

How we use docker for our development environment





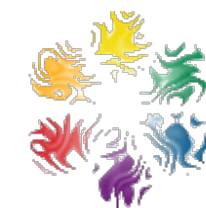
What to expect from this talk

- Docker: Quick introduction
- How we run it on non-Linux-based operating systems
- How we use it

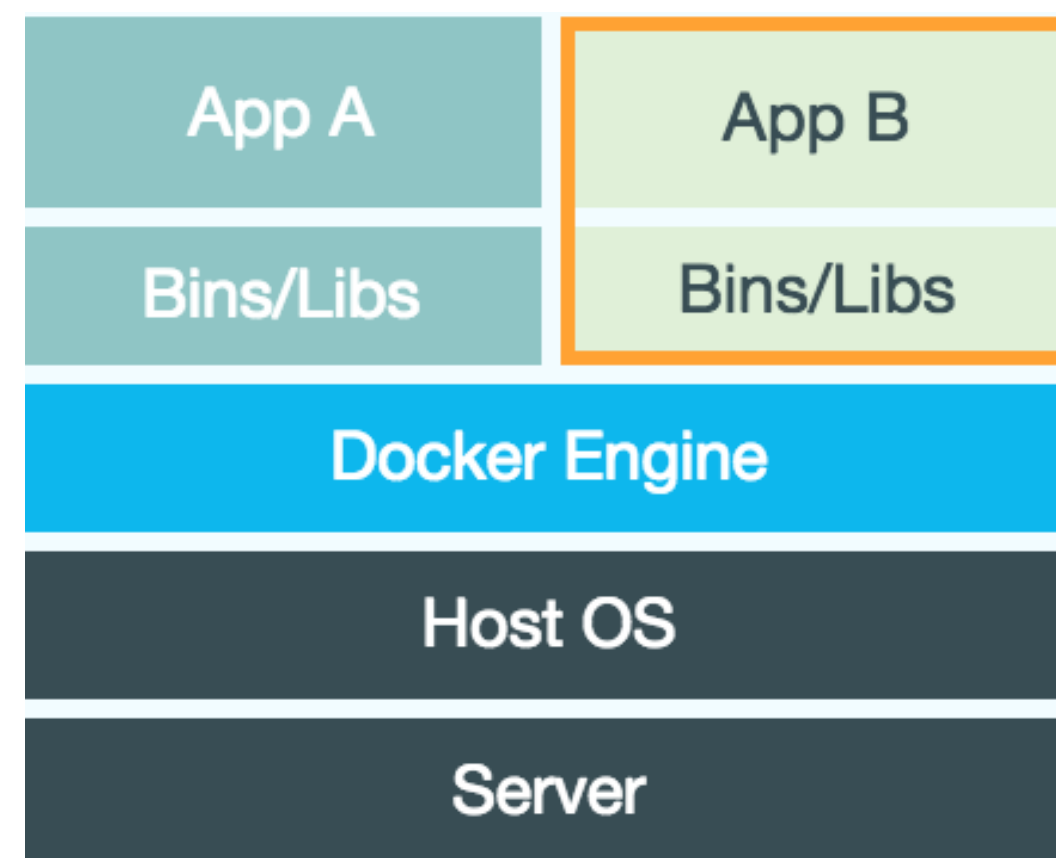
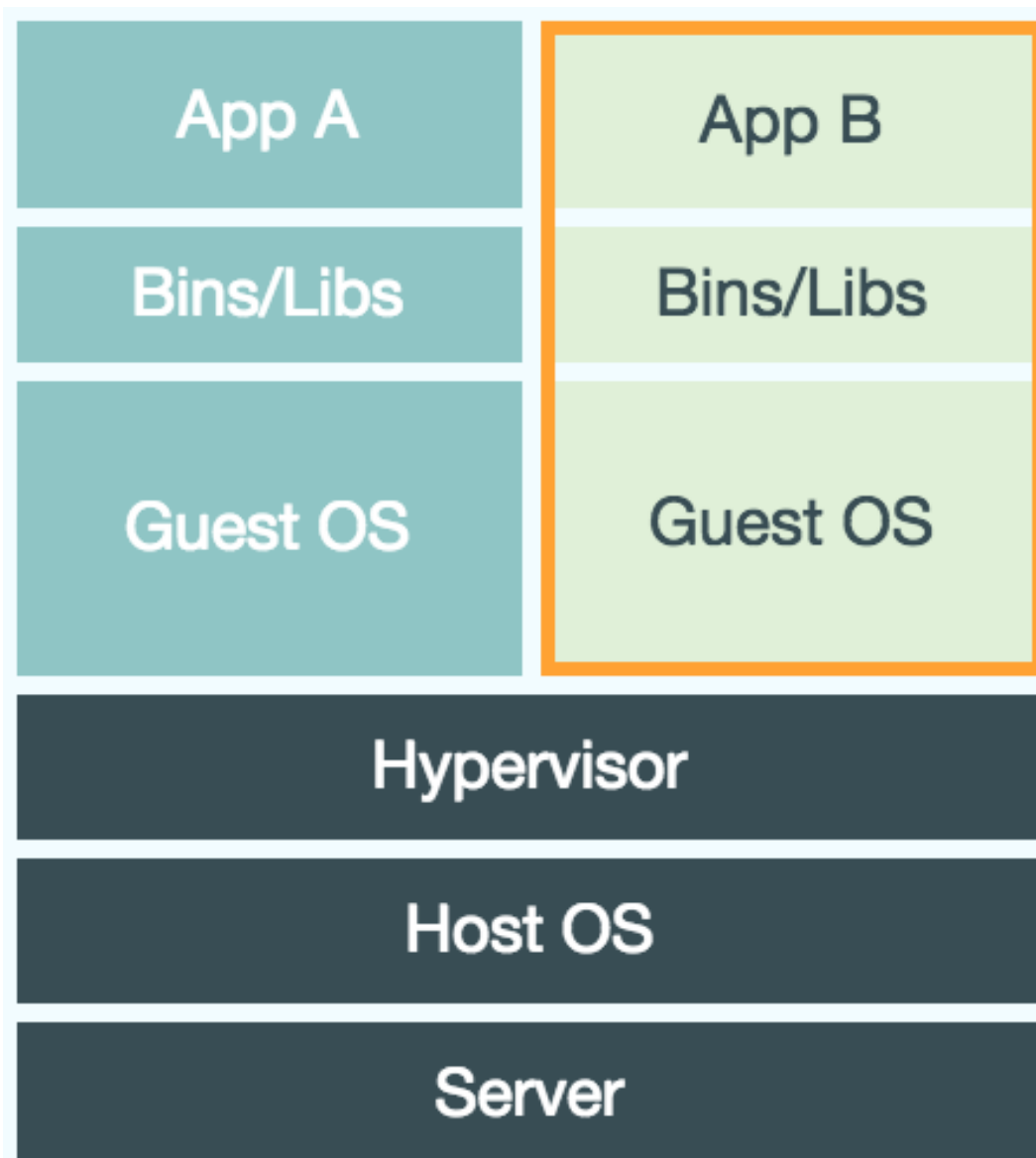


What is Docker?

- Container Engine using Linux kernel features
 - Namespaces (pid, net, mnt, ...)
 - Control Groups (e.g. limit memory)
 - Union file systems
 - Container format (libcontainer, LXC, Jails*, Zones*)
- Client-Server architecture
- Written in Go



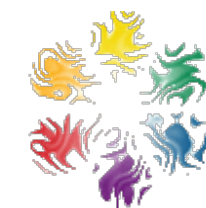
VM vs Containers





How does Docker work?

- Docker Daemon
- Docker Client
- Docker Images
- Docker Containers
- Docker Registry (Docker Hub)



Docker Image - Dockerfile

```
FROM debian:jessie
```

```
RUN apt-get update
```

```
RUN apt-get install -y php5-cli php5-json curl
```

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

```
VOLUME ["/srv"]
```

```
WORKDIR /srv
```

```
ENTRYPOINT ["/composer.phar"]
```



Docker Image - Dockerfile

```
docker build --rm -t adlogix/composer.
```



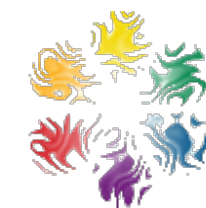
Run the container

```
docker run --rm -i -t -v $PWD:/srv adlogix/composer install
```



How to run on non-Linux-based OS

- Boot2Docker
- Vagrant with Ubuntu/CentOS/Debian ...
- Vagrant Boot2Docker Box



Boot2Docker VS Vagrant B2D Box

	Boot2Docker	Vagrant B2D Box
Setup	Extremely Easy	Less Easy :)
Start VM	boot2docker start	vagrant up
Stop VM	boot2docker stop	vagrant halt
Sharing	vboxsf	nfs



Vagrant - Docker - Architecture

MAC OS X



docker
Client



VAGRANT

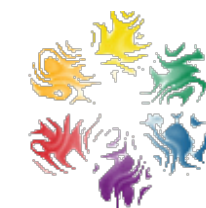
VM



docker
Client
Daemon



docker
Containers



Vagrantfile

```
8 ▼ Vagrant.configure(2) do |config|
9
10   # Make sure same user and group as host is used
11   config.nfs.map_uid = Process.uid
12   config.nfs.map_gid = Process.gid
13
14   config.vm.box = "dduportal/boot2docker"
15
16   config.vm.network "private_network", ip: "192.168.10.10"
17
18   config.vm.synced_folder "/Users", "/Users", type: "nfs", mount_options: ["nolock", "vers=3", "udp"]
19   config.vm.synced_folder ".", "/vagrant"
20
21   # Change virtualbox options
22 ▼ config.vm.provider "virtualbox" do |v|
23     v.memory = 2048
24     v.cpus = 8
25 end
26
```



.profile

```
export DOCKER_HOST=tcp://127.0.0.1:2376  
export DOCKER_CERT_PATH=/Users/toni/dev-environment/tls  
export DOCKER_TLS_VERIFY=1
```



How we use it

- Adlogix Swissknife
- Composer with private repos
- Mysql
- RabbitMQ
- Nginx Reverse Proxy
- PHP-cli / PHP-Apache
- Phpmyadmin
- symfony1 / symfony2



Adlogix Swissknife

- Java
- Ant
- mysql-client
- composer





Composer with private repos

- The problem
 - It needs SSH Key-Based authentication
 - SSH Agent Forwarding natively not supported
- Solutions
 - Share SSH_AUTH_SOCKET
 - ADD your private key in the image
 - Custom entrypoint script
 - Create a volume container with your private key



Composer with private repos

```
docker run \  
  --name ssh-data \  
  -v /root/.ssh \  
  -v ${USER_PRIVATE_KEY}:/root/.ssh/id_rsa \  
  dockerfile/ubuntu \  
  sh -c 'chown -R root:root ~/.ssh && chmod -R  
400 ~/.ssh'
```

```
docker create \  
  --name composer-data \  
  -v /root/.composer \  
  dockerfile/ubuntu
```



Composer with private repos

```
docker run -it --rm \  
  -v $PWD:/src \  
  --volumes-from composer-data \  
  --volumes-from ssh-data \  
  adlogix/swissknife composer $@
```



Mysql

```
docker create \  
    --name mysql55-data \  
    -v /var/lib/mysql \  
    dockerfile/ubuntu
```



Mysql

```
docker run \  
    -e "TZ=Europe/Brussels" \  
    --name="mysql55" \  
    --volumes-from mysql55-data \  
    -p 0.0.0.0:3306:3306 \  
    -e MYSQL_ROOT_PASSWORD=dev \  
    -d \  
mysql:5.5
```



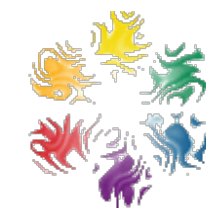
RabbitMQ

```
docker create \  
    --name rabbitmq-data \  
    -v /var/lib/rabbitmq \  
    dockerfile/ubuntu
```



RabbitMQ

```
docker run \  
    -d \  
    -p 5672:5672 -p 15672:15672 \  
    --name="rabbitmq" \  
    --hostname="dev-rabbit" \  
    --volumes-from rabbitmq-data \  
    rabbitmq:management
```



PHP CLI

```
FROM php:5.4-cli
```

```
RUN \
    apt-get update -qq && \
    apt-get install -yqq libmcrypt-dev zlib1g-dev
```

```
RUN \
    docker-php-ext-install \
        bcmath \
        mcrypt \
        pdo_mysql \
        mbstring \
        mysqli \
        zip && \
    pecl install xdebug
```

```
ADD config/php.ini /usr/local/etc/php/php.ini
```

```
WORKDIR /src
```



PHP CLI

```
docker run \  
  --rm \  
  -ti \  
  -v $PWD:/src \  
  adlogix/php-cli:5.4 php $@
```



PHP APACHE

```
FROM php:5.4-apache
```

```
RUN \
    apt-get update -qq && \
    apt-get install -yqq libmcrypt-dev zlib1g-dev
a2enmod rewrite
```

```
...
```

```
RUN \
    echo "IncludeOptional sites-enabled/*.conf" >> /etc/apache2/
apache2.conf
```

```
ADD config/php.ini /usr/local/etc/php/php.ini
ADD 000-default.conf /etc/apache2/sites-enabled/000-default.conf
```

```
WORKDIR /www
```



NGinx Reverse Proxy

```
docker run \  
    -d \  
    -p 80:80 \  
    --name nginx-proxy \  
    -v /var/run/docker.sock:/tmp/docker.sock \  
    jwilder/nginx-proxy
```



phpMyAdmin

```
FROM adlogix/php-apache:5.4
```

```
ENV PHPMYADMIN_VERSION 4.3.9
```

```
RUN \  
  apt-get update -qq && \  
  apt-get install -yqq wget bzip2
```

```
RUN \  
  wget http://downloads.sourceforge.net/project/phpmyadmin/phpMyAdmin/$  
{PHPMYADMIN_VERSION}/phpMyAdmin-${PHPMYADMIN_VERSION}-english.tar.bz2 && \  
  tar -xvjf phpMyAdmin-${PHPMYADMIN_VERSION}-english.tar.bz2 --strip 1 && \  
  rm phpMyAdmin-${PHPMYADMIN_VERSION}-english.tar.bz2
```

```
ADD config/config.inc.php /www/
```

```
ADD config/start-phpmyadmin /usr/local/bin/
```

```
RUN chmod +x /usr/local/bin/start-phpmyadmin
```

```
CMD start-phpmyadmin
```



phpMyAdmin

```
docker run \  
    -ti \  
    -d \  
    --name phpmyadmin \  
    -e "VIRTUAL_HOST=phpmyadmin.dev" \  
    --link mysql55:mysql \  
    adlogix/phpmyadmin
```



symfony2

```
FROM adlogix/php-apache:5.4
```

```
ADD 000-default.conf /etc/apache2/sites-enabled/  
000-default.conf
```



symfony2

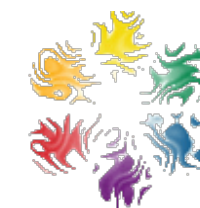
```
<VirtualHost *:80>

    DocumentRoot /www/web
    DirectoryIndex app.php

    <Directory "/www/web">
        RewriteEngine On
        RewriteBase /
        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteRule ^(.*)$ app.php [QSA,L]
        AllowOverride All
        Allow from All
        Require all granted
    </Directory>

    ErrorLog /var/log/error.log
    CustomLog /var/log/access.log combined

</VirtualHost>
```



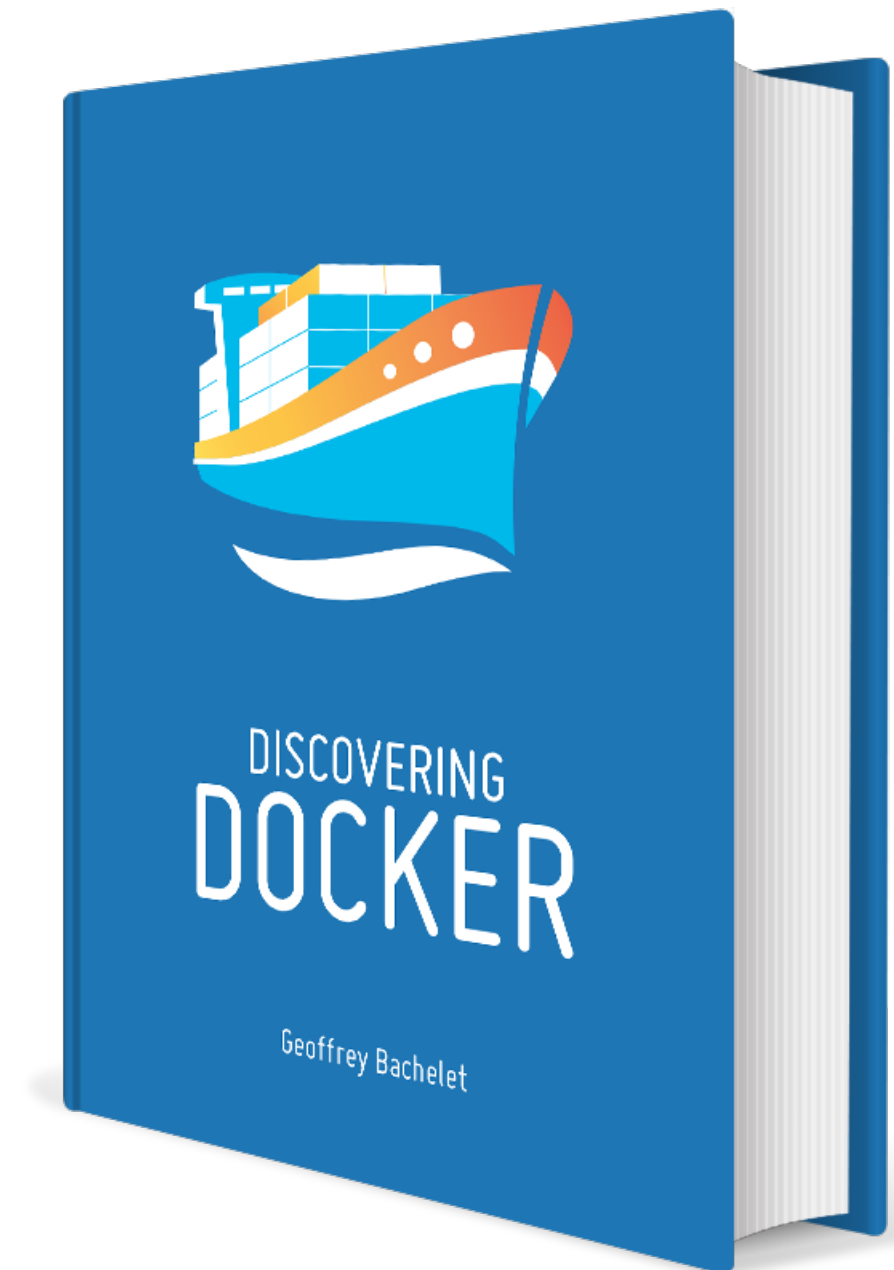
Questions?



Recommended Book

<https://discoveringdocker.com/>

by Geoffrey Bachelet





THANKS!

Please rate this talk:

<https://joind.in/13684>



@tonivdv



tonivdv



www.devexp.eu