

Hello Docker!

<https://github.com/nigel-dunn/docker-workshop>

This is a partial workshop

Focus on development needs

I'm not a sysadmin

It covers a wide range of topics - Docker technology, dev toolchains, servers, etc.

Docker is still developing at a fast pace

... and also a partial workshop

My way of developing with Docker

I develop on and for Unix systems

Use the workshop as a springboard for your own approach

The technology

Docker

Platform first released 2013

Underlying technology - containers

Only runs on Linux

One of a number of container technologies

Google & containers

“From Gmail to YouTube to Search,
everything at Google runs in containers.”

<https://cloud.google.com/containers/>

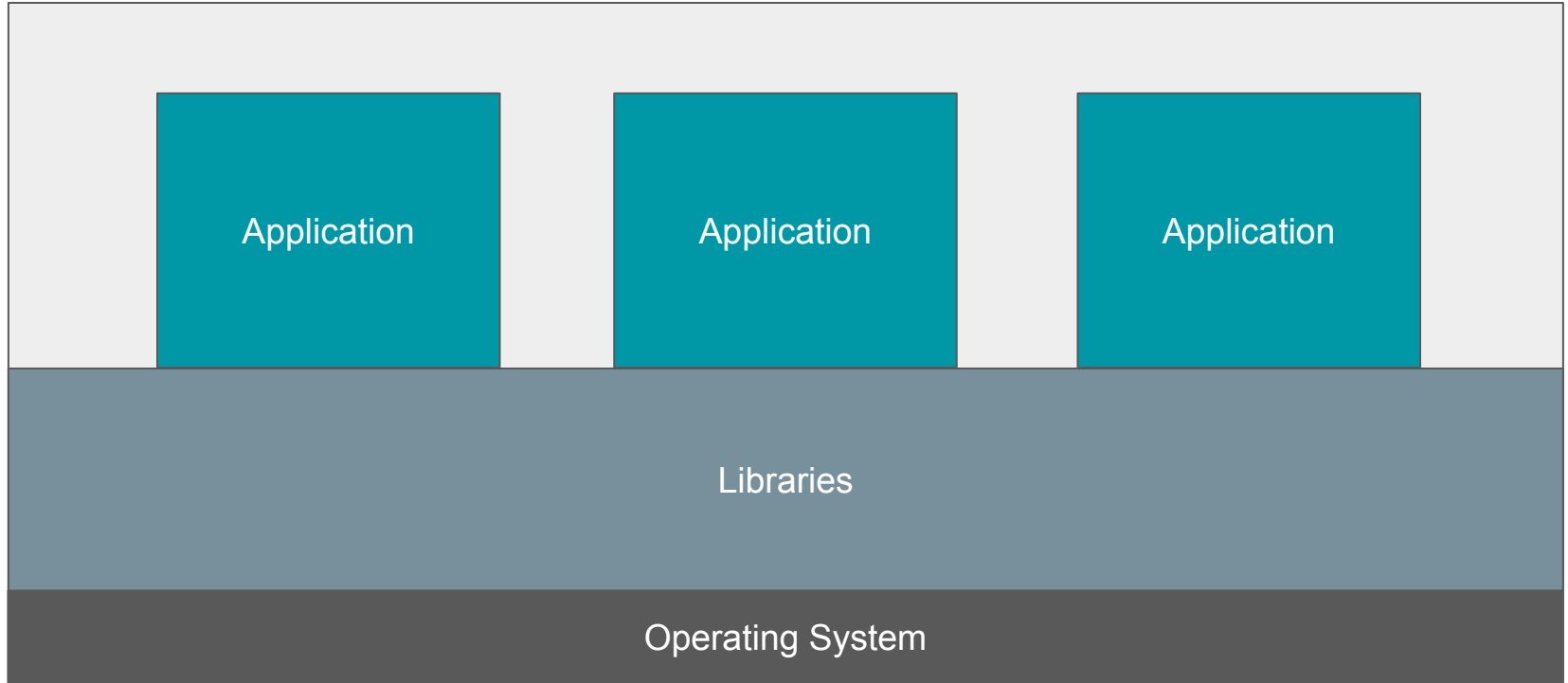
Google & containers

“We now start around seven thousand new containers per second, or over two billion per week.”

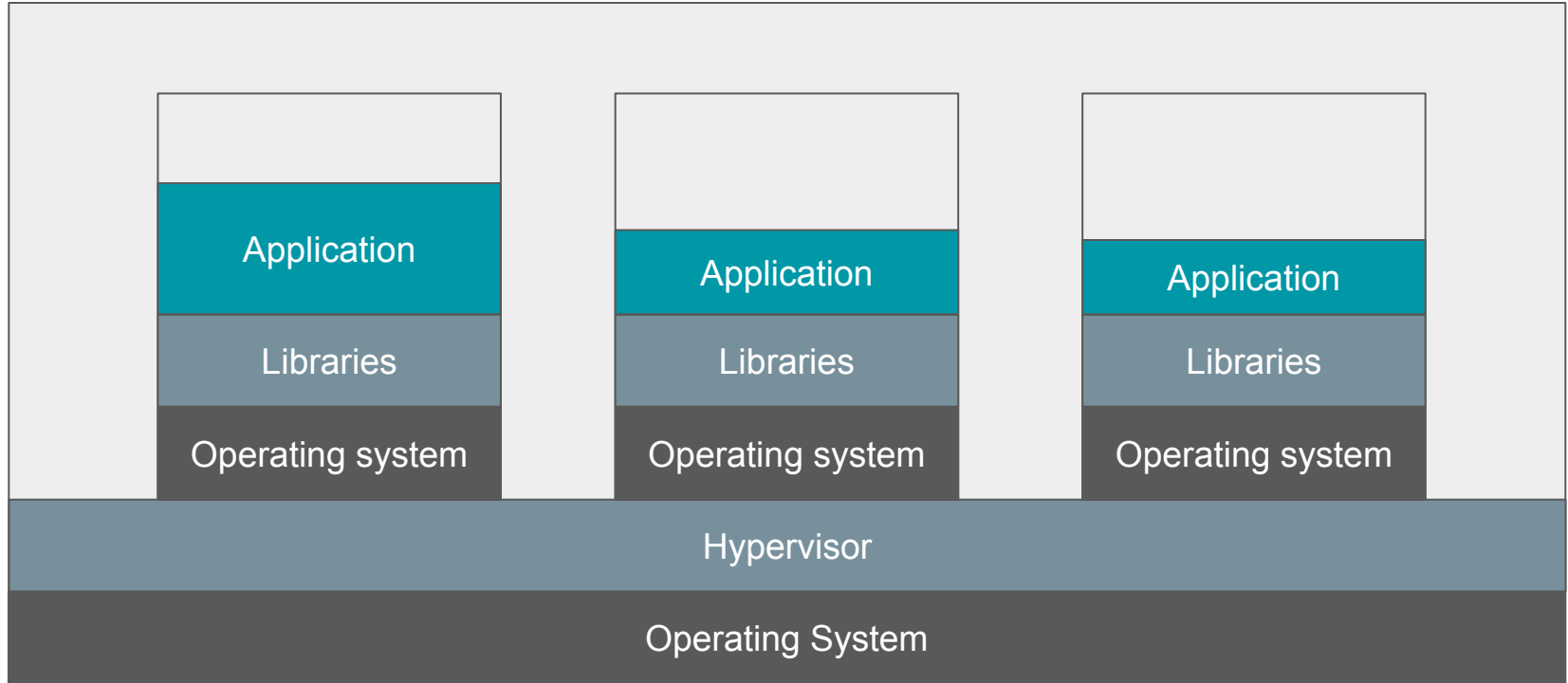
January 2015

<https://cloudplatform.googleblog.com/2015/01/in-coming-weeks-we-will-be-publishing.html>

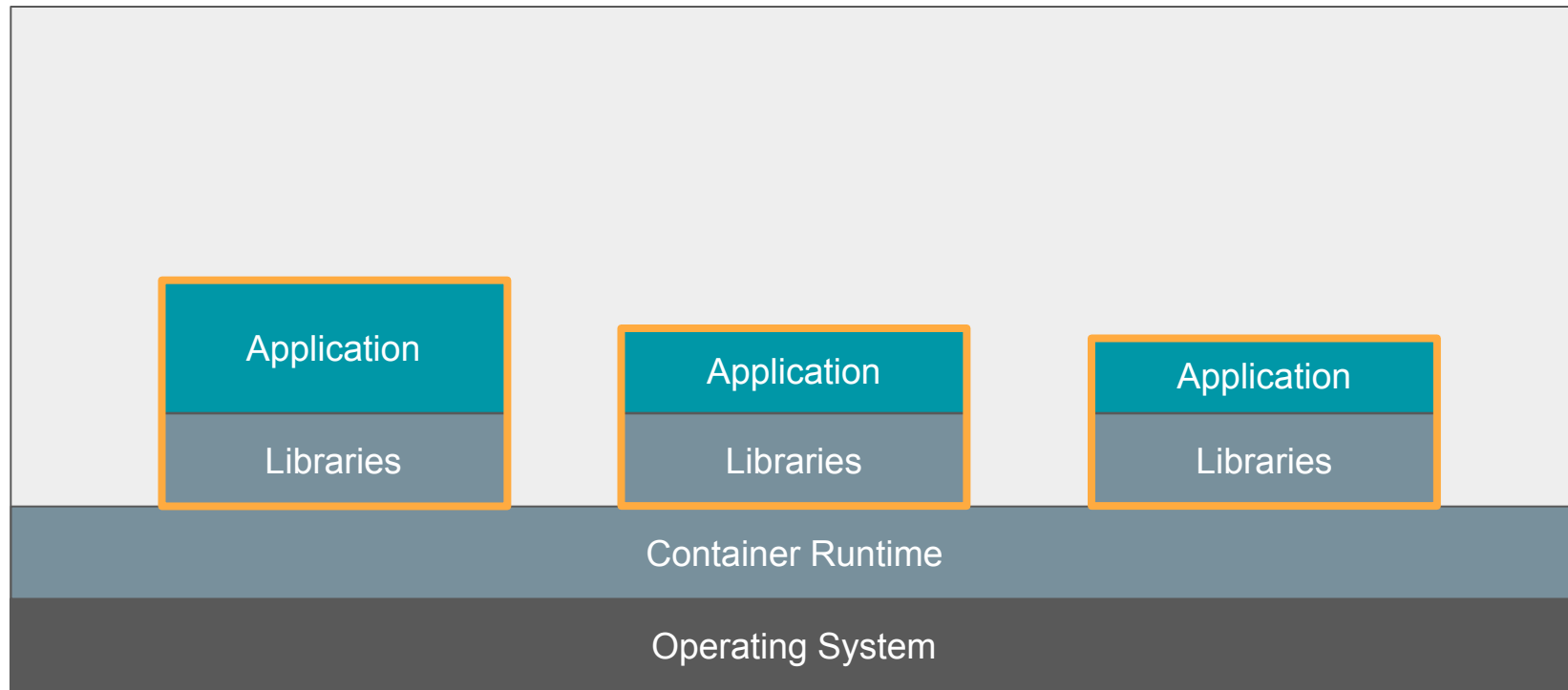
Bare metal



Virtualization



Containers



Hello PHP!

Recap docker-compose.yml

`version` - of docker-compose itself, not your YAML file

`services` - containers

`image` - use pre-built image from repository

`tty` - interactive mode

Recap commands

```
docker-compose up/down
```

```
docker-compose up --detach
```

```
docker-compose exec <service name> bash
```

```
docker logs <container name>
```

```
docker-compose ps
```

```
docker images --all
```

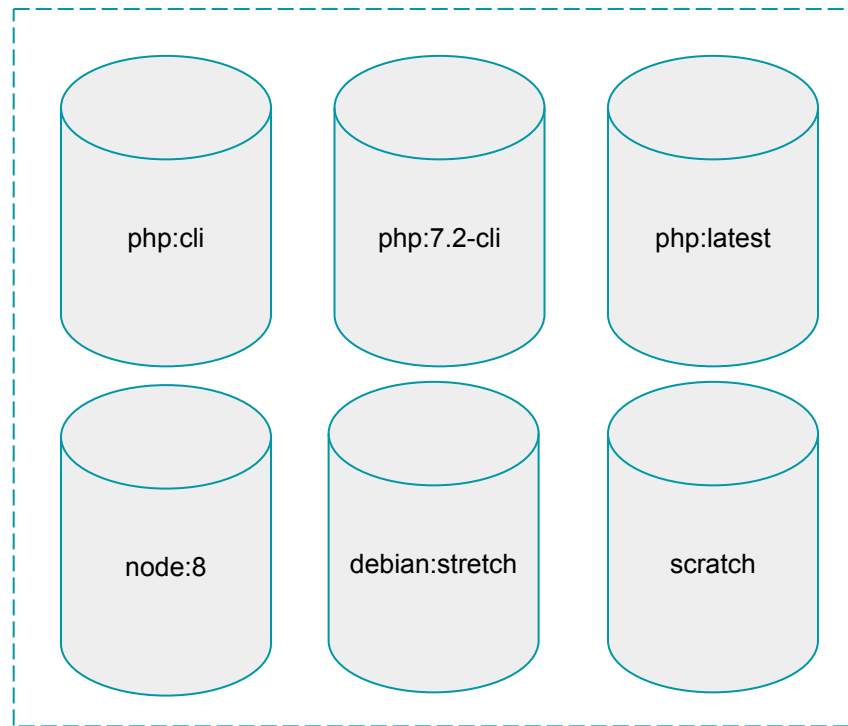
Images

Default repository – Docker Hub

Tags distinguish between versions

Link through to Github to see Dockerfile

All derived from scratch



<https://hub.docker.com>

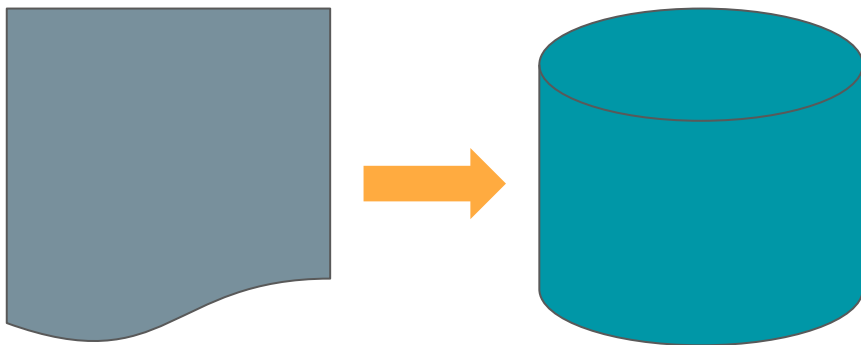
Lifecycle

Lifecycle – source



Dockerfile

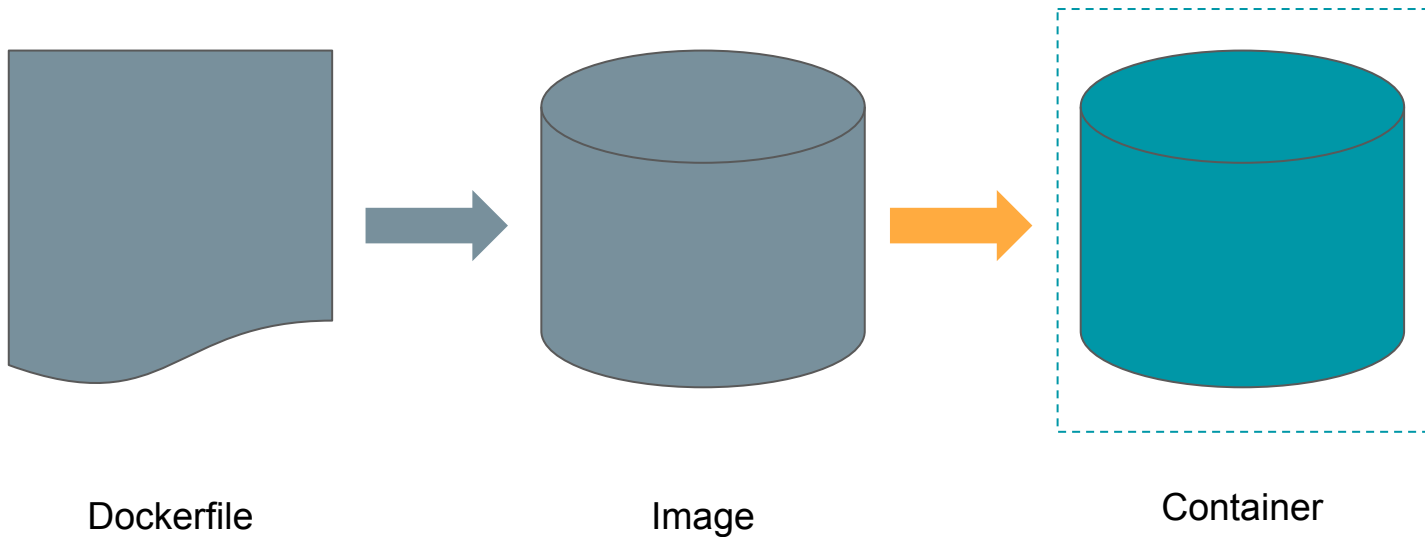
Lifecycle – build



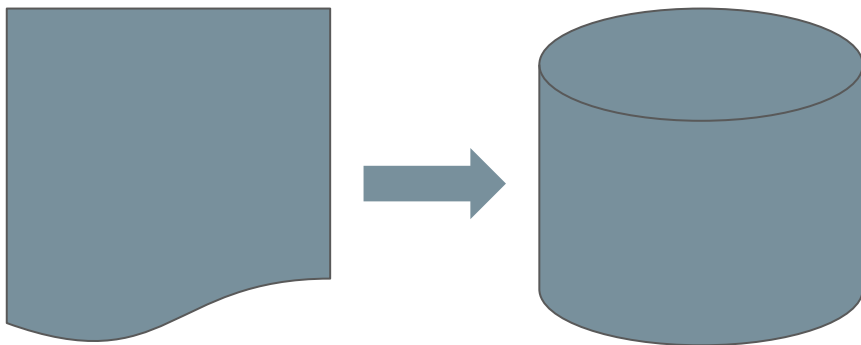
Dockerfile

Image

Lifecycle – create container



Lifecycle – destroy container



Dockerfile

Image

Hello Composer!

Recap Dockerfile

FROM - base image

RUN - shell commands

ARG - variable local to Dockerfile

ENV - environment variable to pass into container

WORKDIR - working directory for commands that follow & interactive sessions

ENTRYPOINT - command to run

Recap docker-compose

```
docker-compose build
```

```
docker-compose build --no-cache
```

```
docker-compose build --pull
```

```
docker-compose up --build
```

```
docker-compose start/stop
```

```
docker-compose build --force-rm
```

```
docker rmi --force $(docker images -q)
```

Building your Dockerfile

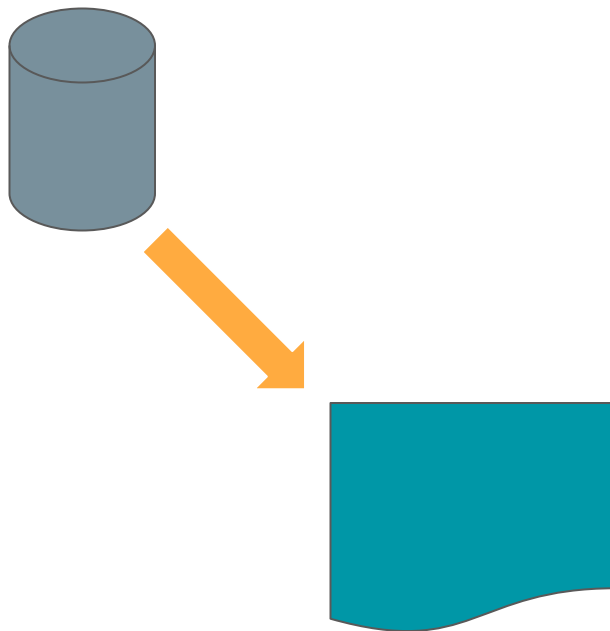
Treat as open-source software

Experimentation

Initial guidance

When was it last built?

Rebuild your images



Storage

Storage – transient

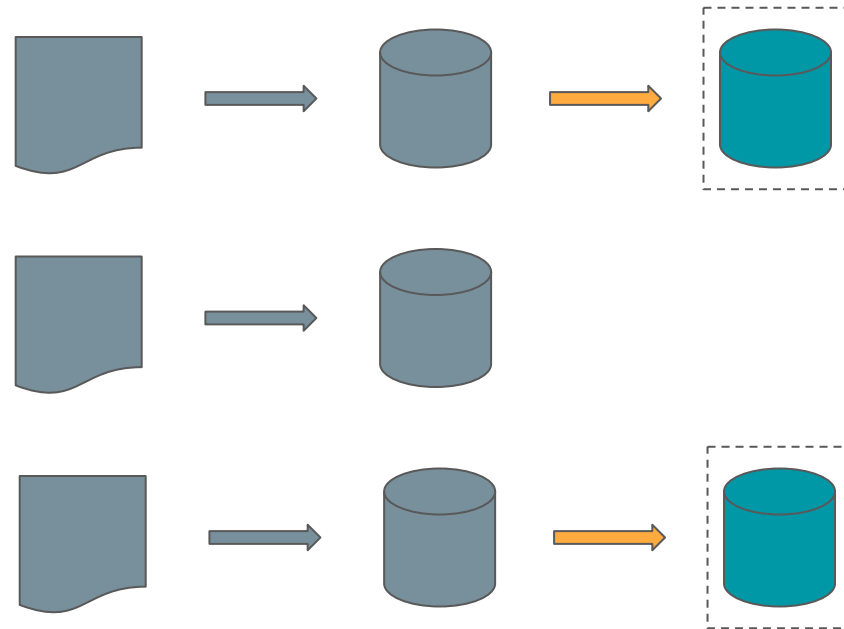
Applications

Custom configuration

Temporary files

Production code

ADD/COPY

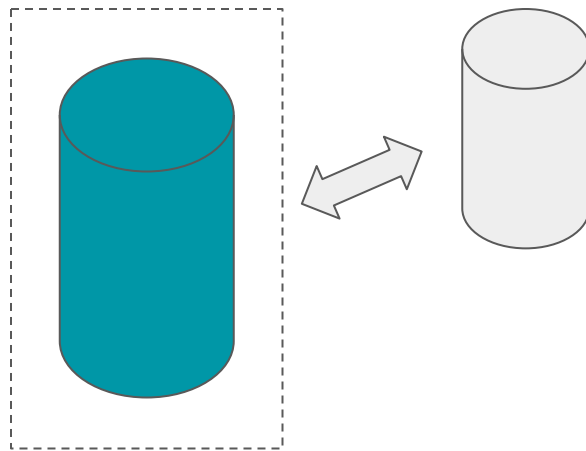


Welcome back Composer!

Storage – named volume

Other client used to access data (e.g. MySQL)

volumes: <name>:<container path>



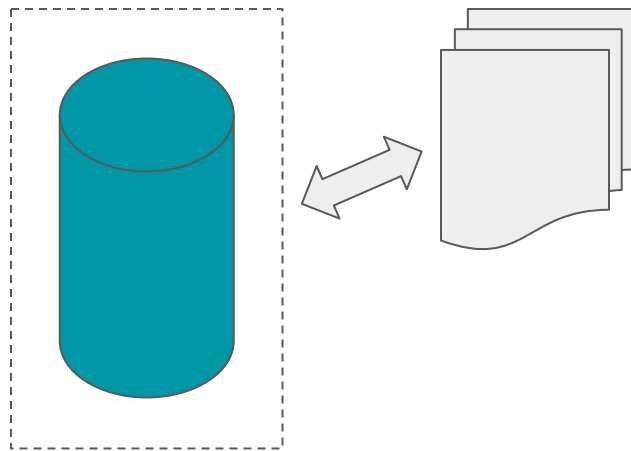
Storage – local path

Code during development

Share secrets from local environment

Override config with dev settings

```
volumes: <local path>:<container path>
```



Hello PHPUnit!

Recap commands

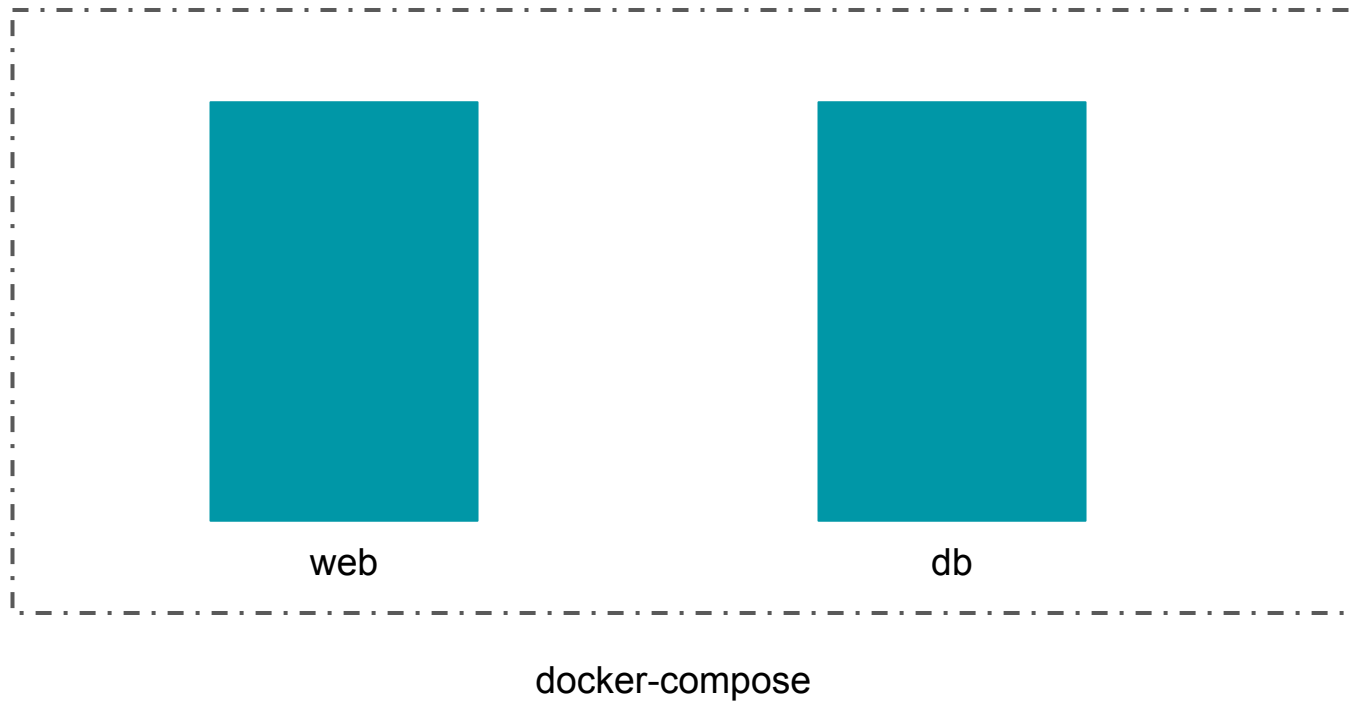
```
docker volume list
```

```
docker volume inspect <name>
```

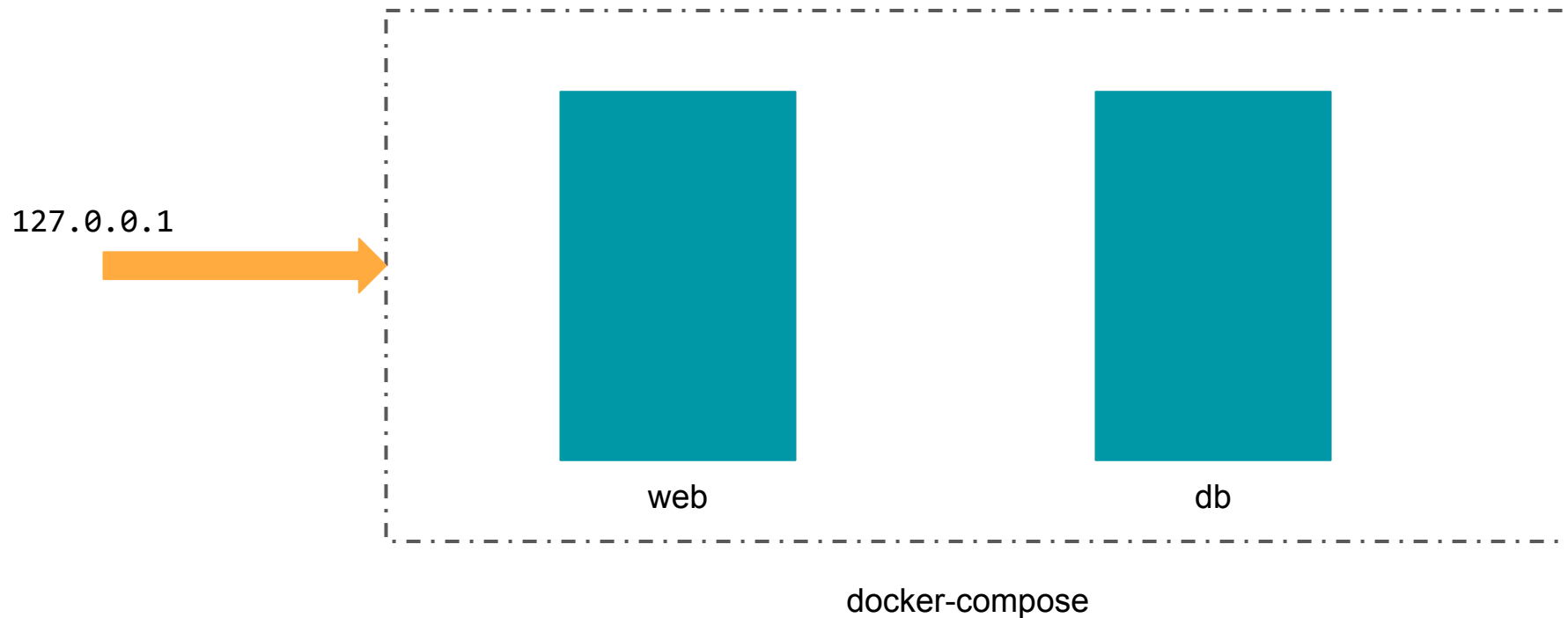
```
docker volume rm <name>
```

Networking

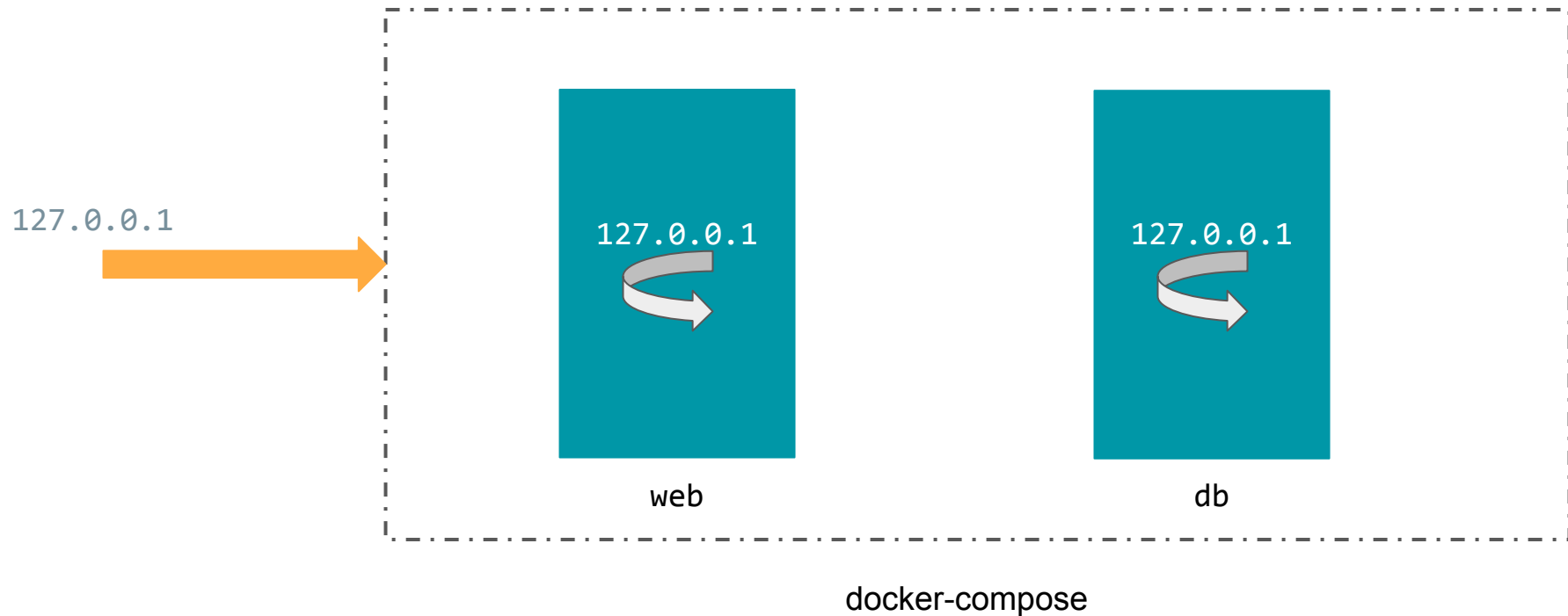
Networking



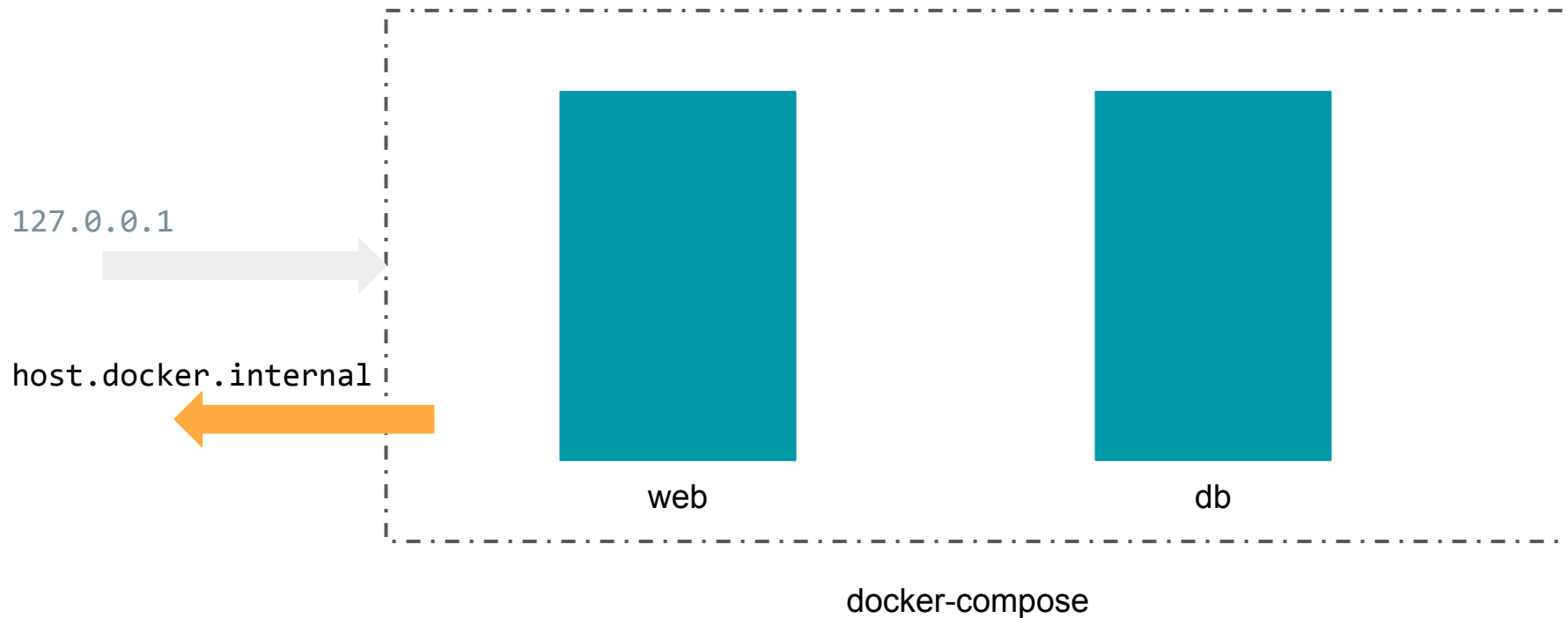
Networking – addresses



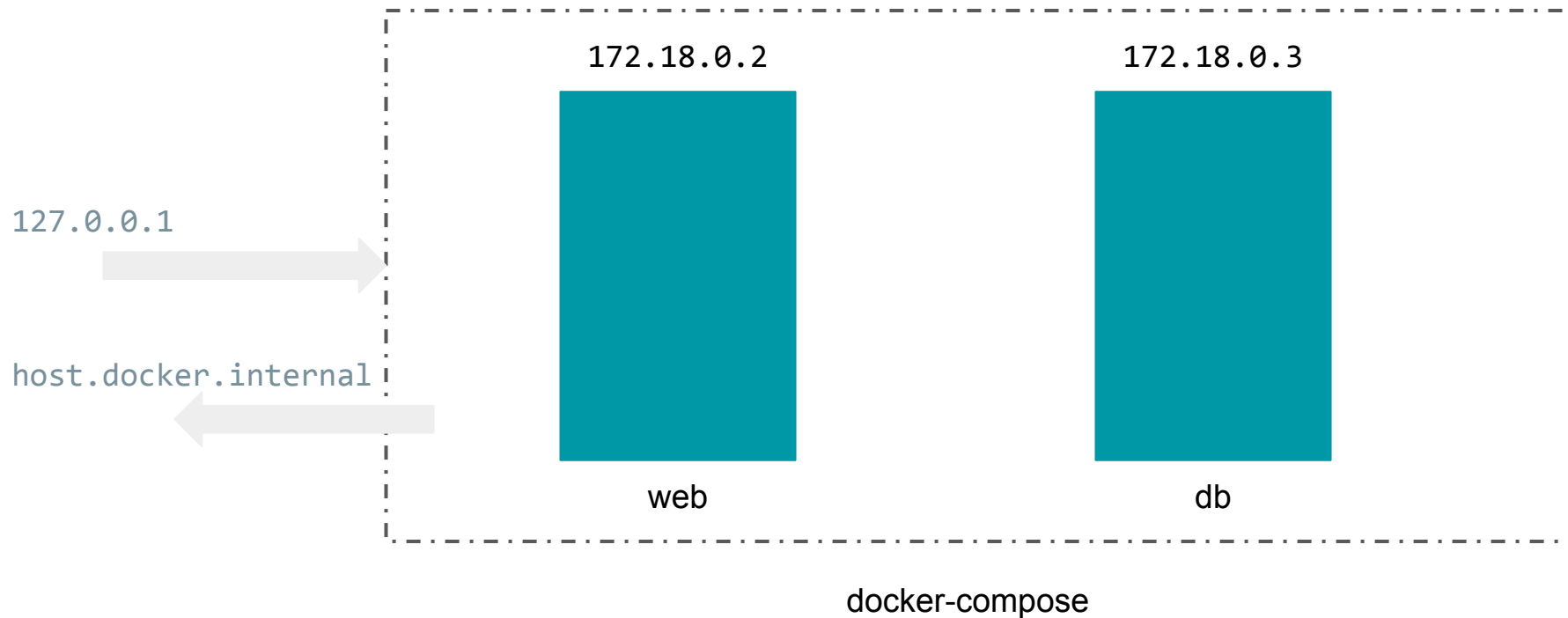
Networking – addresses



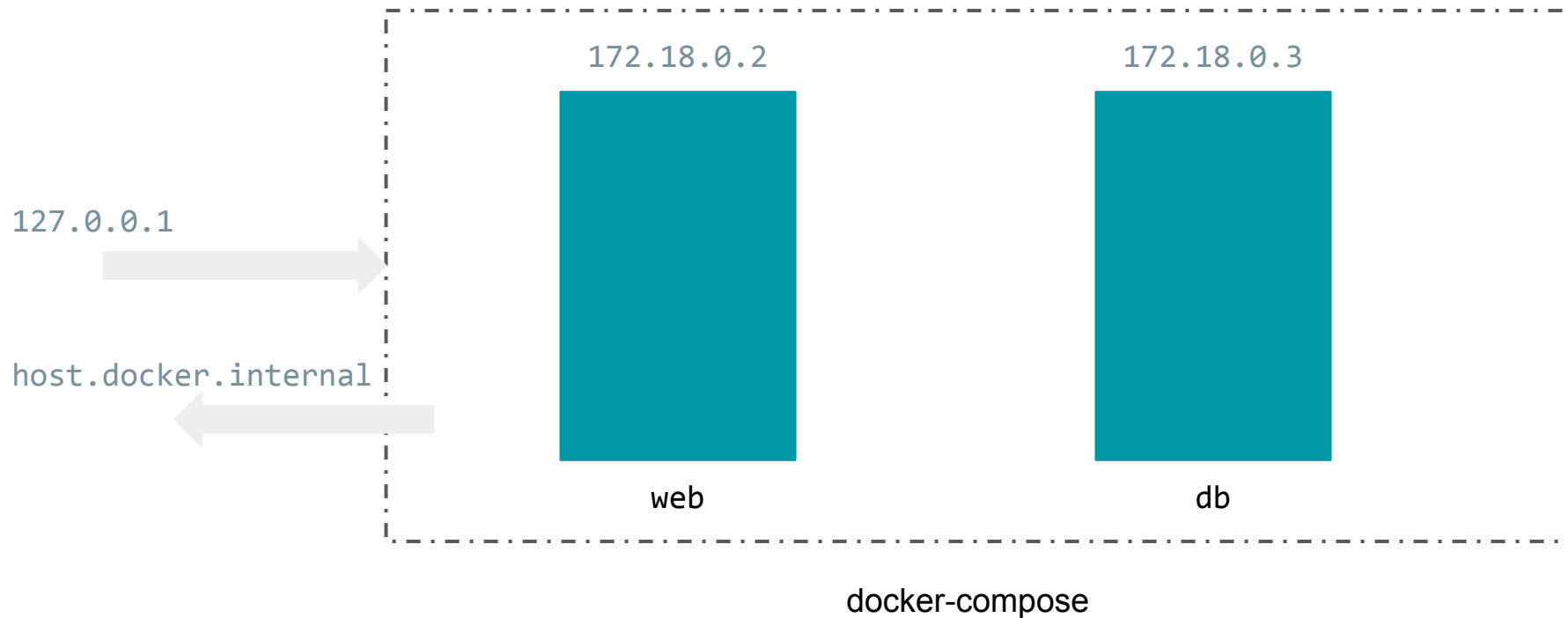
Networking – addresses



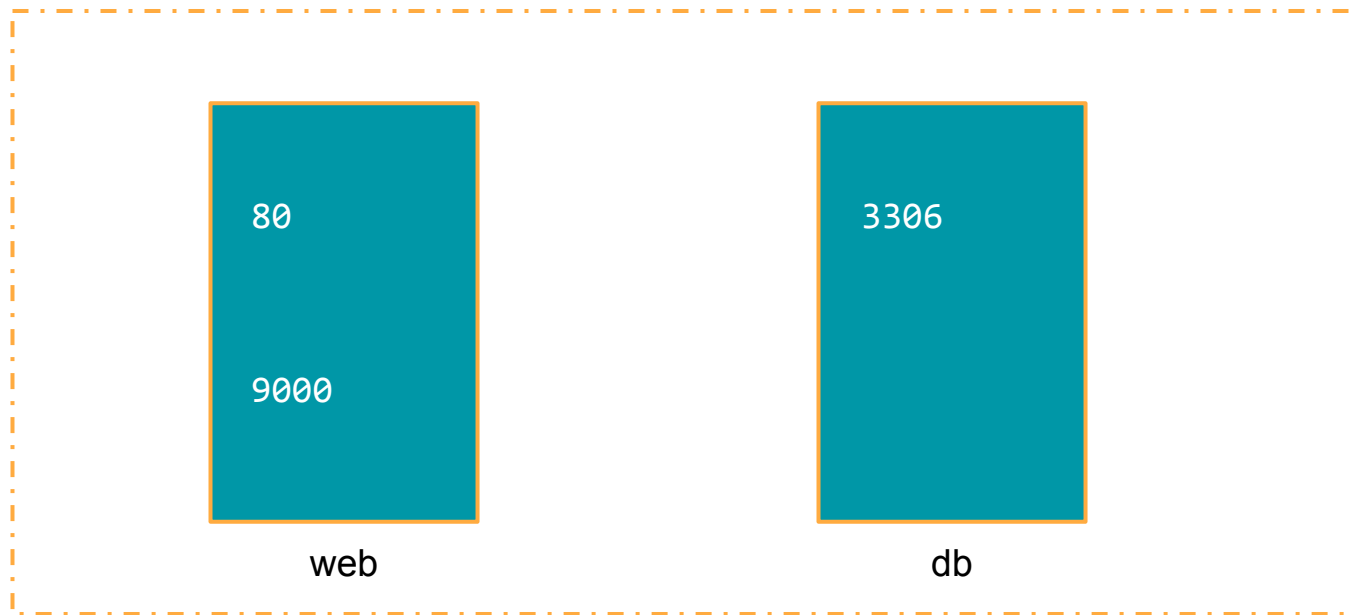
Networking – addresses



Networking – addresses



Networking – ports

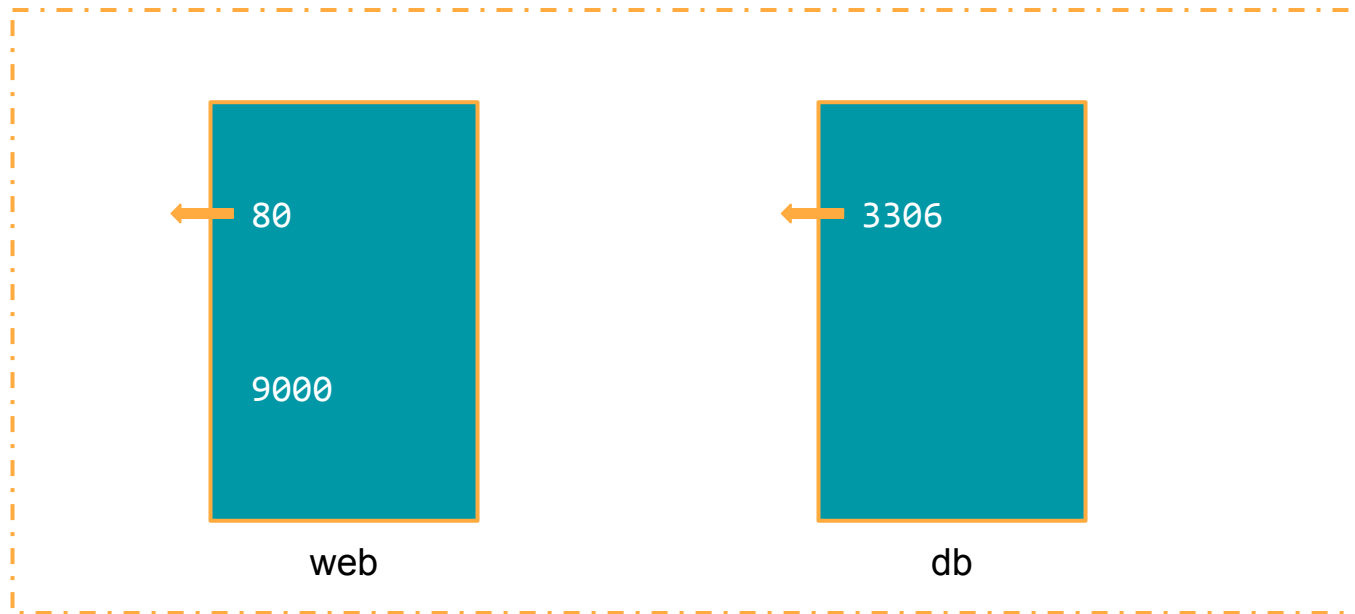


docker-compose

Networking – ports

EXPOSE 80

EXPOSE 3306



docker-compose

Networking – ports

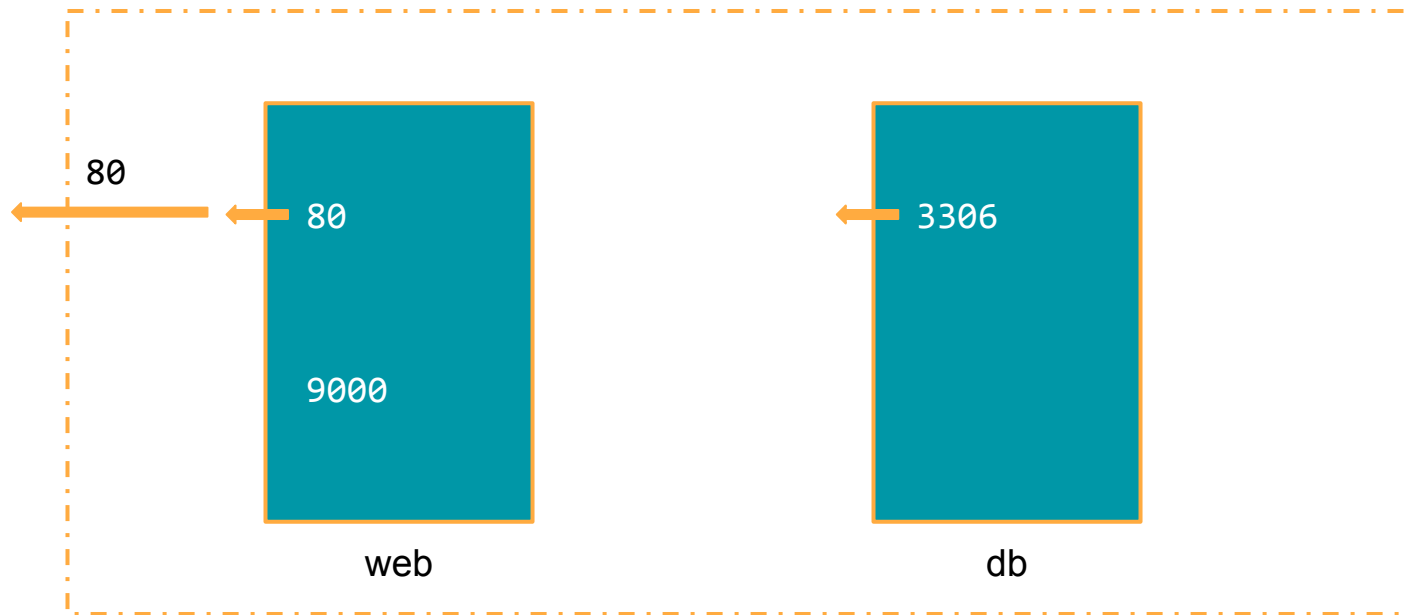
EXPOSE 80

web:

ports:

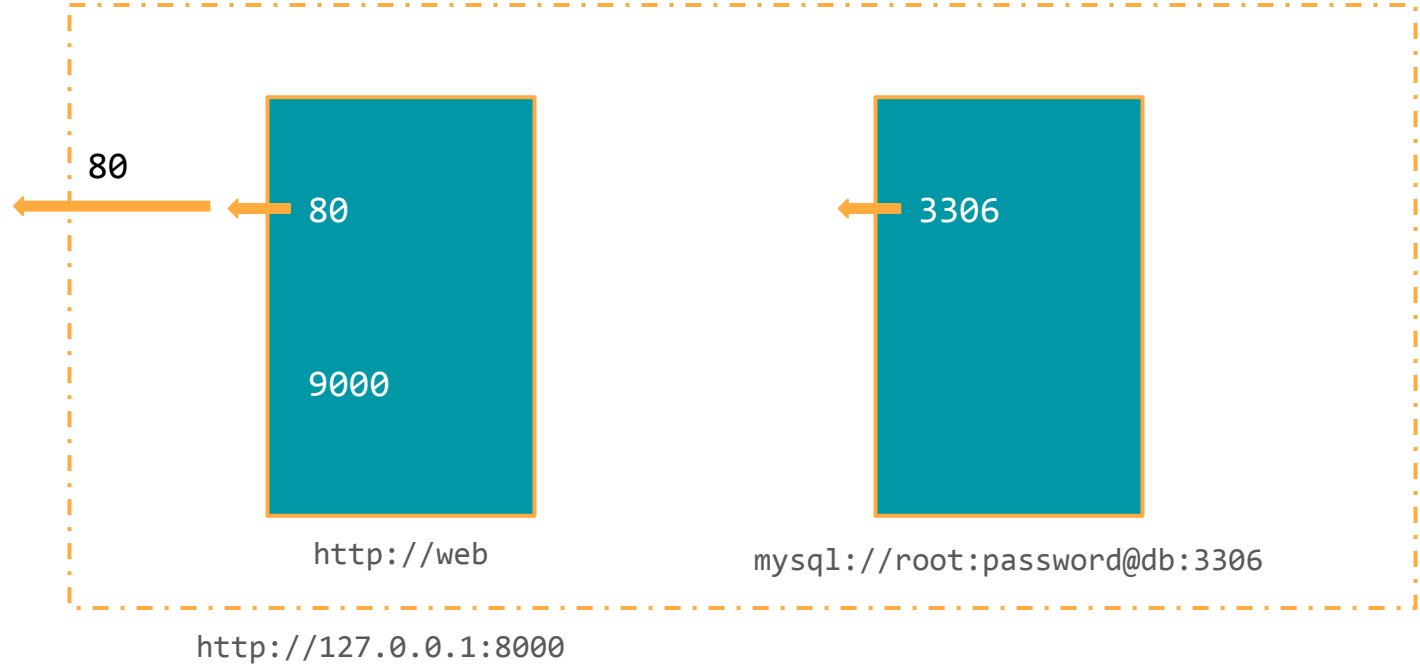
- 80:80

EXPOSE 3306



docker-compose

Networking

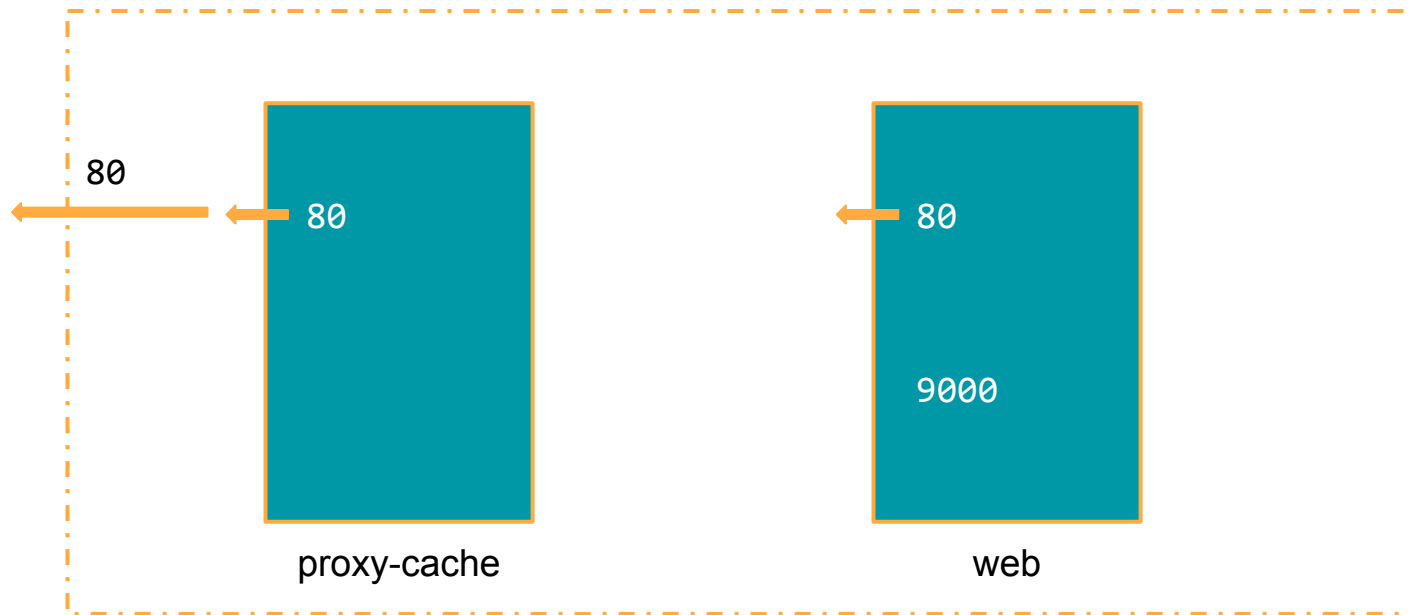


Networking – ports

EXPOSE 80

proxy-cache:
ports:
- 80:80

EXPOSE 80



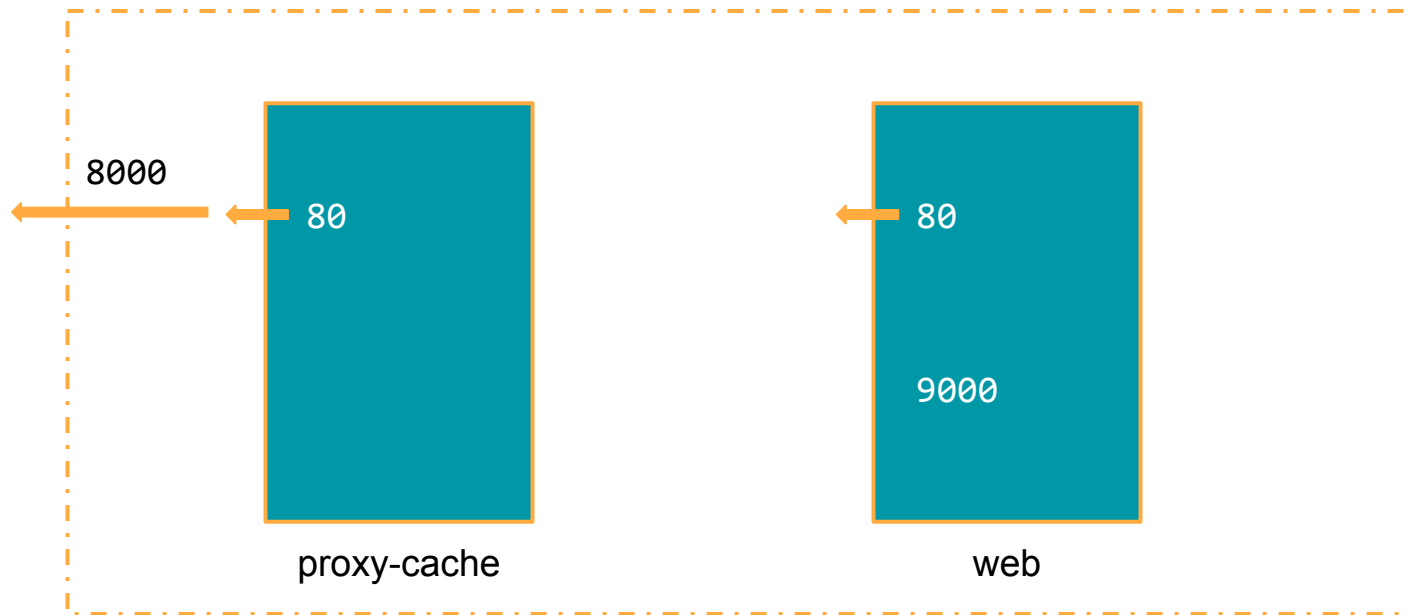
docker-compose

Networking – ports

EXPOSE 80

proxy-cache:
ports:
- 8000:80

EXPOSE 80



docker-compose

Hello apache!

Recap configuration

Dockerfile

EXPOSE <port>

docker-compose.yml

ports: <local port>:<container port>

Your Docker services run on localhost (127.0.0.1)

Recap commands

```
docker network list
```

```
docker network inspect <name>
```

```
docker network rm <name>
```

Digression

Configuring XDebug on Docker for Mac (& probably Windows) is non-intuitive

Google “xdebug docker 10.254.254.254”

Hello WordPress!

Recap docker-compose.yml

`build:` `context` - path to Dockerfile

`build:` `dockerfile` - alternative name for Dockerfile

`environment` - override value in Dockerfile

Goodbye!