with log-bin and server-id

To deploy a customized container we have two options:

```
## OPTION 1 : on the command line
docker run \
    --name mybox  \
    -e MYSQL_ROOT_PASSWORD=secret \
    -d \
    mysql/mysql-server --log-bin --server-id=100
```

MySQL with log-bin and server-id

To deploy a customized container we have two options:

```
## OPTION 2 : with a "volume" file
$ cat minimal.cnf
[mysqld]
user=mysql
log-bin=mysql-bin
server-id=100

$ docker run \
    --name mybox  \
    -e MYSQL_ROOT_PASSWORD=secret \
    -d --hostname mybox \
    -v $PWD/minimal.cnf:/etc/my.cnf
    mysql/mysql-server
```

Inspecting a container

It's an operating system. We can peek inside

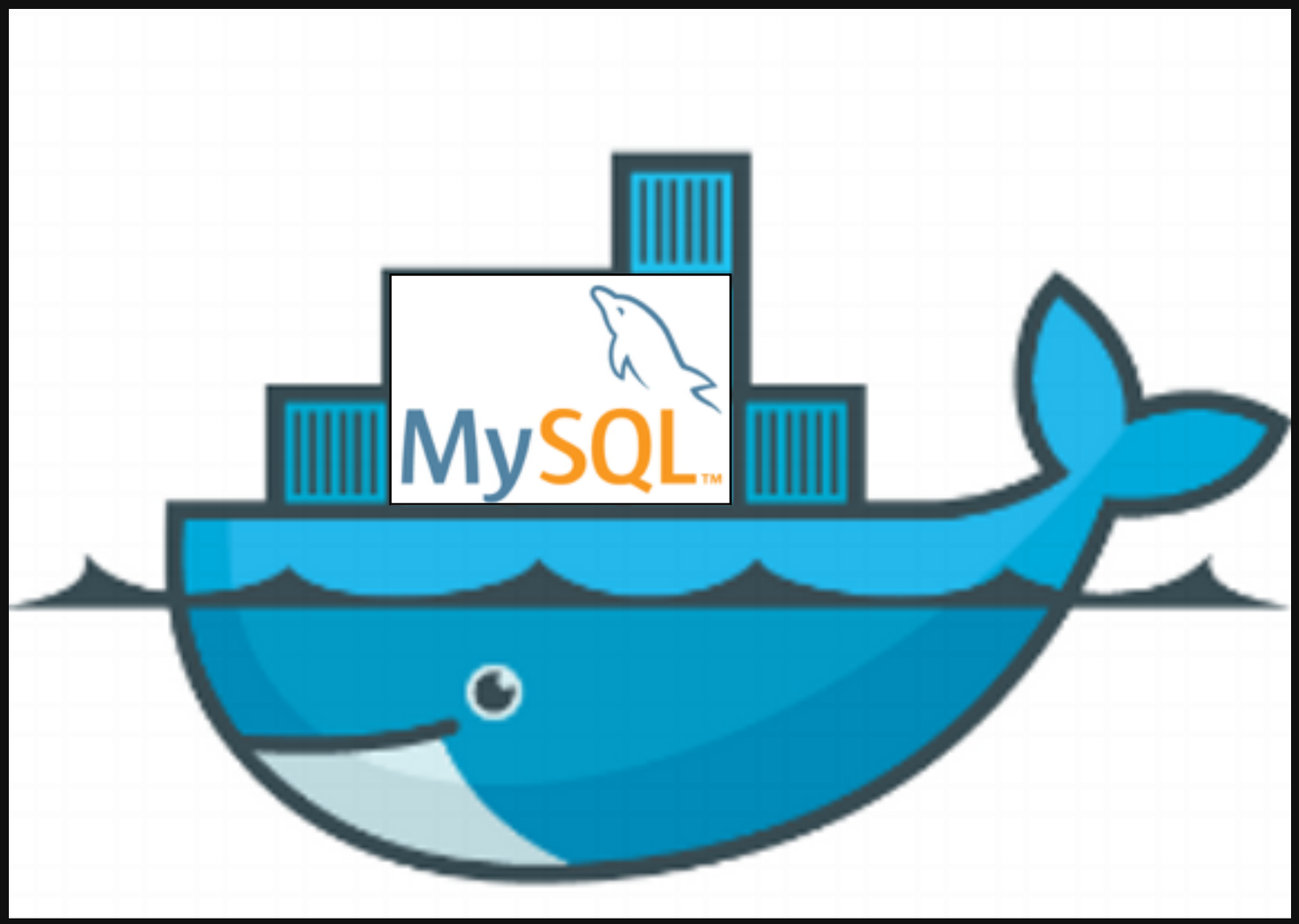
```
docker exec -ti mybox bash
[root@mybox /]# mysql -p
Enter password:
[...]
Server version: 5.7.10-log MySQL Community Server
(GPL)
[...]
mysql> select @@server_id;
+--------------+
| @@server_id |
+--------------+
|          100 |
+--------------+
1 row in set (0.00 sec)
```

Supporting material and software

http://bit.ly/my-rep-samples

Check ./docker/replication

(or look for 'datacharmer' on GitHub)



Q&A