Network
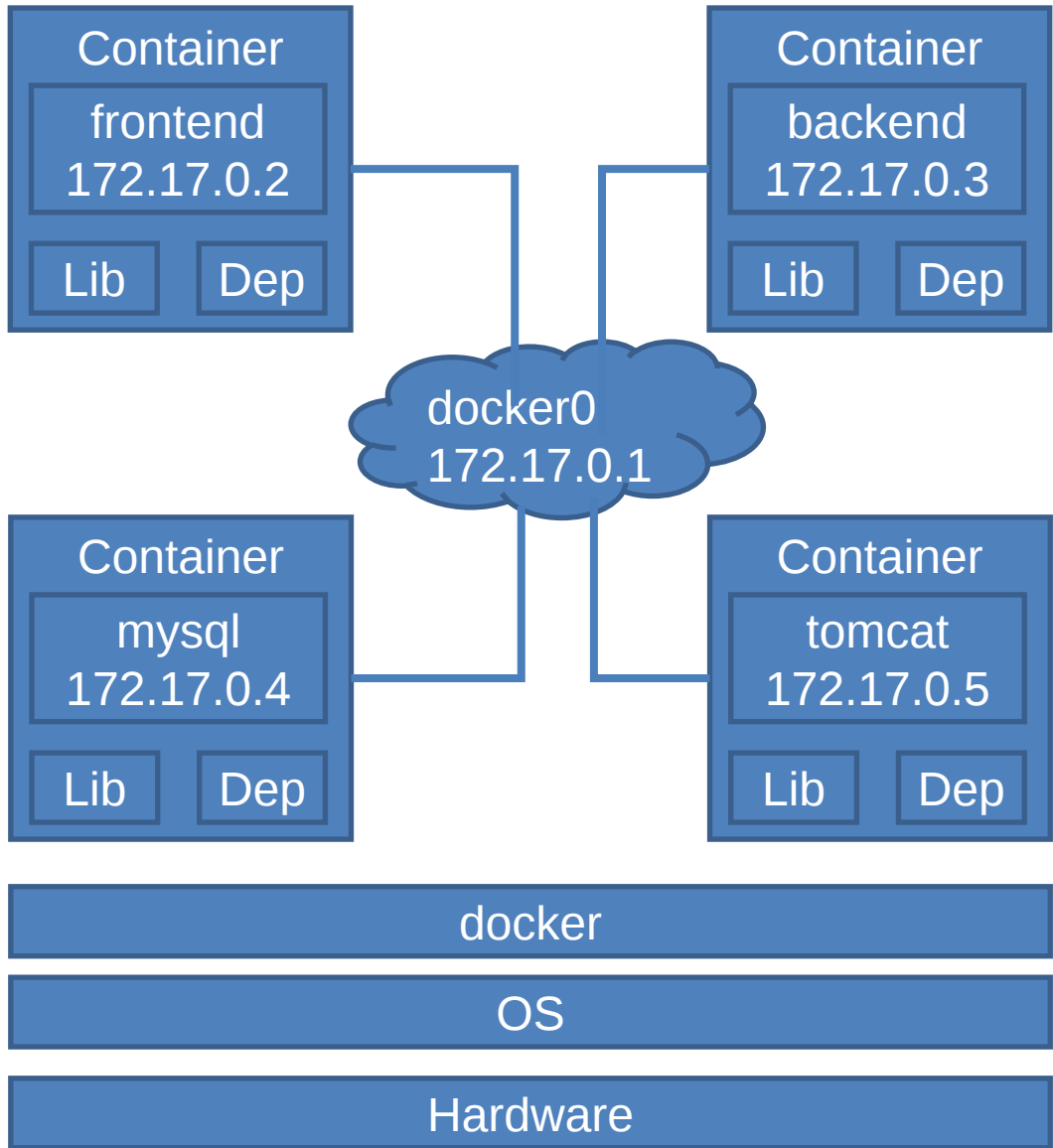
Mag. Thomas Griesmayer

- In terms of Docker, a bridge network uses a software bridge which lets containers connected to the same bridge network communicate, while providing isolation from containers that aren't connected to that bridge network.
- The Docker bridge driver automatically installs rules in the host machine so that containers on different bridge networks can't communicate directly with each other.
- Bridge networks apply to containers running on the same Docker daemon host.
- When you start Docker, a default bridge network (also called bridge) is created automatically, and newly-started containers connect to it unless otherwise specified.
- You can also create user-defined custom bridge networks. User-defined bridge networks are superior to the default bridge network.

https://docs.docker.com/network/drivers/bridge/ (17.2.2024)

Container frontend 172.17.0.2

Lib | Dep

Container backend 172.17.0.3

Lib | Dep

docker0 172.17.0.1

Container mysql 172.17.0.4

Lib | Dep

Container tomcat 172.17.0.5

Lib | Dep

docker

OS

Hardware

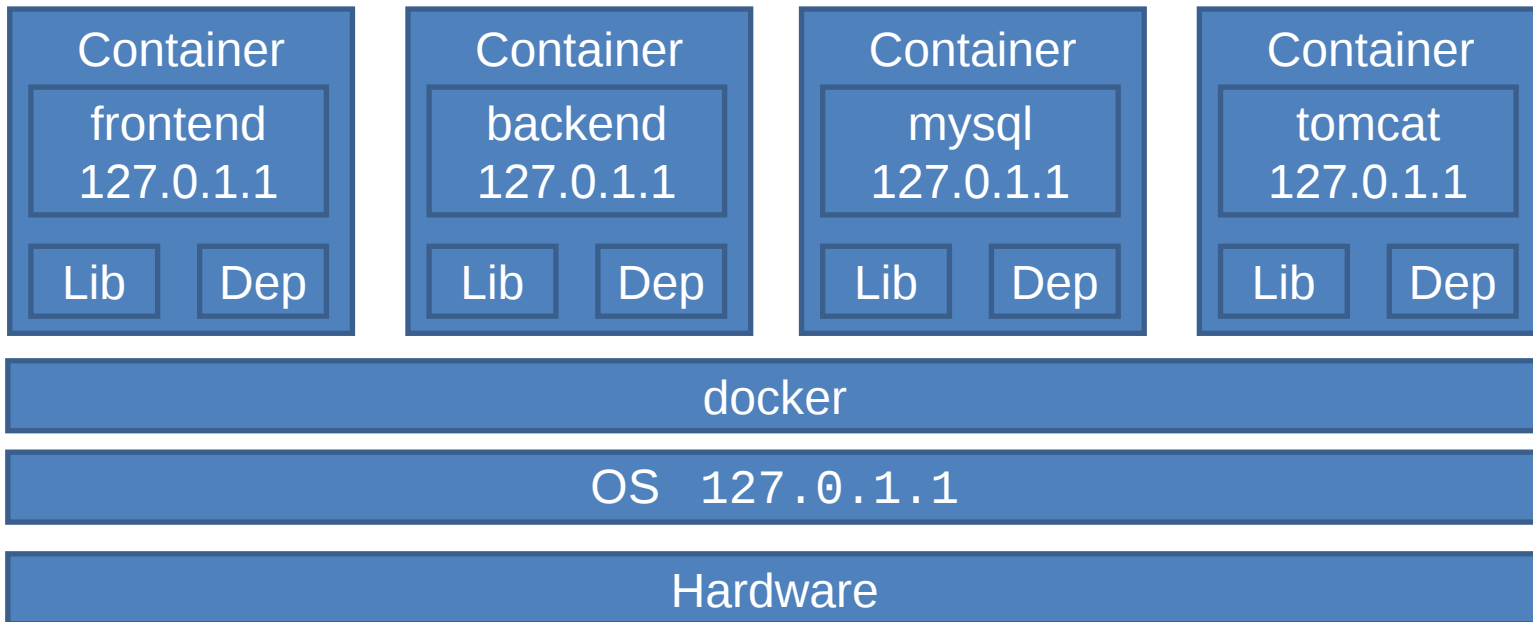## Bridge network driver

```
$ docker run mysql

$ ifconfig
docker0: flags=4099
    mtu 1500
    inet 172.17.0.1
    netmask 255.255.0.0
    broadcast 172.17.255.255
```

Host network driver

- If you use the host network mode for a container, that container's network stack isn't isolated from the Docker host (the container shares the host's networking namespace), and the container doesn't get its own IP-address allocated.

- For instance, if you run a container which binds to port 80 and you use host networking, the container's application is available on port 80 on the host's IP address.

https://docs.docker.com/network/drivers/host/ (17.2.2024)

Host network driver

```
$ docker run --network=host
         mysql
$ hostname -i
127.0.1.1
```

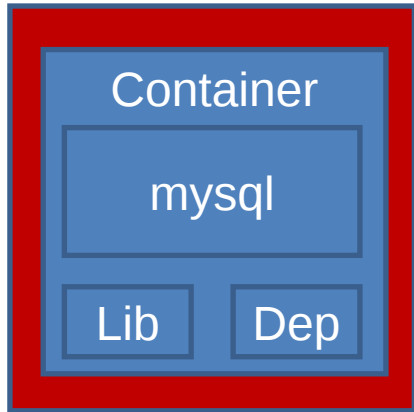| Container	Container	Container	Container
frontend 127.0.1.1	backend 127.0.1.1	mysql 127.0.1.1	tomcat 127.0.1.1
Lib Dep	Lib Dep	Lib Dep	Lib Dep

docker

OS 127.0.1.1

Hardware

- If you want to completely isolate the networking stack of a container, you can use the `--network` none flag when starting the container. Within the container, only the loopback device is created.

None network driver

```
$ docker run --network=none
              mysql
```

Container

mysql

Lib   Dep

docker

OS 127.0.1.1

Hardware

```
$ sudo docker ps
CONTAINER ID   IMAGE    CREATED   PORTS      NAMES
7a945ea323ec   tomcat   4 sec     8080/tcp   compassionate_jackson
$ sudo docker inspect 7a94
...
    "Networks": {
        "bridge": {
            "Gateway": "172.17.0.1",
            "IPAddress": "172.17.0.2",
...
```