

Docker Overview

The background of the slide features the Docker logo, which is a stylized blue whale. On its back is a cargo ship with several blue shipping containers stacked on top. A thin orange horizontal line is positioned below the whale's back.

What's up with this thing called Docker?

How would you
describe your
software
development
process?





Waterloop

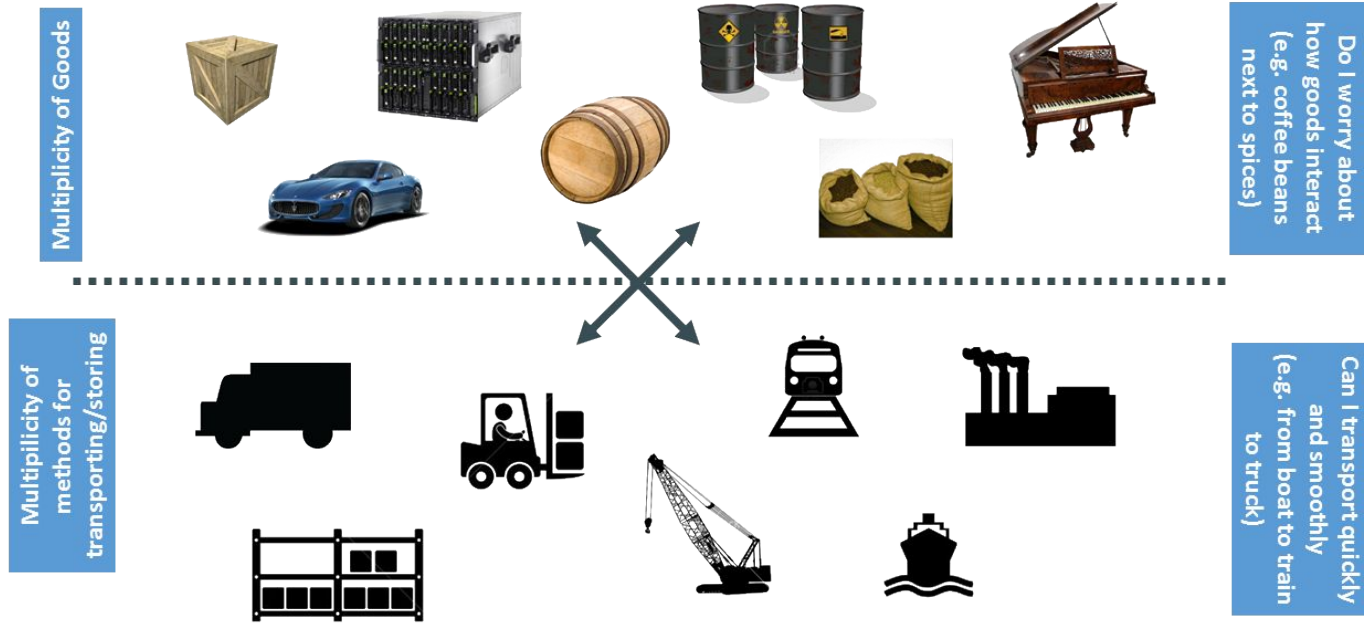
Waterloop



g Bridge Yrs

What makes dev/deploying difficult?

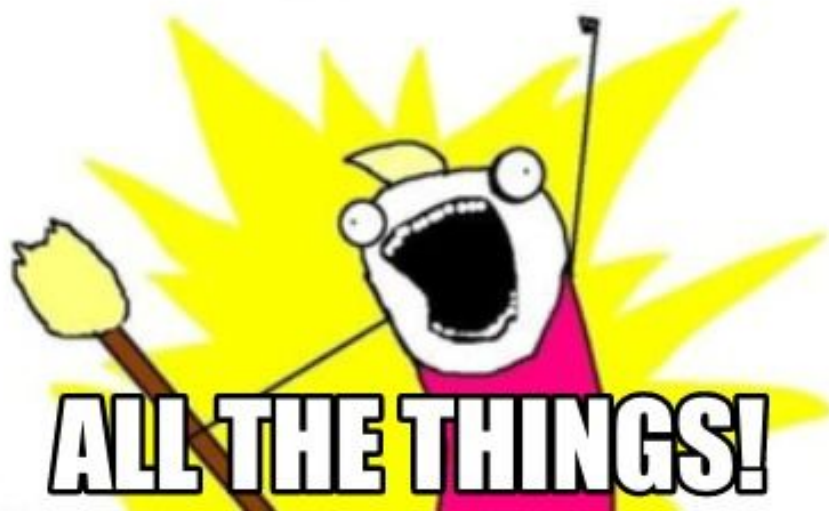
Not the first time we've had this problem...



And the solution was...



CONTAINERIZE



What's Docker providing?



“Docker provides an integrated technology suite that enables development and IT operations teams to build, ship, and run distributed applications anywhere.”

- **Build** - package an application with its dependencies and environment
- **Ship** - share the package with all deployment environments
- **Run** - run, scale, and monitor your application

Developers

IT Operations

BUILD

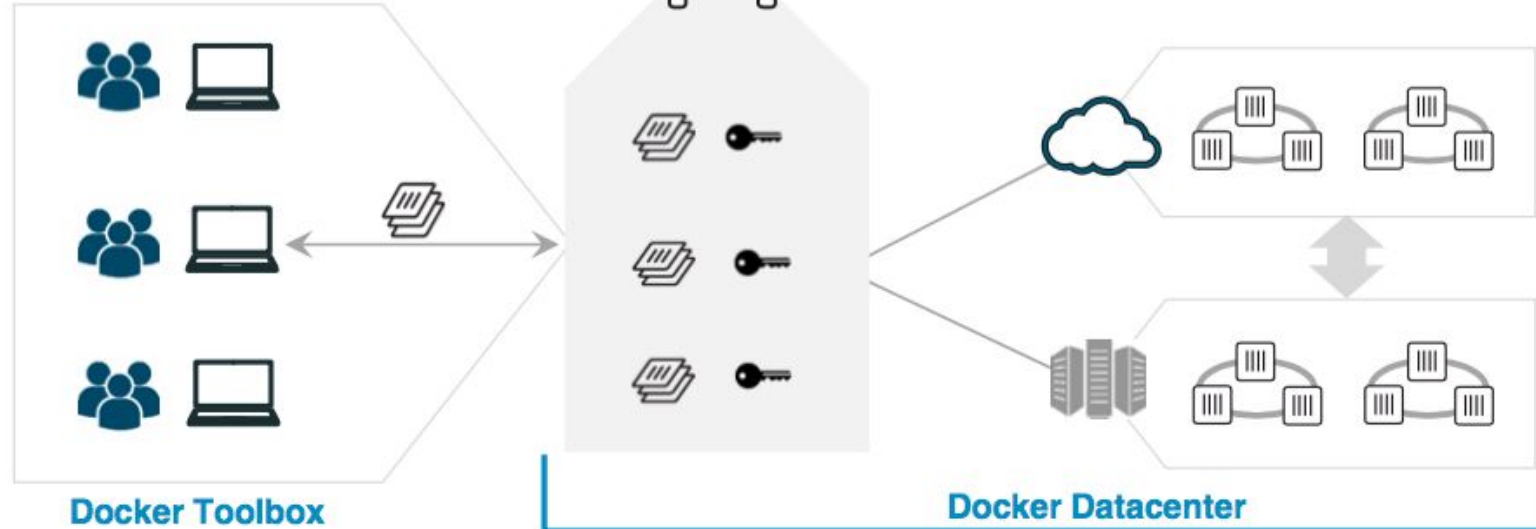
Development Environments

SHIP

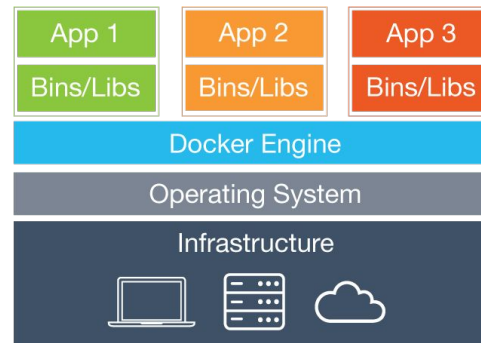
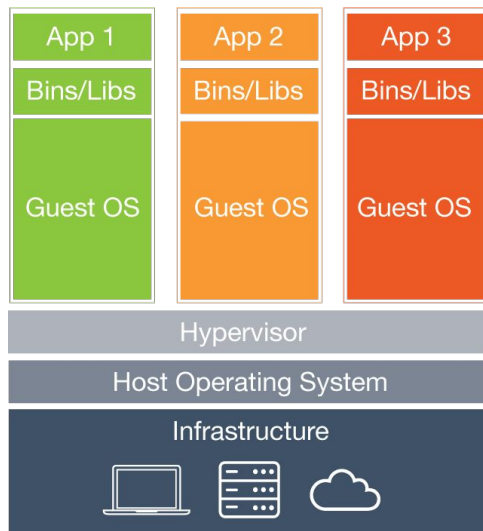
Secure Content & Collaboration

RUN

Deploy, Manage, Scale

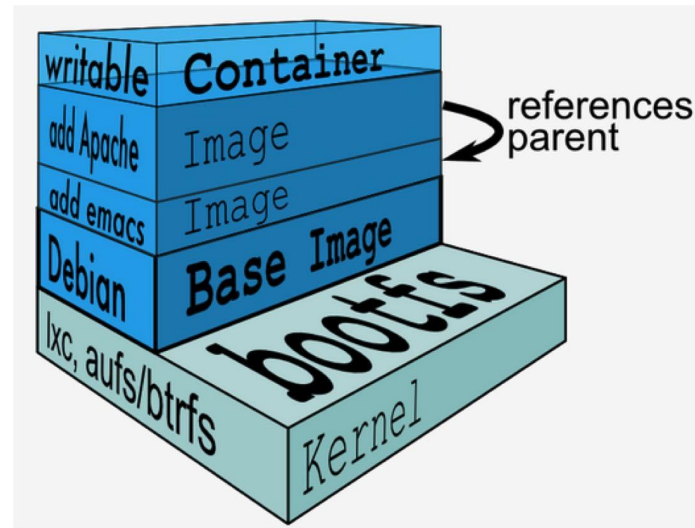


What makes Docker different?



Docker Images and Containers

- Images consist of layers of filesystem changes
- **Images** are *immutable* and *stateless*
 - Only filesystem data is persisted, not running processes
- **Containers** are running *instances* of an image
- Each container is given its own writeable layer
 - Allows multiple concurrent running containers from a single image
- Deleting a container is simply removing its writeable layer. And with that... it's gone!



How do we...

- share/retrieve an image?
- run a container?
- determine what containers are running?
- kill a running container?
- mount a volume and do some work?

Hands-on time!

How are images shared?

- Much like code, images are shared in registries
 - Default registry - Docker Hub (hub.docker.com)
 - Can run on-prem using Docker Registry
 - AWS has Elastic Container Registry (ECR); Azure has Azure Container Service (ACS)
 - GitLab just announced their Container Registry; Nexus has one too
- Images can be pulled using `docker pull <image name>`

```
mikesir@~$ docker pull ubuntu:16.04
16.04: Pulling from library/ubuntu

6d28225f8d96: Pull complete
166102ec41af: Pull complete
d09bfba2bd6a: Pull complete
c80dad39a6c0: Pull complete
a3ed95caeb02: Pull complete
Digest: sha256:eb96613af7b0cd3839dcfb1e4184d94222d2f2127ed5bb4fcb047805ddcf56b6
Status: Downloaded newer image for ubuntu:16.04
mikesir@~$
```

Running Containers

```
docker run -ti ubuntu:16.04
```

- **run** - command that starts a container with the specified image
- **-t** - starts a psuedo-TTY
- **-i** - keeps STDIN open
- **ubuntu:16.04** - run the 16.04 tag of the ubuntu image
 - If the image hasn't been pulled, an attempt will be made
- Runs image's default command, in this case `/bin/bash`. Kill it, container stops.

```
mikesir@nginx$ docker run -ti ubuntu:16.04
root@25221fe81d70:/# ps -aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1   1.5  0.1  18240  3348 ?        Ss   21:13   0:00 /bin/bash
root       11   0.0  0.1  34424  2820 ?        R+   21:13   0:00 ps -aux
root@25221fe81d70:/# exit
exit
mikesir@nginx$
```

Running Containers

- Changes made in a container do NOT update the image
 - Example: Start Ubuntu, install curl, exit. Start a new container, curl not installed

```
mikesir@nginx$ docker run -ti ubuntu:16.04
root@c3405f79709f:/# curl http://www.vt.edu
bash: curl: command not found
root@c3405f79709f:/# apt-get update; apt-get install curl -y
Get:1 http://archive.ubuntu.com/ubuntu xenial InRelease [247 kB]
Get:2 http://archive.ubuntu.com/ubuntu xenial-updates InRelease [94.5 kB]
done.
root@c3405f79709f:/# curl http://www.vt.edu/ -o vt.txt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100  7448  100  7448    0     0  19884      0 --:--:-- --:--:-- --:--:-- 19914
root@c3405f79709f:/# exit
exit
mikesir@nginx$ docker run -ti ubuntu:16.04
root@85ec1d949b92:/# curl http://www.vt.edu/
bash: curl: command not found
```

A “real-life” example

- Want to run and make changes to a static site using Nginx

```
docker run -d -p 80:80 -v $(pwd):/usr/share/nginx/html nginx
```

Run in detached mode
(in the background)

Expose the container's
port 80 to the host's
port 80

Mount the current directory as
a volume to the host's
/usr/share/nginx/html directory

Run the nginx image

What containers are running?

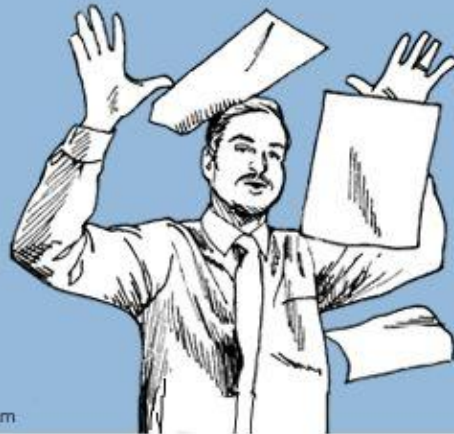
- `docker ps` - view all running containers
- `docker ps -a` - view all containers, including those that are exited
- Container's unique ID - used for other commands
- Container's name - randomly selected if one is not provided when running.
Used for linking (we'll get to that later)
- Ports - shows port mappings and what ports the image wanted to expose (in this case, it's exposing 443, but we're not mapping it)

```
mikesir@nginx$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
4a487e7dd3c0	nginx	"nginx -g 'daemon off'"	2 minutes ago	Up 2 minutes	443/tcp, 0.0.0.0:10080->80/tcp	elated_shockley

```
mikesir@nginx$
```


**drops the Mic and Walks off
stage**



your  cards
someecards.com

Any Questions?
