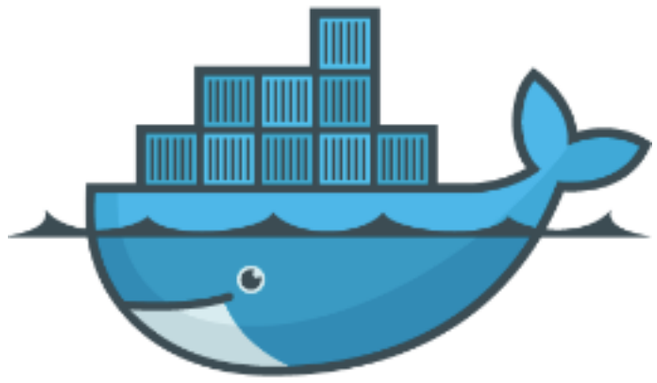


A multi-container Symfony2 setup with Docker.

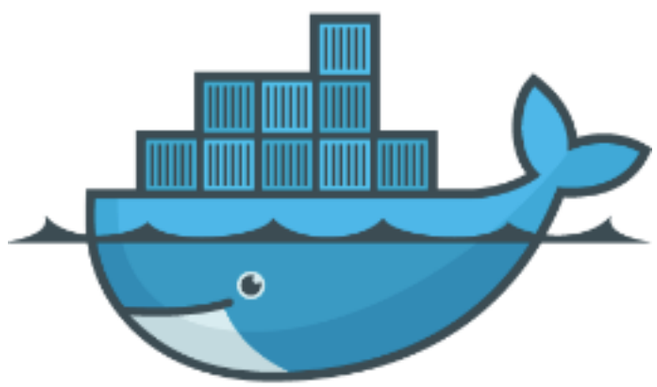
Geoffrey Bachelet – @ubermuda

Container Engine?

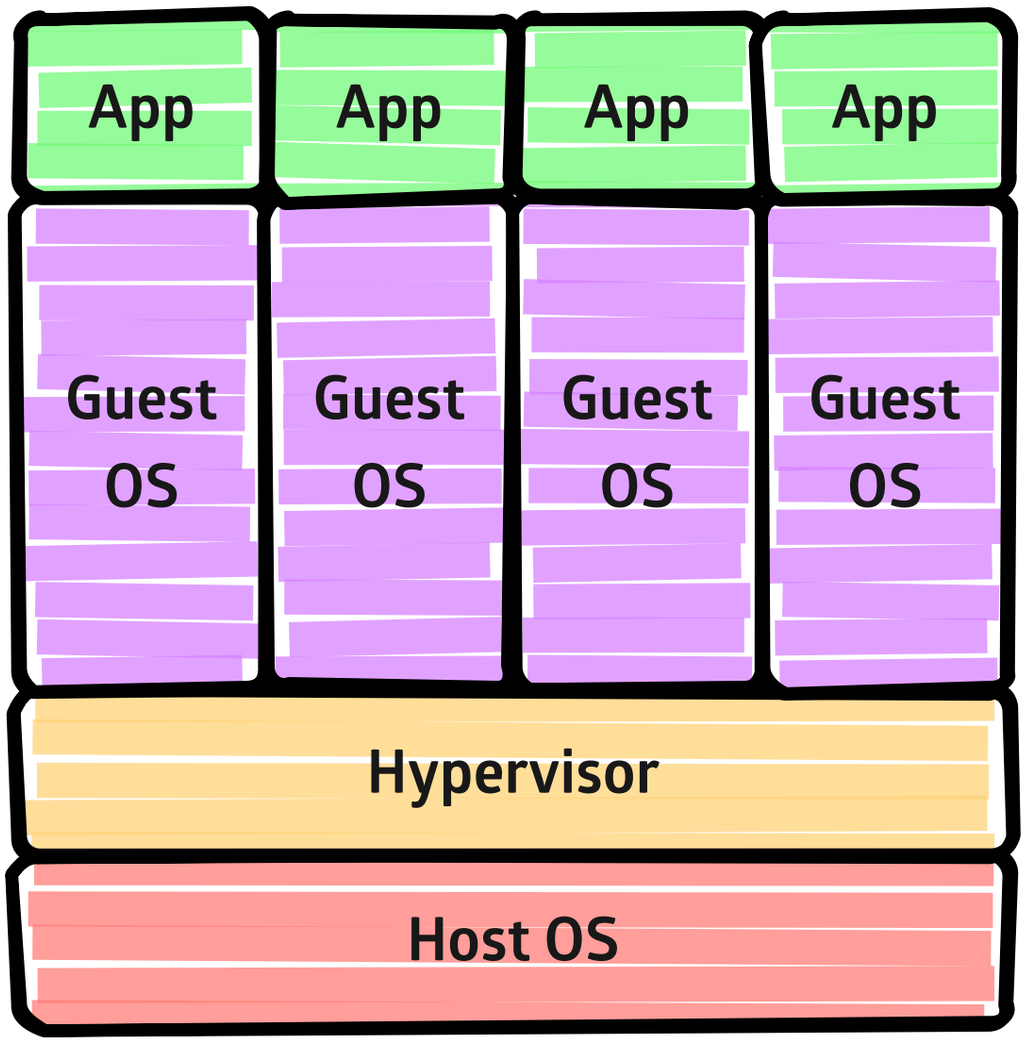
Based on the Linux kernel's **LXC** instructions

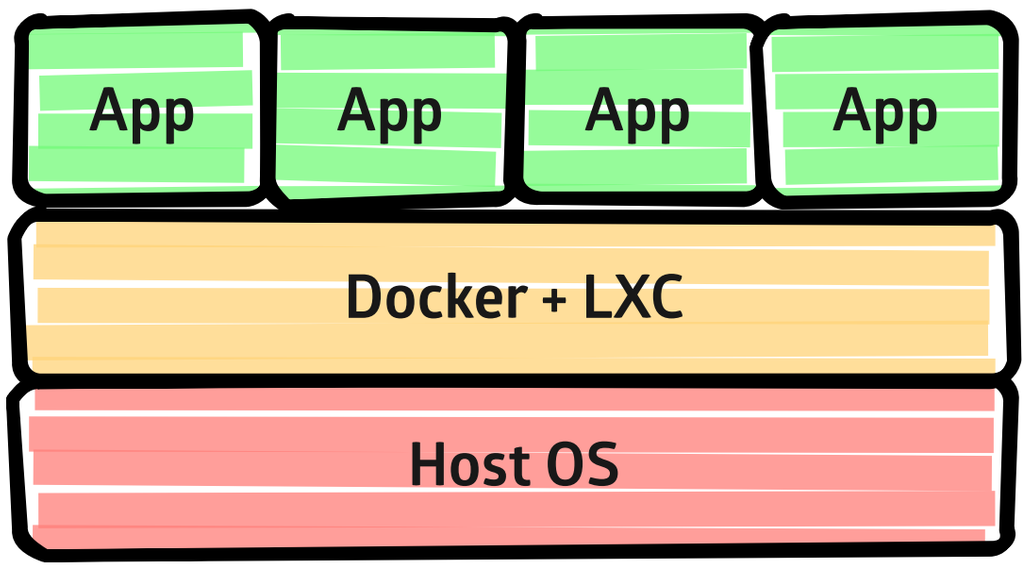
Processes and resources **isolation**

"chroot" on steroids / **super lightweight** VMs

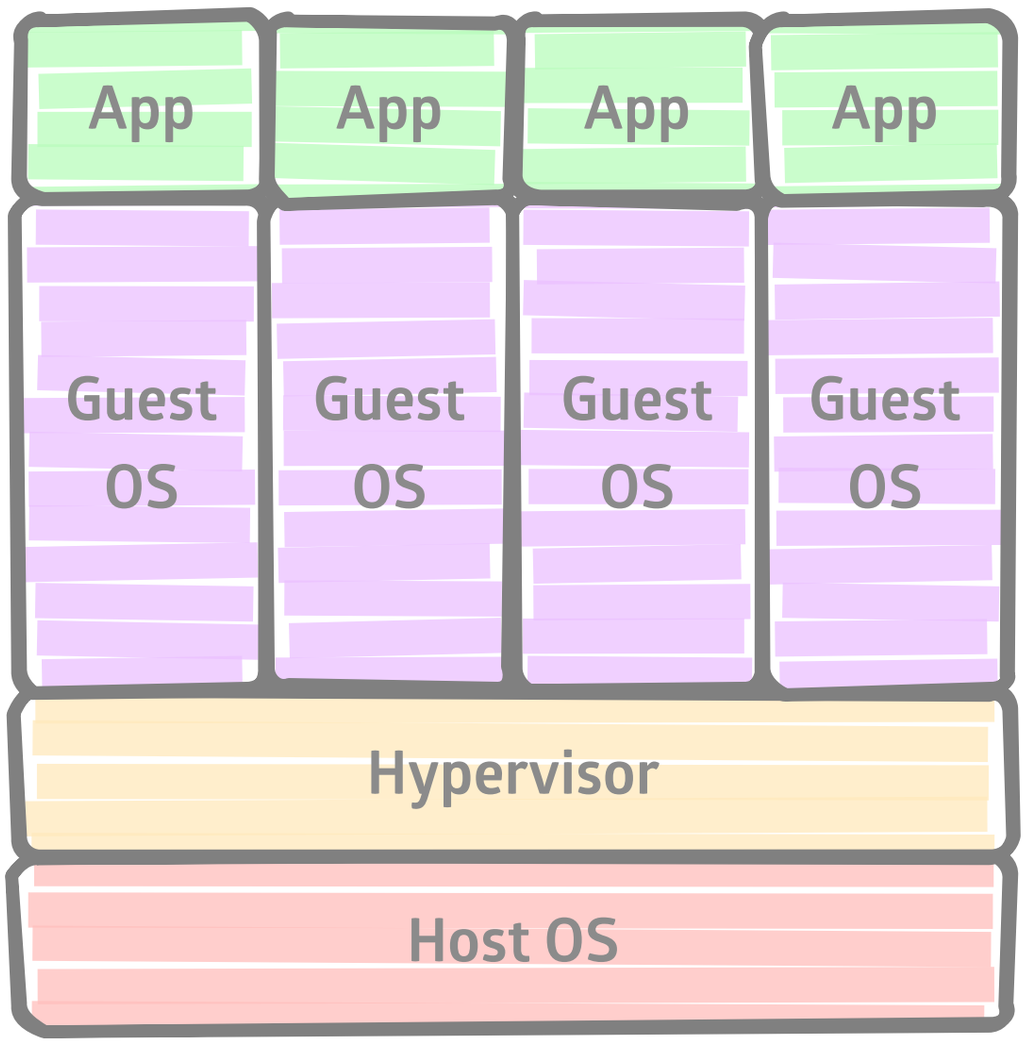


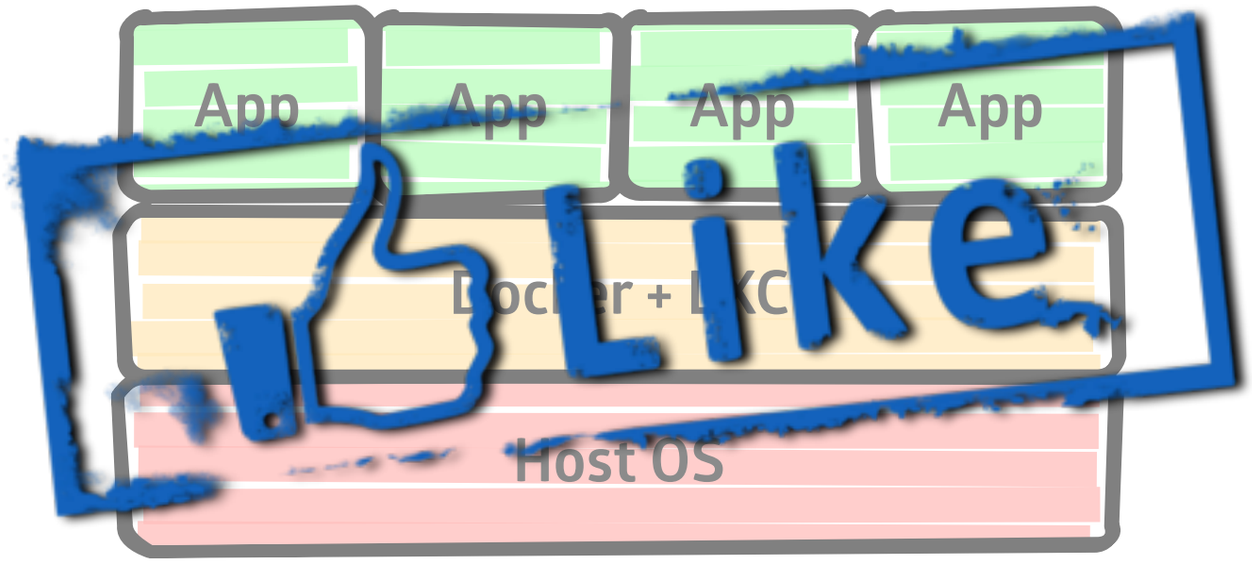
VMs vs Containers





VMs vs Containers





VMs vs Containers

Hypervisor	LXC / Jails / Zones / etc
Boots a complete OS	Uses the host's kernel
Boots in minutes	Boots in seconds
Guest OS' overhead	0 overhead (almost)
Several Go images	Easy to pass around

Install

```
$ echo "deb https://get.docker.io/ubuntu docker main" \
    > /etc/apt/sources.list.d/docker.list
$ apt-get update
$ apt-get install lxc-docker
```

Create a container

```
$ docker pull stackbrew/ubuntu
```

```
$ docker run -i -t stackbrew/ubuntu /bin/bash
```

```
root@21d86a0b8387:/#
```

```
$ docker pull stackbrew/ubuntu
```

```
$ docker run -i -t stackbrew/ubuntu /bin/bash
```

```
root@21d86a0b8387:/#
```

```
$ docker pull stackbrew/ubuntu
```

```
$ docker run -i -t stackbrew/ubuntu /bin/bash

root@21d86a0b8387:/#
```

```
$ docker pull stackbrew/ubuntu
```

```
$ docker run -i -t stackbrew/ubuntu /bin/bash
```

```
root@21d86a0b8387:/#
```

```
$ docker pull stackbrew/ubuntu
```

```
$ docker run -i -t stackbrew/ubuntu /bin/bash
```

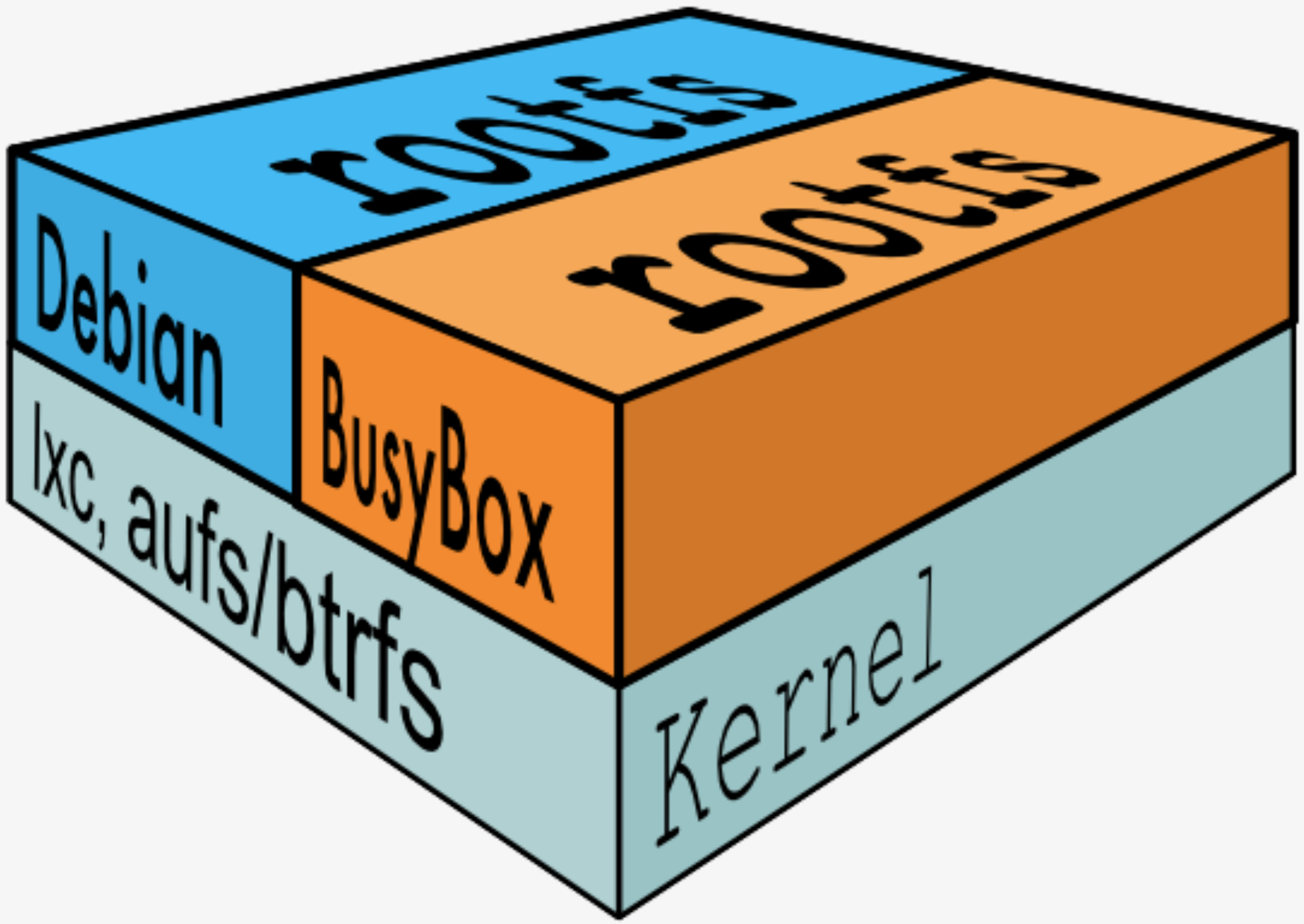
```
root@21d86a0b8387:/#
```

```
$ docker pull stackbrew/ubuntu
```

```
$ docker run -i -t stackbrew/ubuntu /bin/bash
```

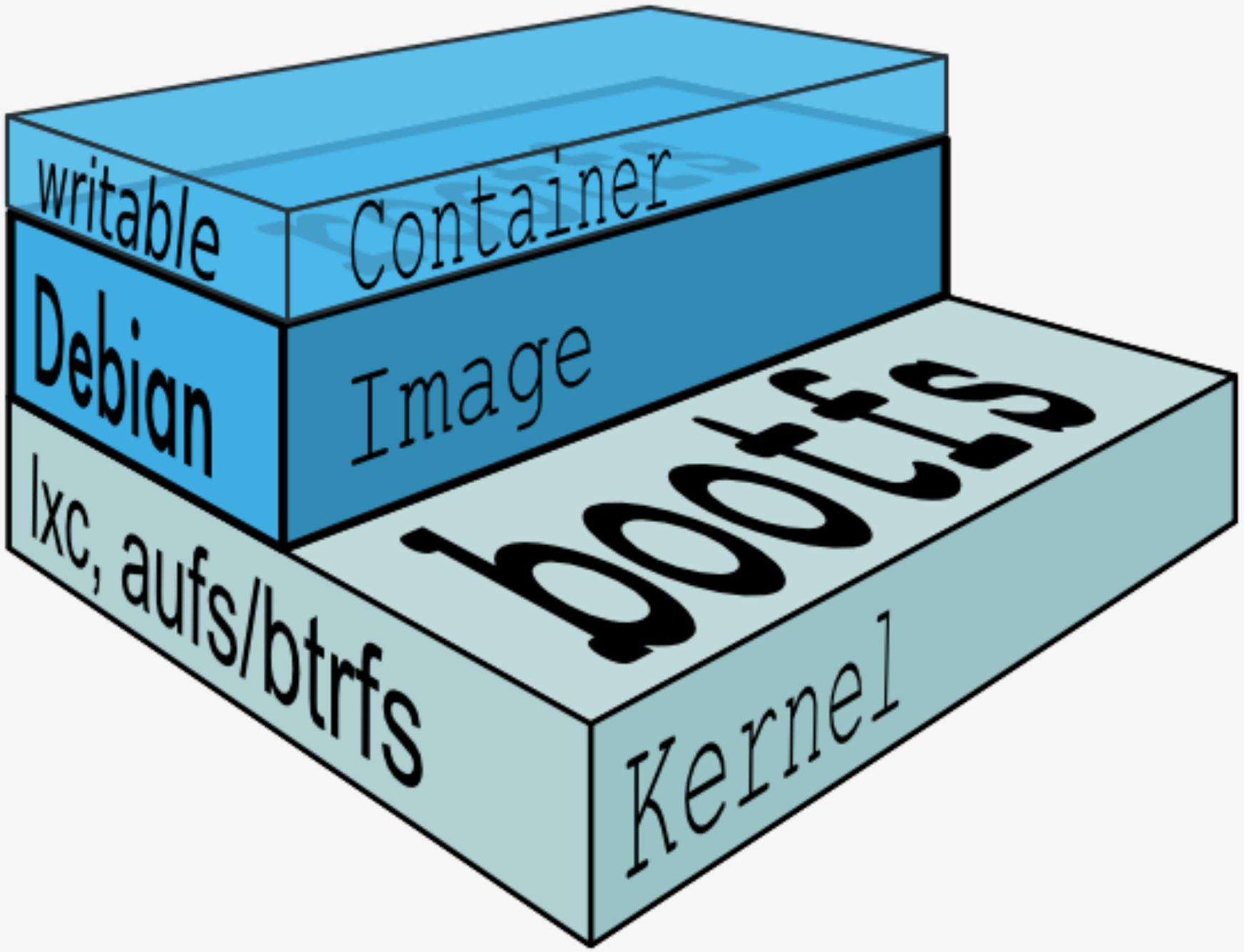
```
root@21d86a0b8387:/#
```

Image



credit: http://docs.docker.io/en/latest/terms/image/

Container



credit: http://docs.docker.io/en/latest/terms/image/

Commit the container

```
root@21d86a0b8387:/# apt-get install nginx

root@21d86a0b8387:/# exit
```

```
$ docker ps -q -n 1

21d86a0b8387
```

```
$ docker commit 21d86a0b8387 sflive/nginx

240198b750c3cc950c60005d6d24cae4fc2dbcc6c31e274574af68d4a2e8
```

```
$ docker images

REPOSITORY      TAG         IMAGE ID

sflive/nginx     latest      240198b750c3
```

```
root@21d86a0b8387:/# apt-get install nginx

root@21d86a0b8387:/# exit
```

```
$ docker ps -q -n 1

21d86a0b8387
```

```
$ docker commit 21d86a0b8387 sflive/nginx

240198b750c3cc950c60005d6d24cae4fc2dbcc6c31e274574af68d4a2e8
```

```
$ docker images

REPOSITORY      TAG         IMAGE ID

sflive/nginx     latest      240198b750c3
```

```
root@21d86a0b8387:/# apt-get install nginx

root@21d86a0b8387:/# exit


$ docker ps -q -n 1
21d86a0b8387

$ docker commit 21d86a0b8387 sflive/nginx
240198b750c3cc950c60005d6d24cae4fc2dbcc6c31e274574af68d4a2e8


$ docker images
REPOSITORY      TAG         IMAGE ID
sflive/nginx     latest      240198b750c3
```

```
root@21d86a0b8387:/# apt-get install nginx

root@21d86a0b8387:/# exit


$ docker ps -q -n 1

21d86a0b8387


$ docker commit 21d86a0b8387 sflive/nginx

240198b750c3cc950c60005d6d24cae4fc2dbcc6c31e274574af68d4a2e8


$ docker images

REPOSITORY      TAG          IMAGE ID

sflive/nginx     latest      240198b750c3
```

```
root@21d86a0b8387:/# apt-get install nginx

root@21d86a0b8387:/# exit


$ docker ps -q -n 1

21d86a0b8387


$ docker commit 21d86a0b8387 sflive/nginx

240198b750c3cc950c60005d6d24cae4fc2dbcc6c31e274574af68d4a2e8


$ docker images

REPOSITORY      TAG         IMAGE ID

sflive/nginx     latest      240198b750c3
```

Run the image

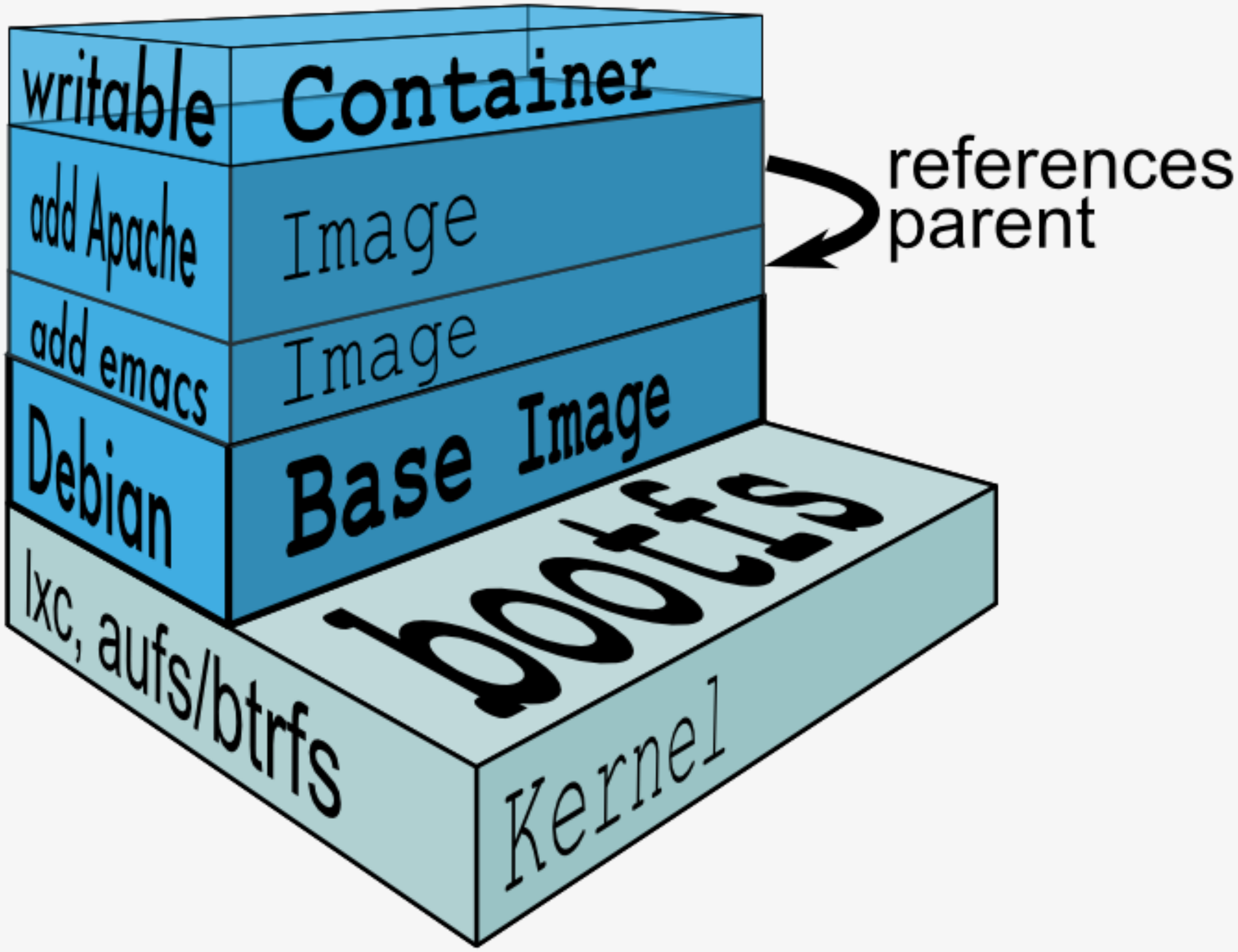
```
$ docker run -p 80 sflive/nginx nginx -g 'daemon off;'
```

```
$ docker ps
CONTAINER ID  ...  PORTS
923cb190dbc3  ...  0.0.0.0:49155->80/tcp

$ curl http://localhost:49155
<html>
<head>
<title>Welcome to nginx!</title>
...
```

```
$ docker ps

CONTAINER ID   ...   PORTS

923cb190dbc3   ...   0.0.0.0:49155->80/tcp
```

```
$ curl http://localhost:49155

<html>

<head>

<title>Welcome to nginx!</title>

...
```

Dockerfile



credit: http://docs.docker.io/en/latest/terms/image/

```
$ cd /project

project $ bin/start

Loading composer repositories with package information

Installing dependencies (including require-dev) from lock file

...
```

Symfony Live - Docker

sflive.dev/

Symfony

Hello, docker!

2.4.2 | php | 5373a3 app | dev | 200 WelcomeController :: indexAction on _welcome | 651 ms | 18.2 MB | 0 | 0

```
FROM sflive/base
ENV DEBIAN_FRONTEND noninteractive

RUN apt-get install -y \
    daemontools curl nginx mysql-server redis-server \
    php5-cli php5-json php5-fpm php5-intl php5-mysqlnd

RUN curl -sS https://getcomposer.org/installer | php
RUN mv composer.phar /usr/local/bin/composer

RUN echo "daemonize=no" > /etc/php5/fpm/pool.d/daemonize.conf
RUN echo "\ndaemon off;" >> /etc/nginx/nginx.conf

ADD vhost.conf /etc/nginx/sites-enabled/default
ADD php.ini /etc/php5/fpm/php.ini
ADD php.ini /etc/php5/cli/php.ini
ADD my.cnf /etc/mysql/my.cnf
ADD services/ /srv/services
ADD entrypoint.sh /usr/local/bin/entrypoint.sh

EXPOSE 80

ENTRYPOINT ["/usr/local/bin/entrypoint.sh"]
```

```
FROM sflive/base

ENV DEBIAN_FRONTEND noninteractive
```

```
RUN apt-get install -y \
    daemontools curl nginx mysql-server redis-server \
    php5-cli php5-json php5-fpm php5-intl php5-mysqlnd

RUN curl -sS https://getcomposer.org/installer | php

RUN mv composer.phar /usr/local/bin/composer
```

```
RUN echo "daemonize=no" \
    > /etc/php5/fpm/pool.d/daemonize.conf

RUN echo "\ndaemon off;" \
    >> /etc/nginx/nginx.conf
```

```
ADD vhost.conf /etc/nginx/sites-enabled/default

ADD php.ini /etc/php5/fpm/php.ini

ADD php.ini /etc/php5/cli/php.ini

ADD my.cnf /etc/mysql/my.cnf

ADD services/ /srv/services


ADD entrypoint.sh /usr/local/bin/entrypoint.sh
```

```
ADD vhost.conf /etc/nginx/sites-enabled/default

ADD php.ini /etc/php5/fpm/php.ini

ADD php.ini /etc/php5/cli/php.ini

ADD my.cnf /etc/mysql/my.cnf

ADD services/ /srv/services


ADD entrypoint.sh /usr/local/bin/entrypoint.sh
```

```
EXPOSE 80

ENTRYPOINT ["/usr/local/bin/entrypoint.sh"]
```

```
symfony2
├── Dockerfile
├── entrypoint.sh
├── my.cnf
├── php.ini
├── services
│   ├── mysql
│   │   └── run
│   ├── nginx
│   │   └── run
│   └── php5-fpm
│       └── run
└── vhost.conf
```

```
symfony2
├── Dockerfile
├── entrypoint.sh
├── my.cnf
├── php.ini
├── services
│   ├── mysql
│   │   └── run
│   ├── nginx
│   │   └── run
│   └── php5-fpm
│       └── run
└── vhost.conf
```

```
symfony2
├── Dockerfile
├── entrypoint.sh
├── my.cnf
├── php.ini
├── services
│   ├── mysql
│   │   └── run
│   ├── nginx
│   │   └── run
│   └── php5-fpm
│       └── run
└── vhost.conf
```

```
#!/bin/bash

exec /usr/bin/mysqld_safe
```

```
#!/bin/bash

exec /usr/bin/nginx
```

```
#!/bin/bash

exec /usr/sbin/php5-fpm
```

```
symfony2
├── Dockerfile
├── entrypoint.sh
├── my.cnf
├── php.ini
├── services
│   ├── mysql
│   │   └── run
│   ├── nginx
│   │   └── run
│   └── php5-fpm
│       └── run
└── vhost.conf
```

```
#!/bin/bash -e

if [ ! -d /var/www ]; then
    echo 'No application found in /var/www'
    exit 1;
fi

cd /var/www

if [ ! -d vendor ]; then
    composer install
fi

if [ -f ./init.sh ]; then
    ./init.sh
fi

exec svscan /srv/services
```

```
symfony2
├── Dockerfile
├── entrypoint.sh
├── my.cnf
├── php.ini
├── services
│   ├── mysql
│   │   └── run
│   ├── nginx
│   │   └── run
│   └── php5-fpm
│       └── run
└── vhost.conf
```

```
#!/bin/bash -e

if [ ! -d /var/www ]; then
    echo 'No application found in /var/www'
    exit 1;
fi

cd /var/www

if [ ! -d vendor ]; then
    composer install
fi

if [ -f ./init.sh ]; then
    ./init.sh
fi

exec svscan /srv/services
```

```
symfony2
├── Dockerfile
├── entrypoint.sh
├── my.cnf
├── php.ini
├── services
│   ├── mysql
│   │   └── run
│   ├── nginx
│   │   └── run
│   └── php5-fpm
│       └── run
└── vhost.conf
```

```
#!/bin/bash -e

if [ ! -d /var/www ]; then
    echo 'No application found in /var/www'
    exit 1;
fi

cd /var/www

if [ ! -d vendor ]; then
    composer install
fi

if [ -f ./init.sh ]; then
    ./init.sh
fi

exec svscan /srv/services
```

```
symfony2
├── Dockerfile
├── entrypoint.sh
├── my.cnf
├── php.ini
├── services
│   ├── mysql
│   │   └── run
│   ├── nginx
│   │   └── run
│   └── php5-fpm
│       └── run
└── vhost.conf
```

```
#!/bin/bash -e

if [ ! -d /var/www ]; then
    echo 'No application found in /var/www'
    exit 1;
fi

cd /var/www

if [ ! -d vendor ]; then
    composer install
fi

if [ -f ./init.sh ]; then
    ./init.sh
fi

exec svscan /srv/services
```

```
symfony2
├── Dockerfile
├── entrypoint.sh
├── my.cnf
├── php.ini
├── services
│   ├── mysql
│   │   └── run
│   ├── nginx
│   │   └── run
│   └── php5-fpm
│       └── run
└── vhost.conf
```

```
#!/bin/bash -e

if [ ! -d /var/www ]; then
    echo 'No application found in /var/www'
    exit 1;
fi

cd /var/www

if [ ! -d vendor ]; then
    composer install
fi

if [ -f ./init.sh ]; then
    ./init.sh
fi

exec svscan /srv/services
```

```
$ docker build -t sflive/symfony2 symfony2/
```

```
$ docker build -t sflive/symfony2 symfony2/
```

```
$ docker run \
    -p 8000:80 \
    -v /project:/var/www \
    sflive/symfony2
```

```
$ docker run \
    -p 8000:80 \
    -v /project:/var/www \
    sflive/symfony2
```

```
$ docker run \
    -p 8000:80 \
    -v /project:/var/www \
    sflive/symfony2
```

bin/start

```
#!/bin/bash

docker run -d \
    -v $(pwd):/var/www \
    -p 80:80 \
    --cidfile app/cache/docker.cid \
    sflive/symfony2
```

bin/stop

```
#!/bin/bash
docker stop $(cat app/cache/docker.cid)
```

```
$ cd /project

project $ bin/start

Loading composer repositories with package information

Installing dependencies (including require-dev) from lock file

...
```

Symfony Live - Docker

sflive.dev/

Symfony

Hello, docker!

2.4.2 | php | 5373a3 app | dev | 200 WelcomeController :: indexAction on _welcome | 651 ms | 18.2 MB | 0 | 0

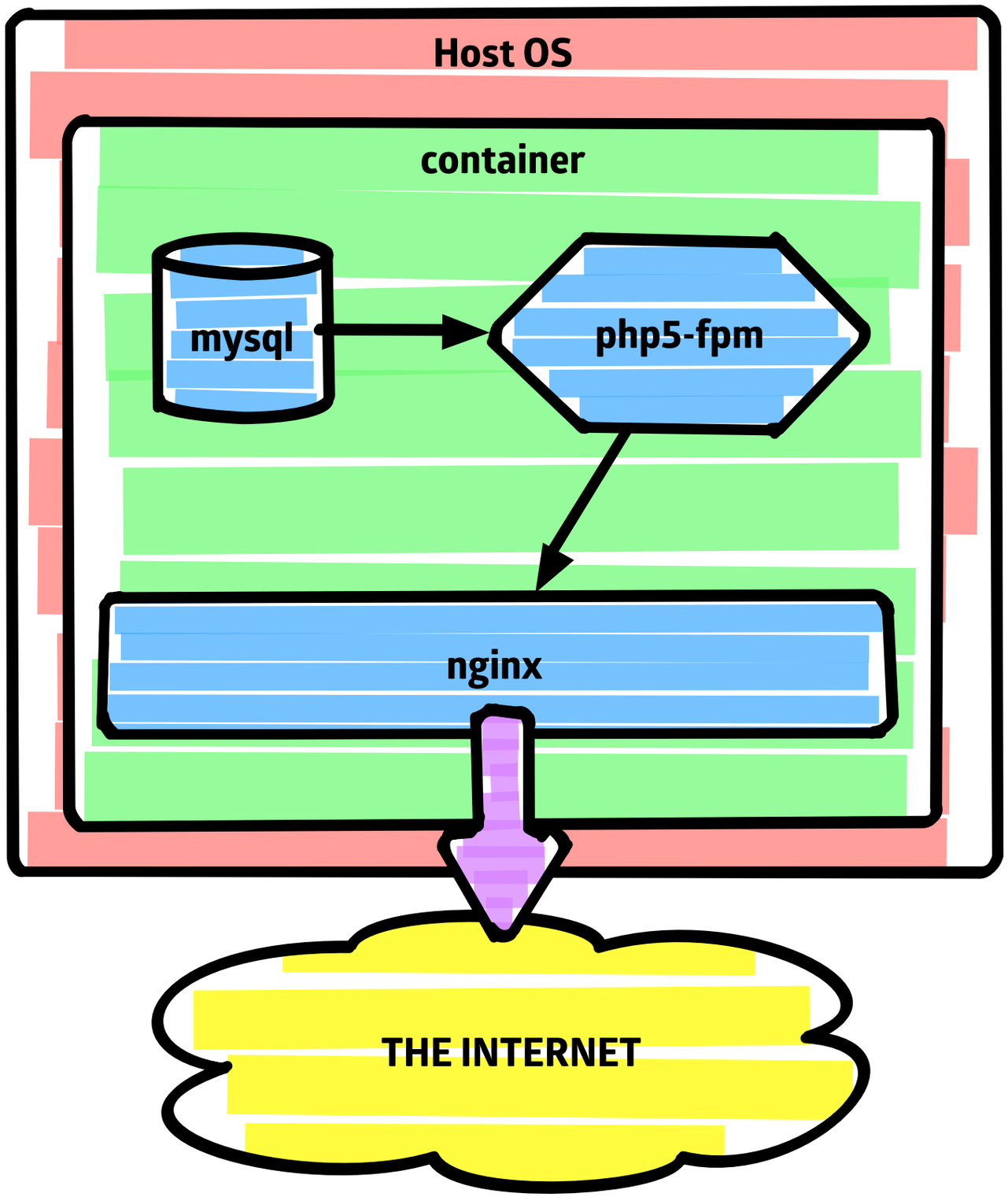
Symfony Live – Docker

sflive.dev/

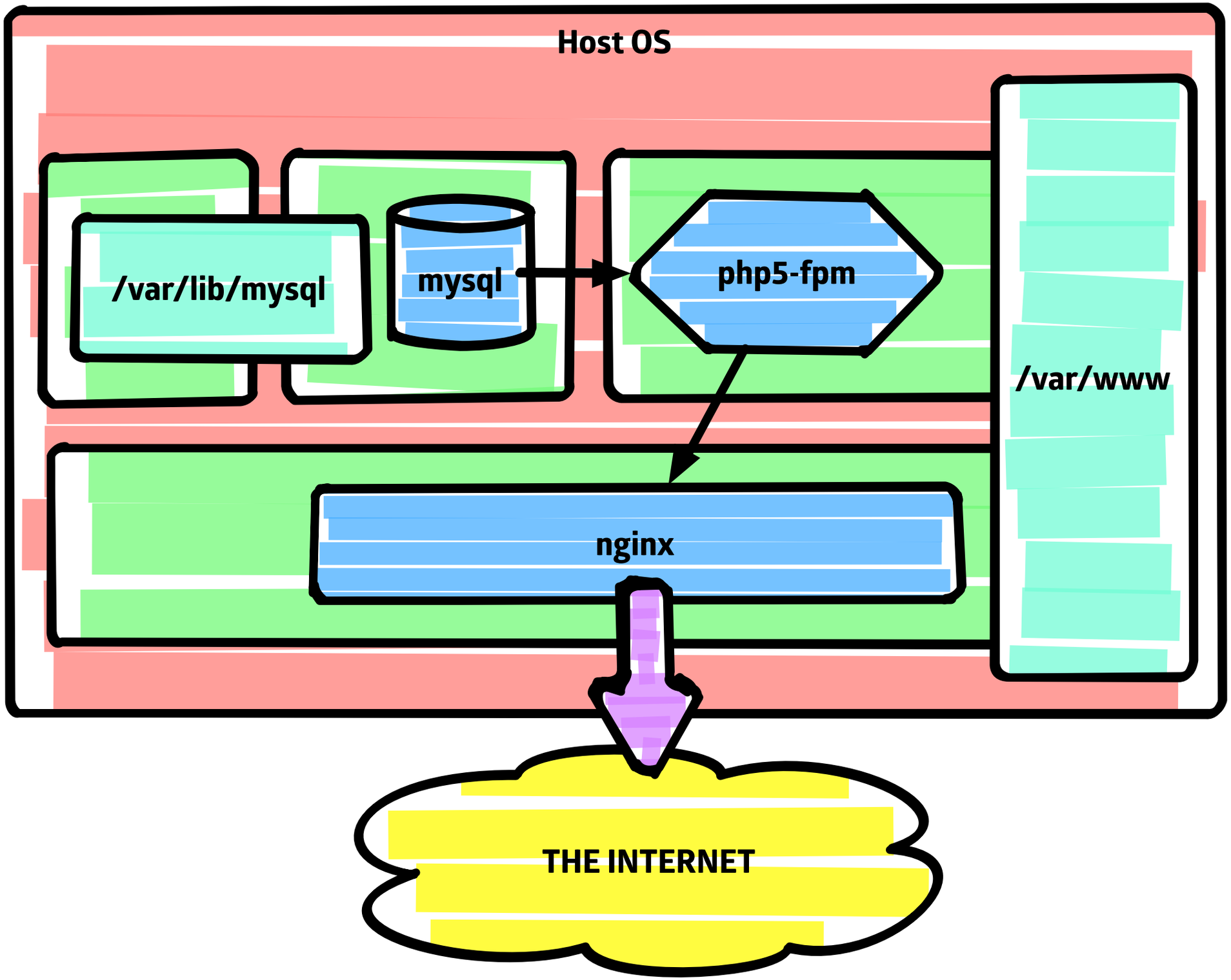
Symfony

Hello, docker!

2.4.2 | 5373a3 app | dev | 200 WelcomeController :: indexAction on _welcome | 651 ms | 18.2 MB | 0 | 0


Host OS
container
mysql
php5-fpm
nginx
THE INTERNET

Multi-container setup


Host OS
/var/lib/mysql
mysql
php5-fpm
/var/www
nginx
THE INTERNET

Containers links

```
$ docker run --name redis sflive/redis
```

```
$ docker run --link redis:foo sflive/symfony2
```

```
$ docker run --name redis sflive/redis
```

```
$ docker run --link redis:foo sflive/symfony2
```

```
$ docker run --name redis sflive/redis
```

```
$ docker run --link redis:foo sflive/symfony2
```

```
$ printenv | grep -E '^FOO'
FOO_PORT=tcp://172.17.0.44:6379
FOO_PORT_6379_TCP_PROTO=tcp
FOO_PORT_6379_TCP_PORT=6379
FOO_PORT_6379_TCP=tcp://172.17.0.44:6379
FOO_NAME=/crimson_squirrel9/redis
FOO_PORT_6379_TCP_ADDR=172.17.0.44
```

```
$ printenv | grep -E '^FOO'
FOO_PORT=tcp://172.17.0.44:6379
FOO_PORT_6379_TCP_PROTO=tcp
FOO_PORT_6379_TCP_PORT=6379
FOO_PORT_6379_TCP=tcp://172.17.0.44:6379
FOO_NAME=/crimson_squirrel9/redis
FOO_PORT_6379_TCP_ADDR=172.17.0.44
```

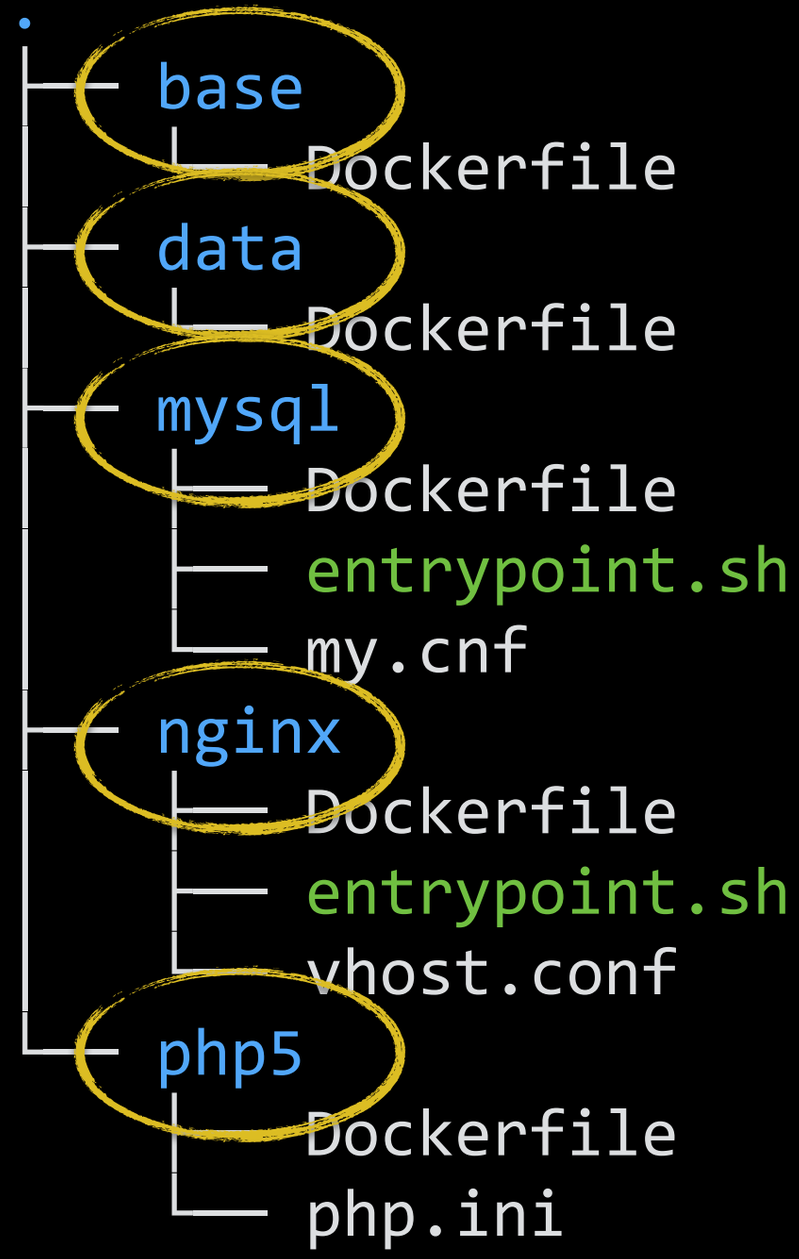
app/config/config.yml

```
imports:
    - { resource: parameters.php }
```

app/config/parameters.php

```
<?php

$container->setParameter(
    'redis_host',
    getenv('FOO_PORT_6379_TCP_ADDR')
);
```

```
.
├── base
│   └── Dockerfile
├── data
│   └── Dockerfile
├── mysql
│   ├── Dockerfile
│   ├── entrypoint.sh
│   └── my.cnf
├── nginx
│   ├── Dockerfile
│   ├── entrypoint.sh
│   └── vhost.conf
└── php5
    ├── Dockerfile
    └── php.ini
```


.
base
Dockerfile
data
Dockerfile
mysql
Dockerfile
entrypoint.sh
my.cnf
nginx
Dockerfile
entrypoint.sh
vhost.conf
php5
Dockerfile
php.ini

```
.
├── base
│   └── Dockerfile
├── data
│   └── Dockerfile
├── mysql
│   ├── Dockerfile
│   ├── entrypoint.sh
│   └── my.cnf
├── nginx
│   ├── Dockerfile
│   ├── entrypoint.sh
│   └── vhost.conf
└── php5
    ├── Dockerfile
    └── php.ini
```

```
FROM sflive/base

VOLUME ["/var/lib/mysql"]

ENTRYPOINT ["true"]
```

```
.
├── base
│   └── Dockerfile
├── data
│   └── Dockerfile
├── mysql
│   ├── Dockerfile
│   ├── entrypoint.sh
│   └── my.cnf
├── nginx
│   ├── Dockerfile
│   ├── entrypoint.sh
│   └── vhost.conf
└── php5
    ├── Dockerfile
    └── php.ini
```

```
FROM sflive/base

VOLUME ["/var/lib/mysql"]

ENTRYPOINT ["true"]
```

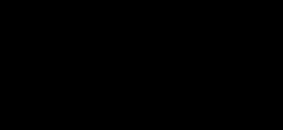
```
.
├── base
│   └── Dockerfile
├── data
│   └── Dockerfile
├── mysql
│   ├── Dockerfile
│   ├── entrypoint.sh
│   └── my.cnf
├── nginx
│   ├── Dockerfile
│   ├── entrypoint.sh
│   └── vhost.conf
└── php5
    ├── Dockerfile
    └── php.ini
```

```
#!/bin/bash


if [ ! -f /var/lib/mysql/ibdata1 ]; then

    mysql_install_db > /dev/null

fi


exec /usr/bin/mysqld_safe
```

```
.
├── base
│   └── Dockerfile
├── data
│   └── Dockerfile
├── mysql
│   ├── Dockerfile
│   ├── entrypoint.sh
│   └── my.cnf
├── nginx
│   ├── Dockerfile
│   ├── entrypoint.sh
│   └── vhost.conf
└── php5
    ├── Dockerfile
    └── php.ini
```

```
#!/bin/bash

if [ ! -f /var/lib/mysql/ibdata1 ]; then

    mysql_install_db > /dev/null

fi

exec /usr/bin/mysqld_safe
```

```
.
├── base
│   └── Dockerfile
├── data
│   └── Dockerfile
├── mysql
│   ├── Dockerfile
│   ├── entrypoint.sh
│   └── my.cnf
├── nginx
│   ├── Dockerfile
│   ├── entrypoint.sh
│   └── vhost.conf
└── php5
    ├── Dockerfile
    └── php.ini
```



bind-address = 0.0.0.0

```
.
├── base
│   └── Dockerfile
├── data
│   └── Dockerfile
├── mysql
│   ├── Dockerfile
│   ├── entrypoint.sh
│   └── my.cnf
├── nginx
│   ├── Dockerfile
│   ├── entrypoint.sh
│   └── vhost.conf
└── php5
    ├── Dockerfile
    └── php.ini
```

```
#!/bin/bash

for env in $(printenv); do

    IFS== read KEY VALUE <<< "$env"

    sed \

        -e "s,\${${KEY}},${VALUE},g" \

        -i /etc/nginx/sites-enabled/default

done;


exec /usr/sbin/nginx
```

```
.
├── base
│   └── Dockerfile
├── data
│   └── Dockerfile
├── mysql
│   ├── Dockerfile
│   ├── entrypoint.sh
│   └── my.cnf
├── nginx
│   ├── Dockerfile
│   ├── entrypoint.sh
│   └── vhost.conf
└── php5
    ├── Dockerfile
    └── php.ini
```

```
location ~ [^/]\.php(/|$) {
    fastcgi_pass ${PHP5_PORT_9000_TCP_ADDR}:9000;
    include fastcgi_params;
}
```

```
.
├── base
│   └── Dockerfile
├── data
│   └── Dockerfile
├── mysql
│   ├── Dockerfile
│   ├── entrypoint.sh
│   └── my.cnf
├── nginx
│   ├── Dockerfile
│   ├── entrypoint.sh
│   └── vhost.conf
└── php5
    ├── Dockerfile
    └── php.ini
```

```
location ~ [^/]\.php(/|$) {
    fastcgi_pass ${PHP5_PORT_9000_TCP_ADDR}:9000;
    include fastcgi_params;
}
```

```
.
├── base
│   └── Dockerfile
├── data
│   └── Dockerfile
├── mysql
│   ├── Dockerfile
│   ├── entrypoint.sh
│   └── my.cnf
├── nginx
│   ├── Dockerfile
│   ├── entrypoint.sh
│   └── vhost.conf
└── php5
    ├── Dockerfile
    └── php.ini
```

```
RUN sed \
    -e "s,127.0.0.1:9000,9000," \
    -i /etc/php5/fpm/pool.d/www.conf
```

```
.
├── base
│   └── Dockerfile
├── data
│   └── Dockerfile
├── mysql
│   ├── Dockerfile
│   ├── entrypoint.sh
│   └── my.cnf
├── nginx
│   ├── Dockerfile
│   ├── entrypoint.sh
│   └── vhost.conf
└── php5
    ├── Dockerfile
    └── php.ini
```

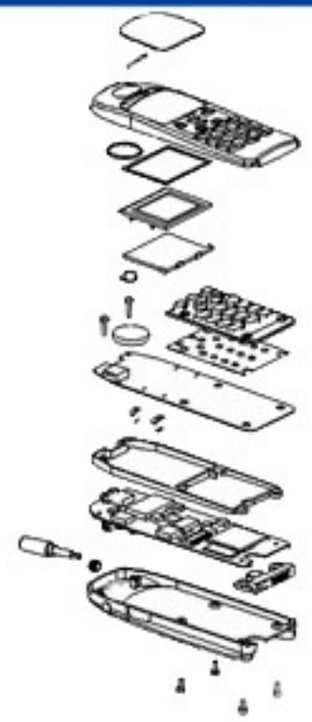
```
RUN sed \
    -e "s,127.0.0.1:9000,9000," \
    -i /etc/php5/fpm/pool.d/www.conf
```

```
$ docker build -t sflive/base base/

$ docker build -t sflive/data data/

$ docker build -t sflive/mysql mysql/

$ docker build -t sflive/nginx nginx/

$ docker build -t sflive/php5 php5/
```

```
#!/bin/bash -e

for name in $(find -maxdepth 1 ! -path . -type d); do
    docker build -t sflive/$(basename $name) --rm $name
done;
```

IKEA

Some assembly required



```
$ docker run -name data sflive/data

$ docker run --volumes-from data -name mysql sflive/mysql

$ docker run -v $(pwd):/var/www -name php5 sflive/php5

$ docker run -p 80:80 -v $(pwd):/var/www \
    -link php5:php5 \
    -link mysql:mysql \
    sflive/nginx
```

```
$ docker run -name data sflive/data

$ docker run --volumes-from data -name mysql sflive/mysql

$ docker run -v $(pwd):/var/www -name php5 sflive/php5

$ docker run -p 80:80 -v $(pwd):/var/www \
    -link php5:php5 \
    -link mysql:mysql \
    sflive/nginx
```

```
$ docker run -name data sflive/data
```

```
$ docker run --volumes-from data -name mysql sflive/mysql
```

```
$ docker run -v $(pwd):/var/www -name php5 sflive/php5
```

```
$ docker run -p 80:80 -v $(pwd):/var/www \
    -link php5:php5 \
    -link mysql:mysql \
    sflive/nginx
```

Symfony Live - Docker

sflive.dev/

Symfony

Hello, docker!

2.4.2 | php | 5373a3 app | dev | 200 WelcomeController :: indexAction on _welcome | 651 ms | 18.2 MB | 0 | 0

Symfony Live – Docker

sflive.dev/

2.4.2 | 5373a3 app | dev | 200 WelcomeController :: indexAction on _welcome | 651 ms | 18.2 MB | 0 | 0

And more!

- docker run -d, attach, logs, top, …
- Orchestration (Gaudi, Fig, …)
- Docker Index
- Docker Remote API (dockerode, Docker-PHP, …)
- Dockerfile: USER, WORKDIR, ONBUILD, …
- …

Recap.

- **Image:** like a VM image. Contains the hard-drive (rootfs) and some configuration.

- **Container:** a running instance of an **image**.

Recap.

- You can **commit** a terminated **container**, and you get a re-usable **image** representing the state of that container.

- **Volumes** are like shared directories. **Containers** can share zero or many volumes.

- **Containers** can be **linked** to one another.

Thanks!

https://github.com/ubermuda/sflive-docker/

Geoffrey Bachelet – @ubermuda