

SHIPPING CONTAINERS WITH DOCKER

SCOTT P. GALLAGHER
SCOTTPGALLAGHER@PSU.EDU
[@SCOTTPGALLAGHER](#)



Me!

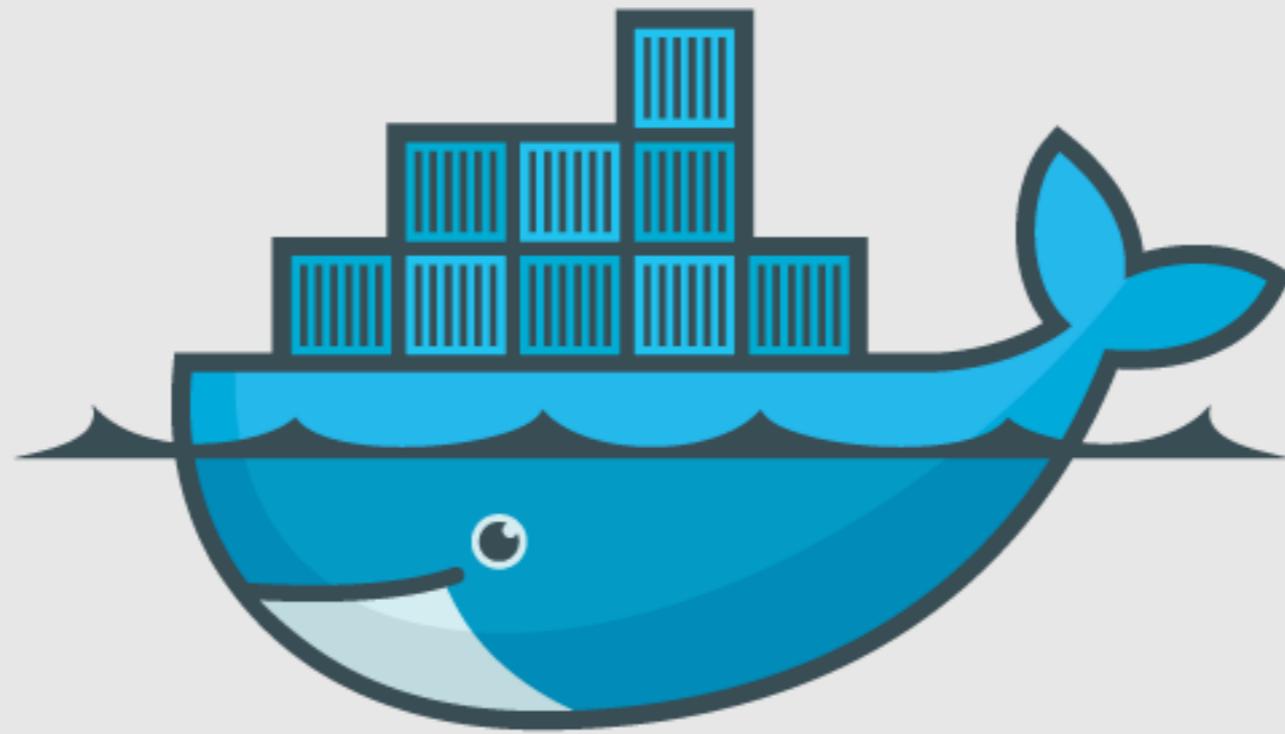
- * Systems Administrator @ Penn State (*nix & Mac)
- * @scottpgallagher
- * <https://hub.docker.com/u/scottpgallagher/>

Docker!???. . .



Docker's vs Levi's

- * Inline seems
- * Pleated legs
- * Boot cut



docker

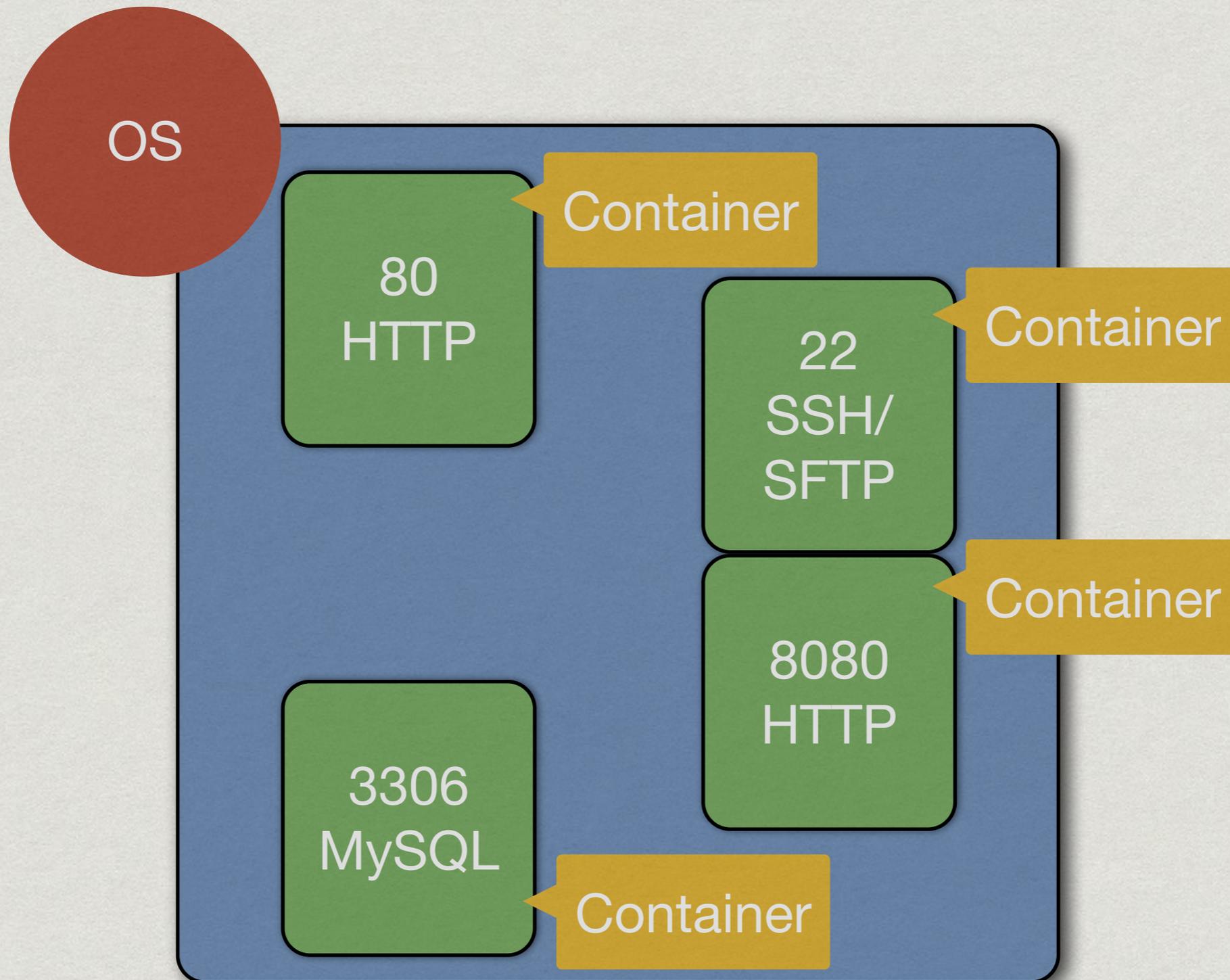
Using Docker?

- * Who's heard of Docker?
- * Anybody go to DockerCon14?
 - * Encourage to watch videos on youtube
- * Anybody running it?
 - * Production?

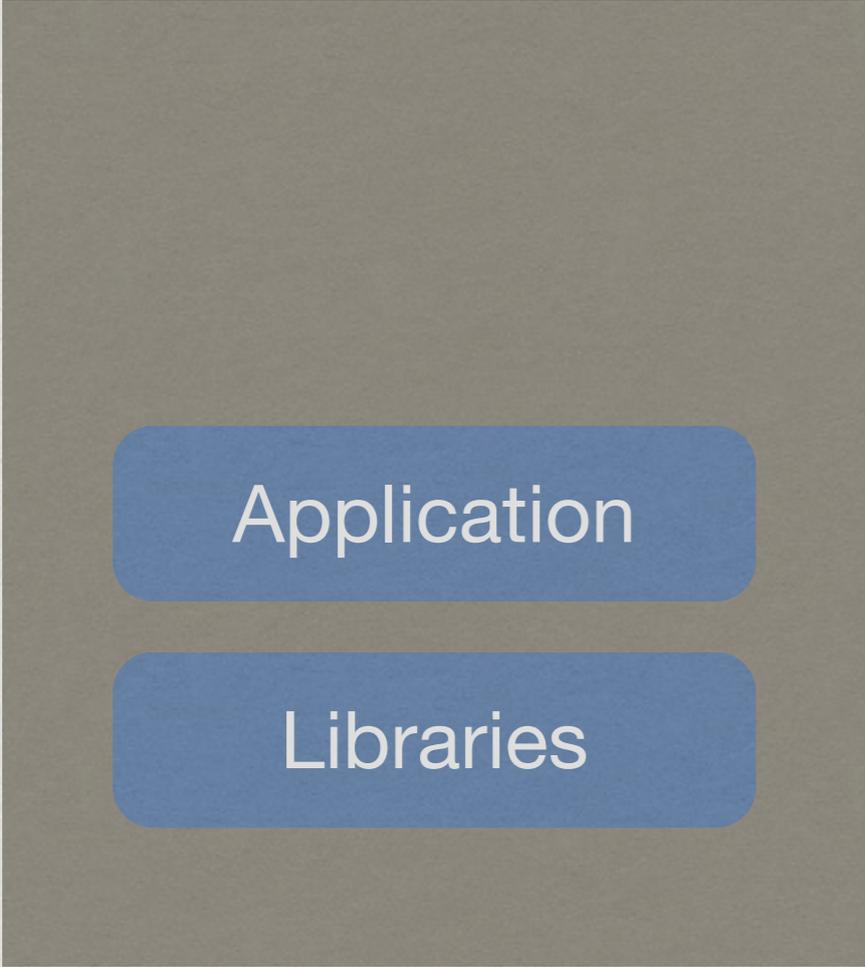
Dock...errrr?

- * So What is it?! History lesson!
- * Started in 2010 as dotCloud; became so great they changed the company name to Docker
- * Lightweight containers with a contained application, configuration files, and dependencies
 - * Takeaway: Lightweight and isolated!

What is a container?



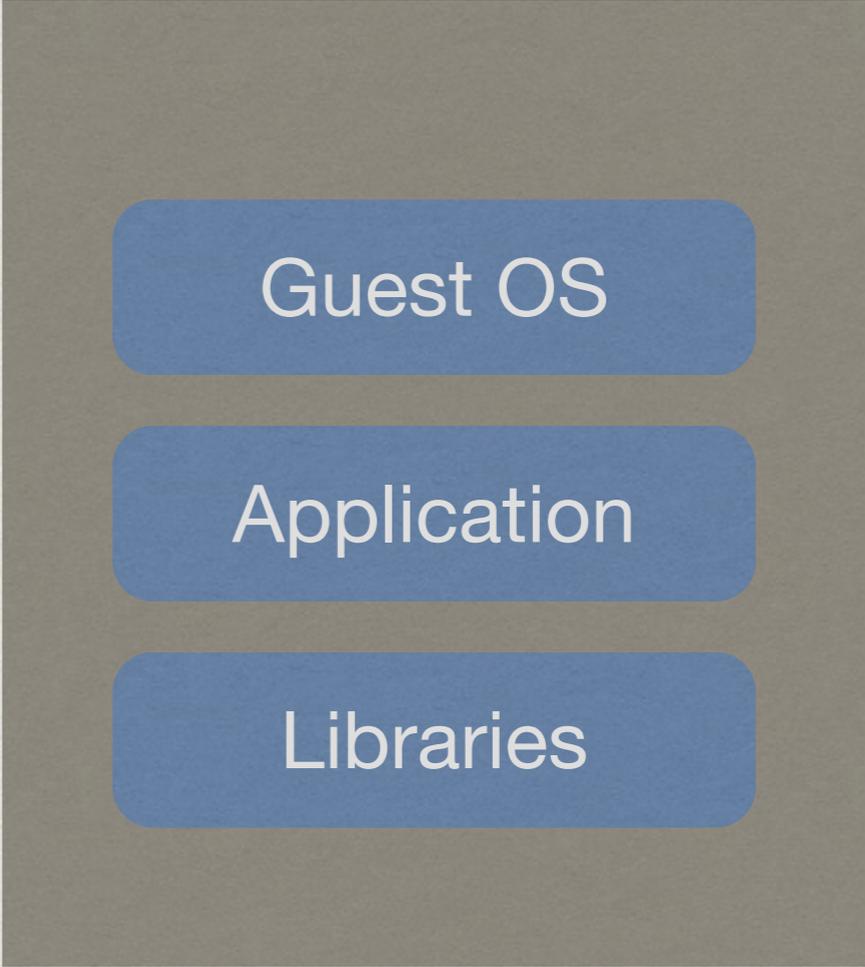
Containers vs VM



Application

The diagram shows a dark gray rectangular container. Inside, there are two blue rounded rectangular boxes stacked vertically. The top box is labeled 'Application' and the bottom box is labeled 'Libraries'.

Libraries



Guest OS

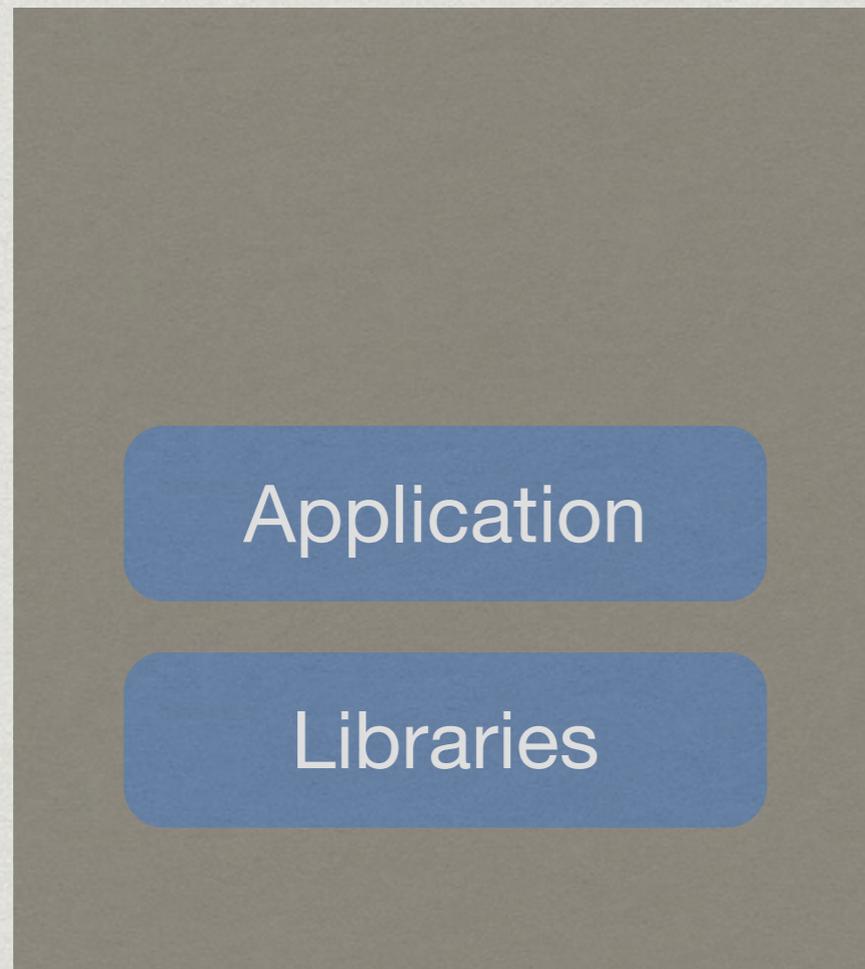
The diagram shows a dark gray rectangular container. Inside, there are three blue rounded rectangular boxes stacked vertically. The top box is labeled 'Guest OS', the middle box is labeled 'Application', and the bottom box is labeled 'Libraries'.

Application

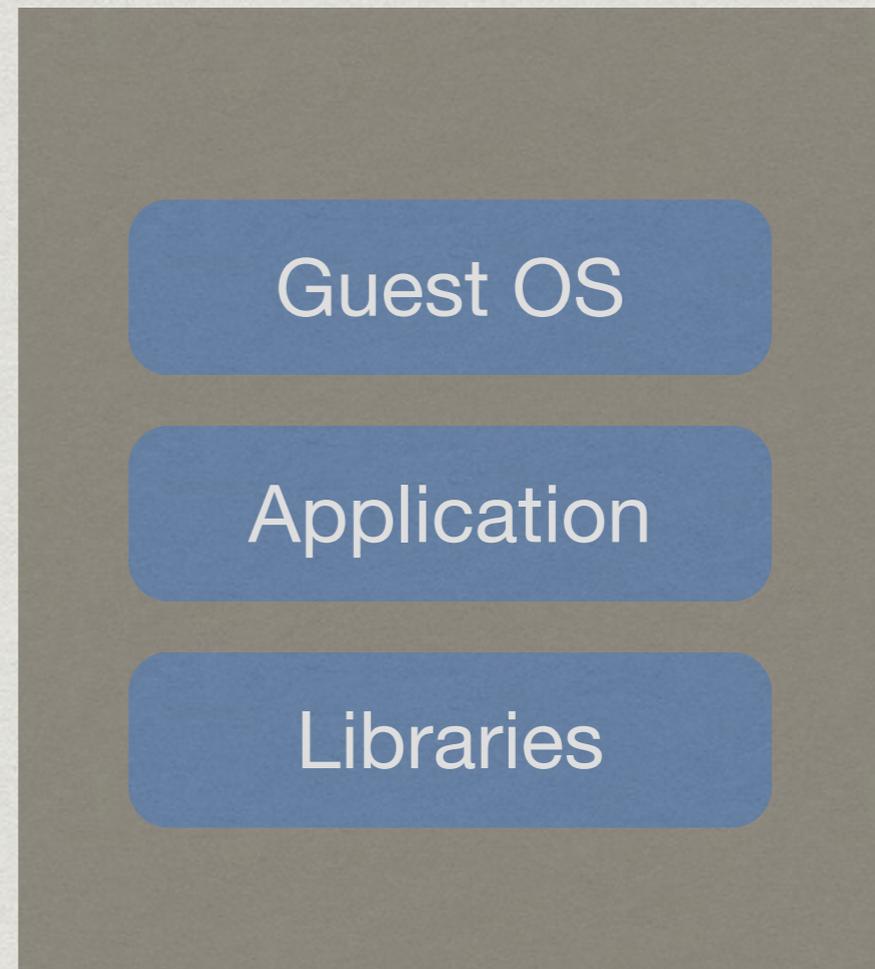
Libraries

Containers vs VM

Container

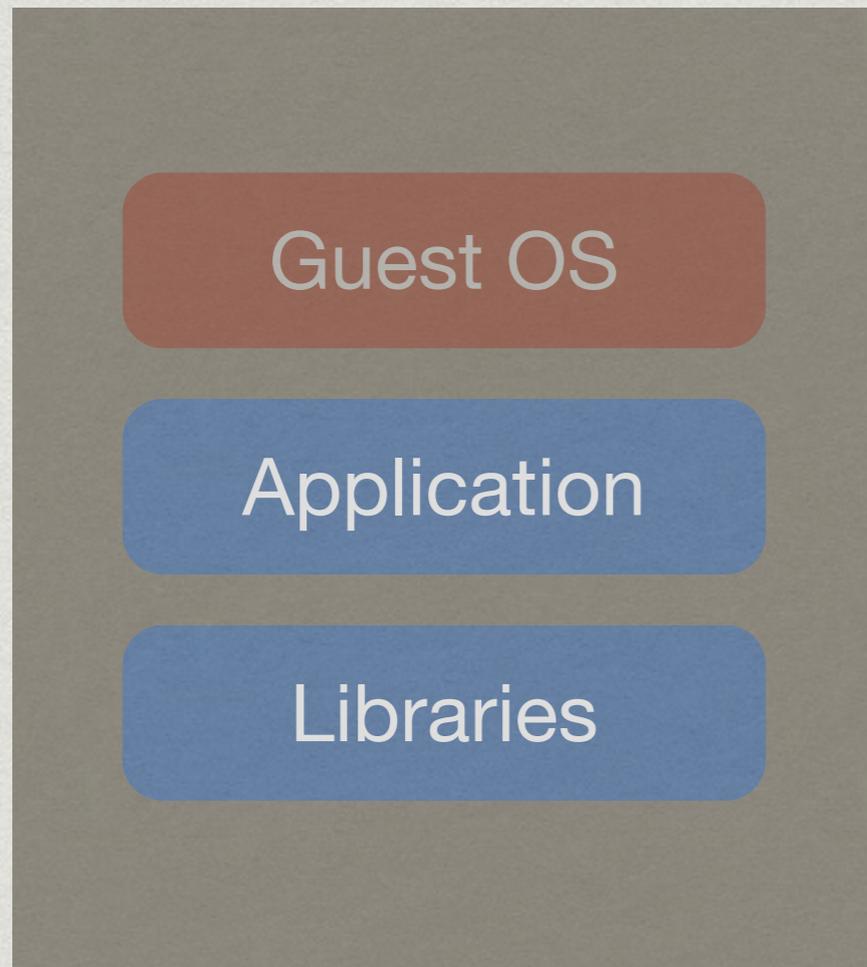


VM

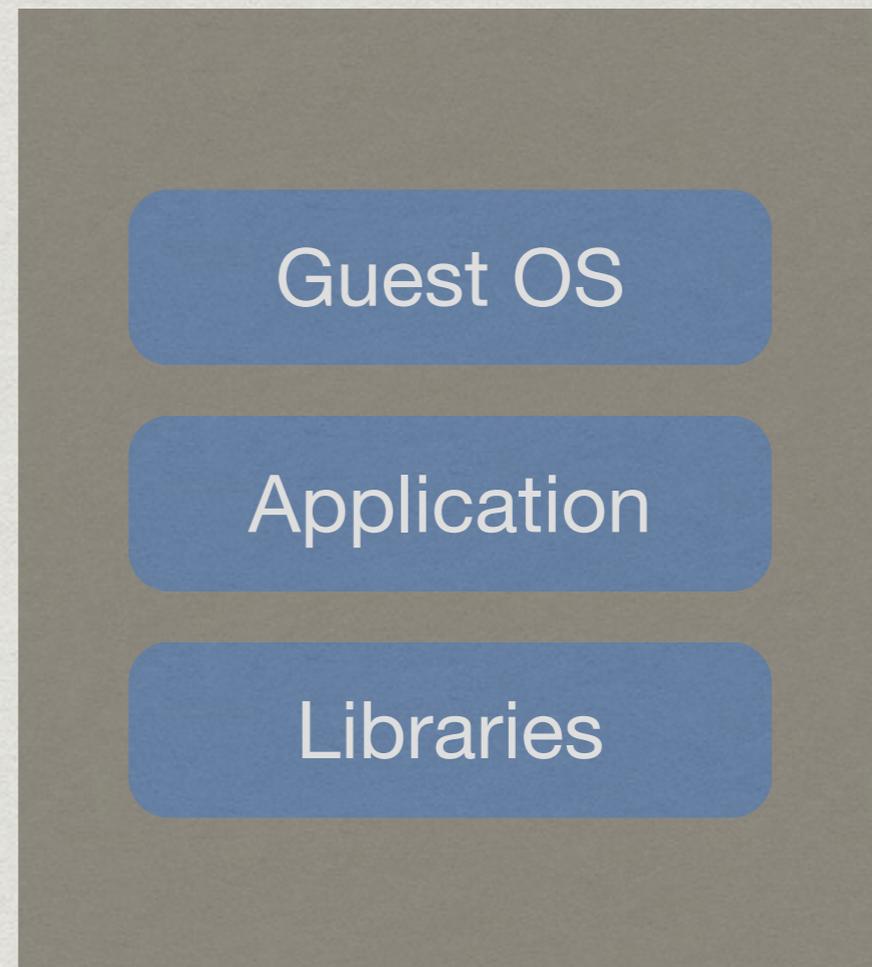


Containers vs VM

Container



VM



Why Docker?

- * People make mistakes!
- * Docker containers are an exact replica of each other
- * Automation to the next level! No need to SSH again! (*by using docker commands*)
- * IP addresses are not tied to a container
- * Open source!

How I'm Using Docker

- * Internal servers to our group (git, MDM, ASR, etc)
- * Sites.psu.edu
 - * CMS - Wordpress
 - * Web, Varnish, Memcached, MySQL, NFS, etc
 - * Staging and DR environments

How I'm Using Docker

- * Continuous Integration
 - * Code forked on GitHub (*by developers*)
 - * Get a pull request (*from developers*)
 - * Do a commit of new code
 - * Web hook make a call to Jenkins

How I'm using Docker

- * Jenkins builds the containers (*using jenkins docker plugin*)
- * Jenkins then sends the updated container to the Docker Hub
 - * Docker Hub then..
 - * Tests the code
 - * Pushes the containers to all locations

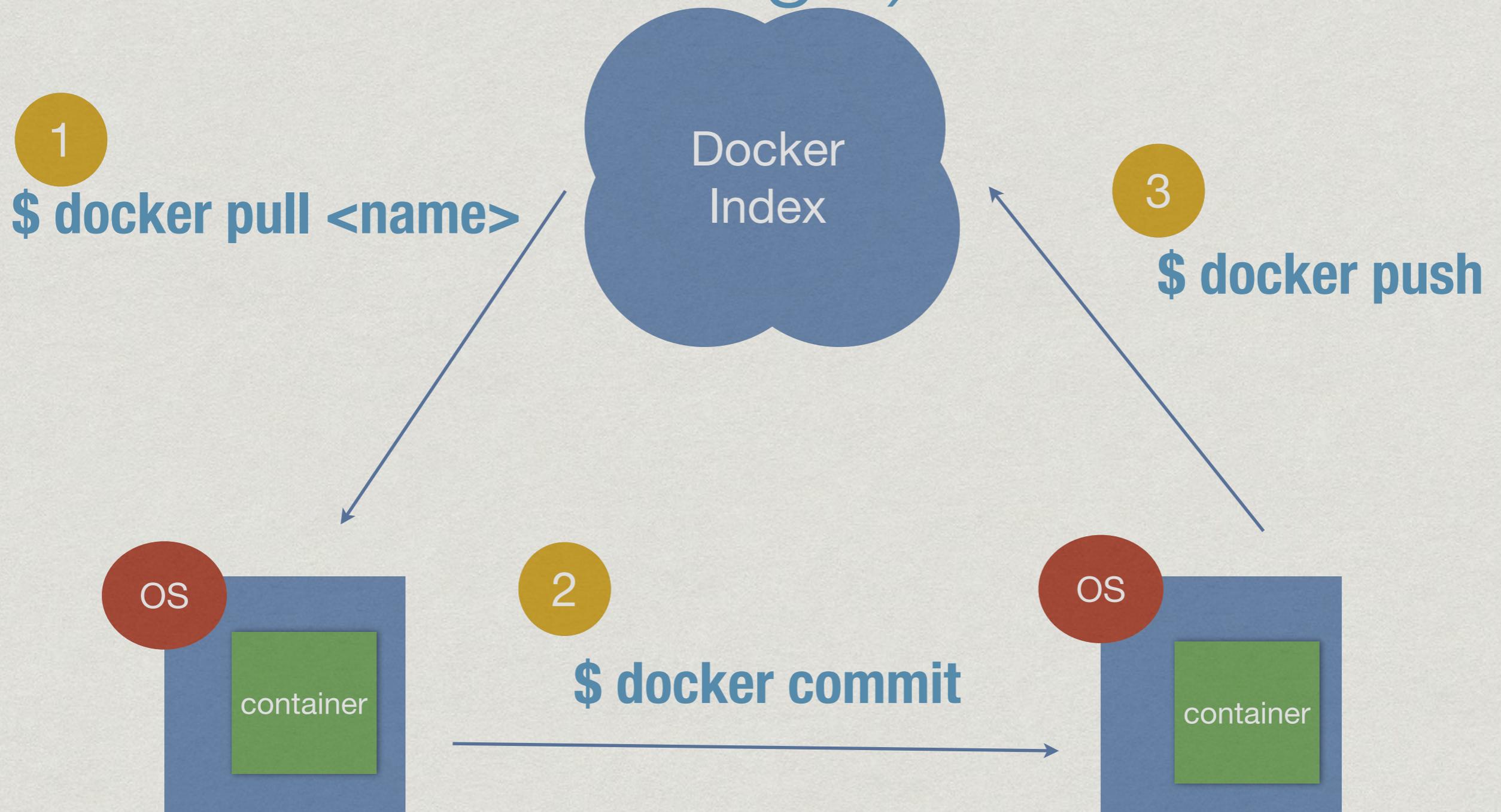
Benefits

- * Built on unison file system (layers)
 - * When making changes only items that are changed are shipped
- * Upgrades
 - * Upgrade base VM could update all the containers!
 - * So upgrading 1 VM running 20 containers as opposed to updating 20 VMs
- * Cross platform
 - * Just need docker on Windows, Mac, or Linux

Challenges Solved

- * Easy rollbacks
 - * Screw something up? Roll it back!
- * Immutable containers
 - * Changes aren't saved until you do the docker commit
- * Makes shipping/sharing items easier
 - * Think of it like shipping code around through Git

How it works (*not unicorn magic*)



Use cases

- * Software environment for student lab use
 - * Software configuration could be become difficult for students to configure on their own
- * Sharing our build/configurations with others
 - * How are you currently doing this?
 - * Shipping a full VM around?
- * Testing!!

Docker commands...

```
Usage: docker [OPTIONS] COMMAND [arg...]  
-H=[unix:///var/run/docker.sock]: tcp://host:port to bind/connect to or unix://path/to/socket to use
```

A self-sufficient runtime for linux containers.

Commands:

attach	Attach to a running container
build	Build an image from a Dockerfile
commit	Create a new image from a container's changes
cp	Copy files/folders from the containers filesystem to the host path
diff	Inspect changes on a container's filesystem
events	Get real time events from the server
export	Stream the contents of a container as a tar archive
history	Show the history of an image
images	List images
import	Create a new filesystem image from the contents of a tarball
info	Display system-wide information
inspect	Return low-level information on a container
kill	Kill a running container
load	Load an image from a tar archive
login	Register or Login to the docker registry server
logs	Fetch the logs of a container
port	Lookup the public-facing port which is NAT-ed to PRIVATE_PORT
pause	Pause all processes within a container
ps	List containers
pull	Pull an image or a repository from the docker registry server
push	Push an image or a repository to the docker registry server
restart	Restart a running container
rm	Remove one or more containers
rmi	Remove one or more images
run	Run a command in a new container
save	Save an image to a tar archive
search	Search for an image in the docker index
start	Start a stopped container
stop	Stop a running container
tag	Tag an image into a repository
top	Lookup the running processes of a container
unpause	Unpause a paused container
version	Show the docker version information
wait	Block until a container stops, then print its exit code

Useful Docker commands...

- * `$ docker inspect CONTAINER|IMAGE`
- * Can view configurations about a container w/o having to actually attach to container
- * Memory, Hostname, IP address, Creation date, Docker version created with, container size, etc

Useful Docker commands...

```
cb104-107:~ scottpgallagher$ docker inspect mysql:latest
[{"Architecture": "amd64",
  "Author": "",
  "Comment": "",
  "Config": {
    "AttachStderr": false,
    "AttachStdin": false,
    "AttachStdout": false,
    "Cmd": [
      "mysqld",
      "--datadir=/var/lib/mysql",
      "--user=mysql"
    ],
    "CpuShares": 0,
    "Cpuset": "",
    "Domainname": "",
    "Entrypoint": [
      "/entrypoint.sh"
    ],
    "Env": [
      "HOME=/",
      "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/local/mysql/bin:/usr/local/mysql/scripts"
    ],
    "ExposedPorts": {
      "3306/tcp": {}
    },
    "Hostname": "3c2933af3ca7",
    "Image": "11a65755206163698ef1bd66e862893ce5c4dc92d3ca589726685dde77b6bb04",
    "Memory": 0,
    "MemorySwap": 0,
    "NetworkDisabled": false,
    "OnBuild": [],
    "OpenStdin": false,
    "PortSpecs": null,
    "StdinOnce": false,
    "Tty": false,
    "User": "",
    "Volumes": {
      "/var/lib/mysql": {}
    },
    "WorkingDir": "/usr/local/mysql"
  }
}]
```

Useful Docker commands...

- * `$ docker events`
- * Shows real time events that occur on your containers
- * Options: `--since` & `--until`
 - * `$ docker events --since="09:49:03"`
 - * `$ docker events --until="12:32:40"`

Useful Docker commands...

```
cb104-107:~ scottpgallagher$ docker events
```

```
[2014-06-12 21:12:53 -0400 EDT] 9692d73eb6bbc8de5a16e2c990bfcd9f2d03300c3098eaefea46cf3ceeb9f15e: (from r  
edhat/rhel7:0) create  
[2014-06-12 21:12:53 -0400 EDT] 9692d73eb6bbc8de5a16e2c990bfcd9f2d03300c3098eaefea46cf3ceeb9f15e: (from r  
edhat/rhel7:0) start  
[2014-06-12 21:14:01 -0400 EDT] 9692d73eb6bbc8de5a16e2c990bfcd9f2d03300c3098eaefea46cf3ceeb9f15e: (from r  
edhat/rhel7:0) die
```

Useful Docker commands...

- * `$ docker history CONTAINER|IMAGE`
- * Shows the history of a container; items that have changed

Useful Docker commands...

```
cb104-107:~ scottpgallagher$ docker history ubuntu:latest
```

IMAGE	CREATED	CREATED BY	SIZE
ad892dd21d60	8 days ago	/bin/sh -c apt-get update && apt-get install	82.87 MB
9db365ecbcbb	8 days ago	/bin/sh -c sed -i 's/^#\s*\s*(deb.*universe\)\$/'	1.903 kB
23f361102fae	8 days ago	/bin/sh -c echo '#!/bin/sh' > /usr/sbin/polic	194.5 kB
e465fff03bce	8 days ago	/bin/sh -c #(nop) ADD file:481ca6c4ad290cd6ec	192.5 MB
511136ea3c5a	12 months ago		0 B

Useful Docker commands...

- * `$ docker logs CONTAINER ID`
- * Allows you to view container logs
- * `--follow` option allows you to stream logs
- * It's like ease dropping on the container!

Useful Docker commands...

- * `$ docker search TERM`
- * Allows you to search Docker hub for pre-packaged containers
- * Very helpful so you don't have to switch between terminal & browser
 - * Increased productivity!

Useful Docker commands...

- * \$ docker top
 - * View running processes on a container

Useful Docker commands...

```
cb104-107:~ scottpgallagher$ docker top dcae6c6ecdf9
PID          USER        COMMAND
874          root        /bin/bash
892          root        top
```

More Benefits!

- * Docker allows you to almost never have to connect to the container to get information from it
- * Pause and unpause containers
- * `docker diff`
 - * See what has changed, been added, or deleted

Deploying Docker

- * Docker file (\$ docker build)

```
# Dockerizing MongoDB: Dockerfile for building MongoDB images
# Based on ubuntu:latest, installs MongoDB following the instructions from:
# http://docs.mongodb.org/manual/tutorial/install-mongodb-on-ubuntu/

FROM ubuntu:latest
MAINTAINER Docker

# Installation:
# Import MongoDB public GPG key AND create a MongoDB list file
RUN apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 7F0CEB10
RUN echo 'deb http://downloads-distro.mongodb.org/repo/ubuntu-upstart dist 10gen' | tee /etc/apt/sources.list.d/10gen.list

# Update apt-get sources AND install MongoDB
RUN apt-get update
RUN apt-get install -y -q mongodb-org

# Create the MongoDB data directory
RUN mkdir -p /data/db

# Expose port #27017 from the container to the host
EXPOSE 27017

# Set usr/bin/mongod as the dockerized entry-point application
ENTRYPOINT usr/bin/mongod
```

- * hub.docker.com

- * \$ docker pull ubuntu

Managing Containers

- * Shipyard
- * Cockpit
- * Geard
 - * Redhat
- * Helios
 - * Used/Managed by Spotify
- * Kubernetes
 - * Redhat and Google collaboration
- * Many many others!

Using Docker

- * Installation
- * Images
 - * Searching (command line or browser)
 - * Pulling
- * Committing changes
- * Pushing Image (why not share it!?)
- * Docker hub

Using Docker

- * Configuring Virtualbox for docker (*since it runs in a VM and not directly on Mac*)
- * Port forwarding configuration
- * `-p 8000:8000` option in docker

Linking containers

- * So by default containers are self contained...but you can link them.....so they are aware of each other
- * Lets you securely link containers together
- * For example: Linking a web server to a database server

DEMO TIME!

Docker 1.1.0 Features

- * Released July 3rd
- * `.dockerignore`
- * Pauses during commits (*inconsistent file states if container was processing during commit*)
- * `docker log --tail`
- * Pass `.tar` files during docker build

Resources

- * docker.com
- * <http://www.docker.com/tryit/>
- * hub.docker.com
- * <http://blog.docker.com>
- * youtube.com/user/dockerrun/

Resources

- * <https://github.com/dotcloud/docker>
- * @docker on twitter
- * Freenode IRC channels
 - * #docker
 - * #docker-dev
- * <http://crosbymichael.com>

Q & A

- * spg14@psu.edu
- * @scottpgallagher
- * <https://hub.docker.com/u/scottpgallagher/>
- * <http://j.mp/psumac67>