# Intro to Docker for CMS

# About me

- I am Michael Klatsky

- System Admin and Architect
- Worked for Javanet, RCN, CT Telephone, TNR Global and Esperdyne Technologies
- Avid husband, father, homebrewer and dog lover

# About Esperdyne Technologies, LLC

Esperdyne Technologies

(www.esperdyne.com)

builds performant and scalable full-stack content analysis, parametric and full-text search solutions for publicly facing companies. Our professional services staff works closely with clients to properly identify project scope and mission critical milestones to ensure project success.

Leverage OUR industry veterans as YOUR secret weapon!

# What is Docker?

- Docker is a lightweight container system
- Docker eliminates the overhead associated with virtualization (think VMWare, VirtualBox)
- Docker allows for portability of containers
- Docker is a developer's dream – rapidly test dependencies and versions

ESPERDYNE
TECHNOLOGIES

# How do we use it?

- Company systems such as Redmine, LDAP, run in containers
- Deploy a highly specialized service in a container on an existing EC2 host
- Rapid testing of apps with various dependencies
- Easily scale apps from singles instances to multiple instance with one command

ESPERDYNE
TECHNOLOGIES

# Today's Agenda

Today, we will:

- learn how to install Docker
- learn how to run a container
- learn basic tasks
- learn how to deploy Drupal using Docker
- learn how to upgrade Drupal
- learn how to migrate a container from one host to another

# Install Docker on OSX

- Docker requires a linux kernel
- To use Docker on OSX, obtain the installable package from here:

  https://github.com/boot2docker/osx-installer/releases/download/v1.2.0/Boot2Docker-1.2.0.pkg

- Follow the instructions here:

  http://docs.docker.com/installation/mac/

- This installs 'boot2docker'

# Install Docker on CentOS 7

- Enable the CentOS-Extras repository
- yum install docker (yum install docker-io if using CentOS 6)
- service docker start
- chkconfig docker on

# Install Docker on Windows

- Get the binary from:
  https://github.com/boot2docker/windows-installer/releases/download/v1.2.0/docker-install.exe

- Follow the instructions at:
  http://docs.docker.com/installation/windows/

# Let's run a container

- When running a container, Docker will search locally for an image, and will download one from the Docker registry if not found
- docker run centos
- docker run ubuntu

# Let's run a container and enter it to look around

- docker run --rm -t -i centos /bin/bash
- docker run --rm -t -i ubuntu /bin/bash

The above commands will run a container and place you into a bash shell in the container. When you exit, the container will exit and go away.

# How to deploy a Drupal container

- First, let's grab our images:

- docker pull esperdyne/mysql
- docker pull esperdyne/drupal:7.28
- docker pull esperdune/drupal:7.31

  Now, lets run the mysql container:
- docker run -d  -p 3306:3306 --name DB -e MYSQL_ROOT_PASSWORD=mysecretpassword -e MYSQL_DATABASE=drupal -e MYSQL_USER=drupal esperdyne/mysql

continued...

# How to deploy a Drupal container

Now, let's run our Drupal container and link it to our running mysql container:

- docker run -d  --name drupal --link DB:DB -p 8888:80 esperdyne/drupal:7.28

Browse to http://hostaddress:8888/install.php and run through the install process as usual (insert your hostname/IP address in the URL)

# Let's upgrade Drupal!

- docker rm -f drupal
- docker run -d  --name drupal --link DB:DB -p 8888:80 esperdyne/drupal:7.31

  Browse again to http://hostname:8888/install.php - check the status of your installation to see the new version!

# What about data?

But Houston, we have a problem!
- Data dies with the container
- if we delete the container, we lose the data
- For example, when we rebuild an image to add features,
or to upgrade?

A simple solution- bind mounts....

# Save our data

- docker run -d -v /opt/mysql:/var/lib/mysql -p 3306:3306 --name DB -e MYSQL_ROOT_PASSWORD=mysecretpassword -e MYSQL_DATABASE=drupal -e MYSQL_USER=drupal esperdyne/mysql
- docker run  -d   --name drupal --link DB:DB -p 8888:80 esperdyne/drupal:7.31

The above will persist the mysql data on the host, rather than in the container.

# Common tasks

- docker version
- docker inspect container
- docker stop container
- docker rm container
- docker ps
- docker ps -a
- docker images
- docker rmi image

More info:
https://docs.docker.com/reference/commandline/cli/

# Build containers

- Container built using Dockerfiles

- Contain instructions for building images

- Can be simple, or complex

# Dockerfile

FROM centurylink/apache-php:latest
MAINTAINER CenturyLink
# Install packages
RUN apt-get update && DEBIAN_FRONTEND=noninteractive apt-get -y
upgrade && \
  DEBIAN_FRONTEND=noninteractive apt-get -y install supervisor pwgen && \
  apt-get -y install mysql-client postgresql-client
# Download v7.31 of Drupal into /app
RUN rm -fr /app && mkdir /app && cd /app && \
  curl -O http://ftp.drupal.org/files/projects/drupal-7.31.tar.gz && \
  tar -xzvf drupal-7.31.tar.gz &&  rm drupal-7.31.tar.gz && \
  mv drupal-7.31/* drupal-7.31/.htaccess ./ &&  mv drupal-7.31/.gitignore ./ && \
  rmdir drupal-7.31
#Config and set permissions for setting.php
ADD settings.php app/sites/default/settings.php
RUN mkdir app/sites/default/files && \
    chmod  600 app/sites/default/settings.php && \
    chown www-data app/sites/default/settings.php && \
    chmod 755 app/sites/default &&  chmod 777 app/sites/default/files
EXPOSE 80
CMD exec supervisord -n

# Dockerfiles, deux

While docker files can be "busy", they offer a repeatable way to build an image.

The previous Dockerfile will build a drupal image. Simply issue the following command in the directory containing the Dockerfile:

```
docker build -t mydrupal
```

# Dockerfile(simpler)

FROM ubuntu:12.04

RUN apt-get update
RUN apt-get install -y apache2

ENV APACHE_RUN_USER www-data
ENV APACHE_RUN_GROUP www-data
ENV APACHE_LOG_DIR /var/log/apache2

EXPOSE 80

ENTRYPOINT ["/usr/sbin/apache2"]
CMD ["-D", "FOREGROUND"]

- This builds a container which just runs a webserver.

# Upgrade the build

## Feel like building an upgraded image?

```
RUN rm -fr /app && mkdir /app && cd /app && \
curl -O http://ftp.drupal.org/files/projects/drupal-7.28.tar.gz && \
tar -xzvf drupal-7.28.tar.gz &&  rm drupal-7.28.tar.gz && \
mv drupal-7.28/* drupal-7.28/.htaccess ./ &&  mv drupal-7.28/.gitignore ./ && \
rmdir drupal-7.28
```

In the above lines from our Dockerfile, simply change the version (7.28 -> 7.31), then run:

docker build -t mydrupal

Restart your container and voila!

# Copy to a new host

How do we copy a container to a new host?

- docker export db_new > db_new.tar

  Copy the tarfile to a new host (scp, ftp, etc)
- cat db_new.tar | docker import – db_new
  /usr/local/bin/run

  Now, just run the db_new image:
- docker run –name my_db_new -d db_new

# Additional tools

Fig:
http://www.fig.sh

Kitematic (OSX)
https://kitematic.com

# Credits and more reading

Century Link:
http://www.centurylinklabs.com/category/docker-posts/

Kartar (The Docker Book):
http://kartar.net

Docker:
http://www.docker.io

Thomas Uhrig:
http://tuhrig.de/tag/docker/

Digital Ocean:
https://www.digitalocean.com/community/tags/docker?primary_filter=tutorials

# Contact info

For more information:

Michael Klatsky
**Esperdyne Technologies, LLC**
Phone: 413-200-TECH (8324)

http://www.esperdyne.com
mklatsky@esperdyne.com