

Hello Docker!

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

Aims of the workshop

- Understanding
 - Technology
 - Configuration
 - Commands
- Development environments
 - Command line
 - Service
- Configuration
 - From scratch
 - Amending existing config

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

Docker

Platform first released 2013

Underlying technology - containers

Only runs on Linux

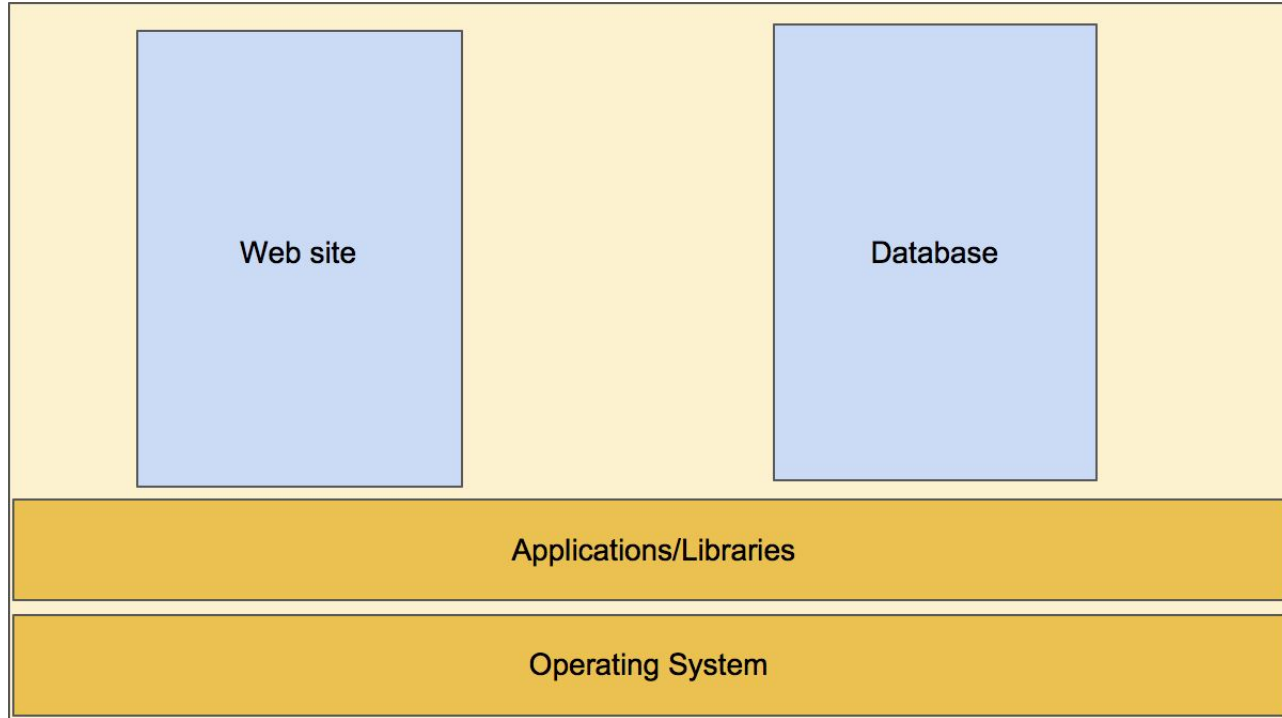
One of a number of container technologies

In 2014 -

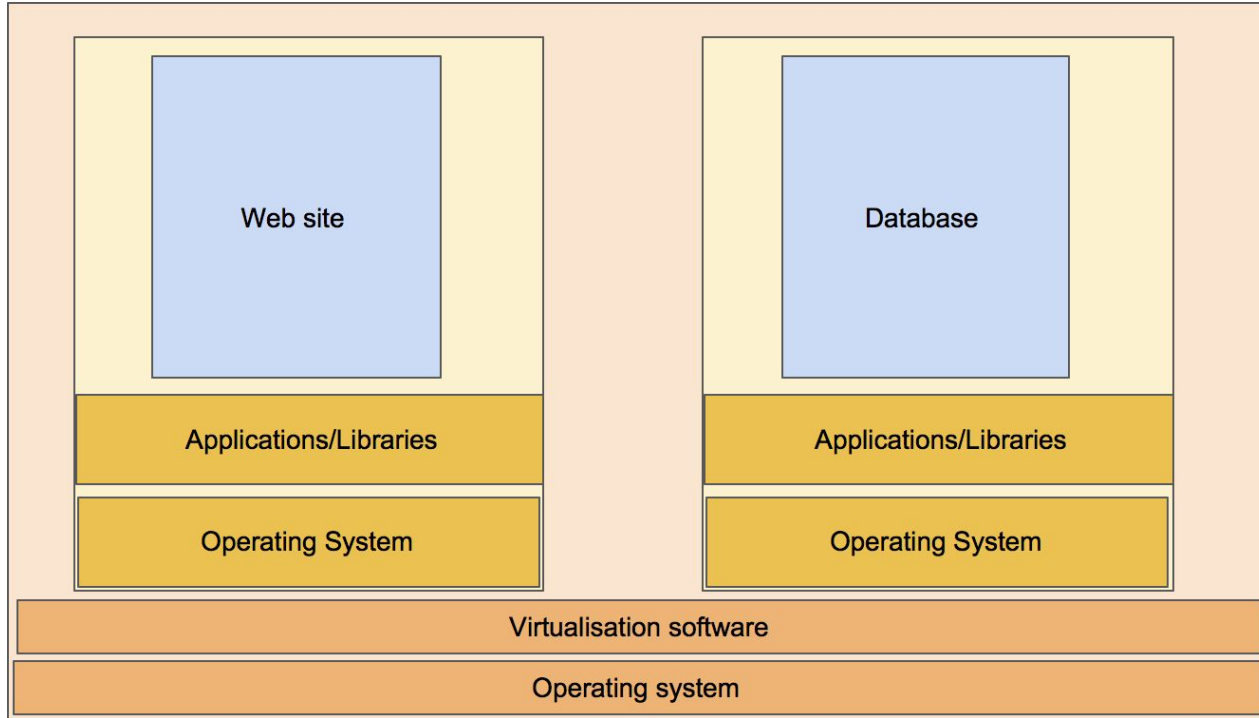
“Everything at Google runs in a container”

“[Google] start over two billion containers per week”

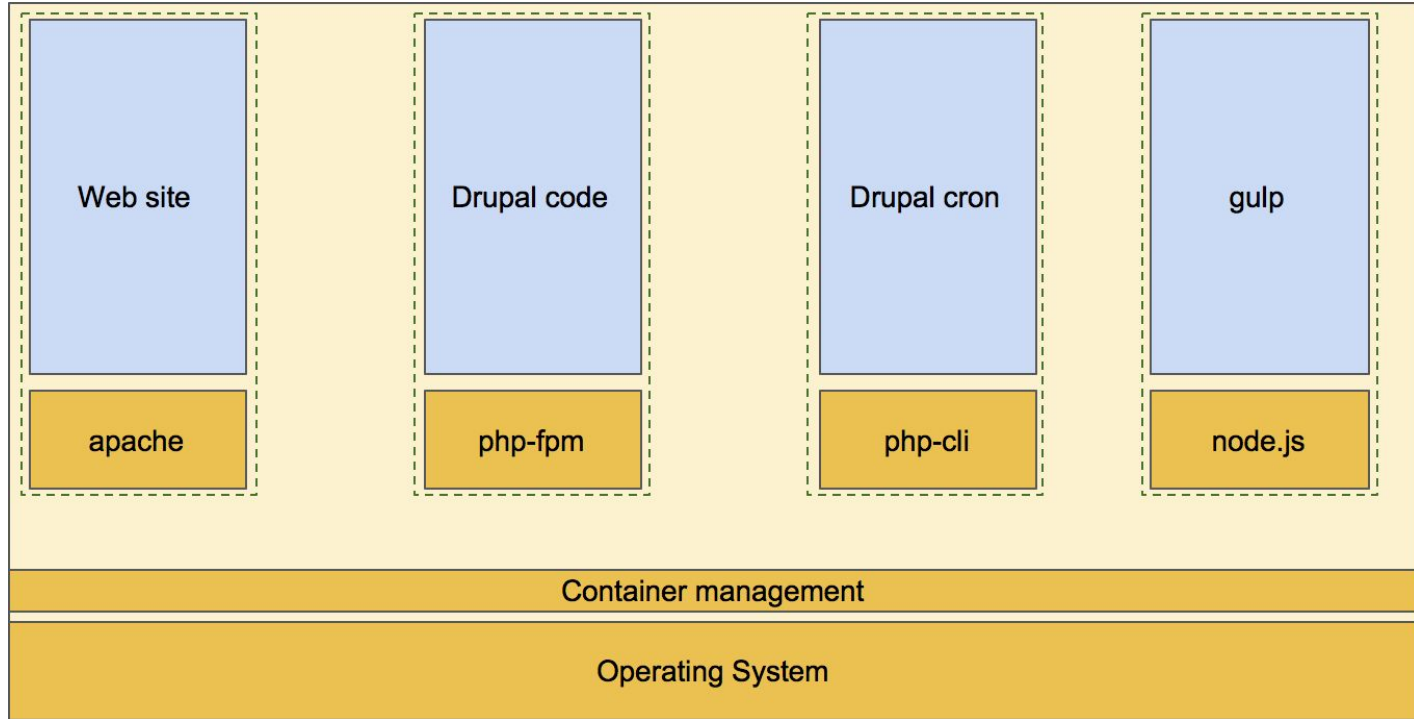
Bare metal



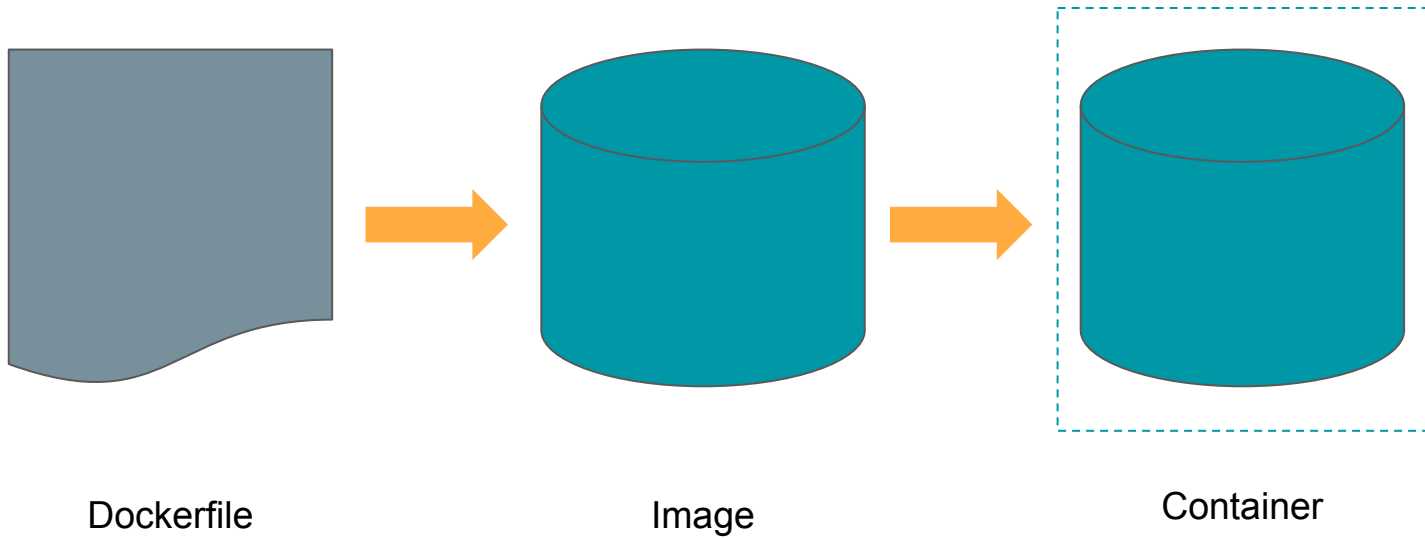
Virtualization



Containers



Lifecycle



Docker terminology

Dockerfile - source code

Image - compiled version of Dockerfile

Container - environment running an image

Docker Hub - official image repository

Docker terminology

Dockerfile - source code

Image - compiled version of Dockerfile

Container - environment running an image

Docker Hub - official image repository

Docker Compose - 'dev' management of a group of containers

Kubernetes - Google-derived orchestration tool

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-compose start/stop`

`docker images --all`

Images

Default repository - Docker Hub

Tags - allows specific versions

Image listings will link through to source repository

Images

Default repository - Docker Hub

Tags - allows specific versions

Image listings will link through to source repository

Only use reputable repositories

Treat images as you would other open-source software

Beware of old images

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 --force-rm
```

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

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

Storage scenarios

	IMMUTABLE	MUTABLE
THIRD-PARTY FILES	Applications	Database
FIRST-PARTY FILES	Custom configuration	Code

Storage scenarios

	IMMUTABLE	MUTABLE
THIRD-PARTY FILES	Applications	Database
FIRST-PARTY FILES	Custom configuration	Code

Immutable - part of the image

Third-party as they came with the installation of applications

Welcome back Composer!

Storage scenarios

	IMMUTABLE	MUTABLE
THIRD-PARTY FILES	Applications	Database
FIRST-PARTY FILES	Custom configuration	Code

`docker-compose.yml`

`volumes: <named volume>:<container path>`

Storage scenarios

	IMMUTABLE	MUTABLE
THIRD-PARTY FILES	Applications	Database
FIRST-PARTY FILES	Custom configuration	Code

`docker-compose.yml`

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

Hello PHPUnit!

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)

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!