Build your dev environment with Docker
What does your dev environment look like?
What does your production environment look like?
add_action( 'init', function () {
    remove_post_type_support( 'post', 'editor' );
}, 99 );
$result = $this->multidimensional( &$root, $keys, true );

- https://core.trac.wordpress.org/ticket/20163

Fatal error: Call-time pass-by-reference has been removed

PHP 5.4
$wpdb->query(
    "DELETE FROM $wpdb->posts WHERE ID = $a_post_id"
);
$post = get_post( $a_post_id );
add_action( 'shutdown', function() {
    fastcgi_finish_request();
    start_a_background_process();
}, 100, 0 );

Fatal error: Call to undefined function fastcgi_finish_request()
What is Docker?
Docker is an open platform for developers and sysadmins to build, ship, and run distributed applications.

— https://www.docker.com/whatisdocker/
A framework for building and running applications using containers
A framework for building and running applications using containers
What is a container?

Virtual machine = Hardware isolation

Container = Process isolation
Let's Start

- **Install Docker (with boot2docker)**
  - [https://docs.docker.com/installation/mac/](https://docs.docker.com/installation/mac/)
  - [https://docs.docker.com/installation/windows/](https://docs.docker.com/installation/windows/)

- **Install Compose**
  - [https://docs.docker.com/compose/install/](https://docs.docker.com/compose/install/)

- **Turn it on**
  - `boot2docker init`
  - `boot2docker up`
  - `eval "$(boot2docker shellinit)"`
docker run hello-world
$ docker run hello-world

Hello from Docker.
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub. (Assuming it was not already locally available.)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

For more examples and ideas, visit:
http://docs.docker.com/usetrguide/
1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.

https://registry.hub.docker.com/_/hello-world/
Containers for a WP Dev Environment

- Apache / nginx
- PHP (multiple versions)
- MySQL
- Memcached
Memcached

- docker run --rm --name memcached -p 11211:11211 memcached:1.4

- telnet $(boot2docker ip) 11211
MySQL

- `docker create -v /var/lib/mysql --name mysqldata busybox /bin/true`

- `docker run --rm --volumes-from mysqldata --name mysql -p 3306:3306 mysql:5.5`
- docker run --rm \
  -v ~/project:/var/www/html \
  --name php \
  --link mysql:mysql \
  --link memcached:memcached \
  php:5.5-fpm
FROM php:5.5-fpm

# required libs for extensions
RUN apt-get update && apt-get install -y \
    build-essential wget libmemcached-dev \
    zlib1g-dev libpng12-dev less \
    && rm -rf /var/lib/apt/lists/*

# memcached extension
RUN echo '' | pecl install memcached
RUN echo "extension=memcached.so" > /usr/local/etc/php/conf.d/ext-memcached.ini

# xdebug extension
RUN pecl install xdebug
COPY ext-xdebug.ini /usr/local/etc/php/conf.d/ext-xdebug.ini

# etc.
jbrinley/php:5.5-fpm

- docker build \
  -t jbrinley/php:5.5-fpm \
  path/to/php/5.5-fpm
## Automated Build Repository

### jbrinley / php

PHP containers built with WordPress development in mind

<table>
<thead>
<tr>
<th>Information</th>
<th>Dockerfile</th>
<th>Build Details</th>
<th>Tags</th>
</tr>
</thead>
</table>

Supported tags and respective Dockerfile links:

- 5.2-fpm ([5.2-fpm/Dockerfile](5.2-fpm/Dockerfile))
- 5.3-fpm ([5.3-fpm/Dockerfile](5.3-fpm/Dockerfile))
- 5.4-fpm ([5.4-fpm/Dockerfile](5.4-fpm/Dockerfile))
- 5.5-fpm ([5.5-fpm/Dockerfile](5.5-fpm/Dockerfile))
- 5.4-apache ([5.4-apache/Dockerfile](5.4-apache/Dockerfile))
- 5.5-apache ([5.5-apache/Dockerfile](5.5-apache/Dockerfile))

### Properties

- Created: 2015-05-11 15:59:02
- Updated: jbrinley

### Settings

- Webhooks

### Build Details

- Source Project Page
- Source Repository
PHP

- docker run --rm \
  -v ~/project:/var/www/html \
  --name php \
  --link mysql:mysql \
  --link memcached:memcached \
  jbrinley/php:5.5-fpm
wp-config.php

```php
define( 'DB_USER', 'root' );
define( 'DB_PASSWORD', 'password' );
define( 'DB_HOST', 'mysql' );

$GLOBALS['memcached_servers']
    = array( 'memcached' );
```
```bash
- docker run --rm --name nginx \\
  -p 80:80 \\
  -v path/to/nginx:/etc/nginx:ro \\
  -v ~/project:/var/www/html:ro \\
  --link php:php \\
  nginx
```
nginx/nginx.conf

# ...

fastcgi_pass php:9000;

# ...
Orchestrate your containers

docker-compose
docker-compose.yml

memcached:
  image: memcached:1.4
  ports:
    - "11211"

mysql:
  image: mysql:5.5
  volumes_from:
    - mysqldata
  ports:
    - "3306"
  environment:
    - MYSQL_ROOT_PASSWORD=password

php:
  image: jbrinley/php:5.5-fpm
  volumes:
    - ~/project:/var/www/html
  links:
    - memcached
    - mysql

nginx:
  image: nginx:latest
  volumes:
    - ~/project:/var/www/html:ro
    - path/to/nginx:/etc/nginx:ro
  links:
    - php
  ports:
    - "80:80"
docker-compose up
https://github.com/jbrinley/docker-config
https://github.com/jbrinley/docker-config
Questions?

http://xplus3.net/
@jbrinley
jbrinley@flightless.us
CREDITS

- Presentation template by SlidesCarnival
- Diagrams created in LucidChart
- Docker client/host diagram from docker.com