Docker JumpStart

put it in and ship it

Agenda

- What to expect here
- What is Docker
- Get your hands wet
  - creating / start / stop
  - Container, Volumes, Images
  - Dockerfile
  - Networking
  - Compose

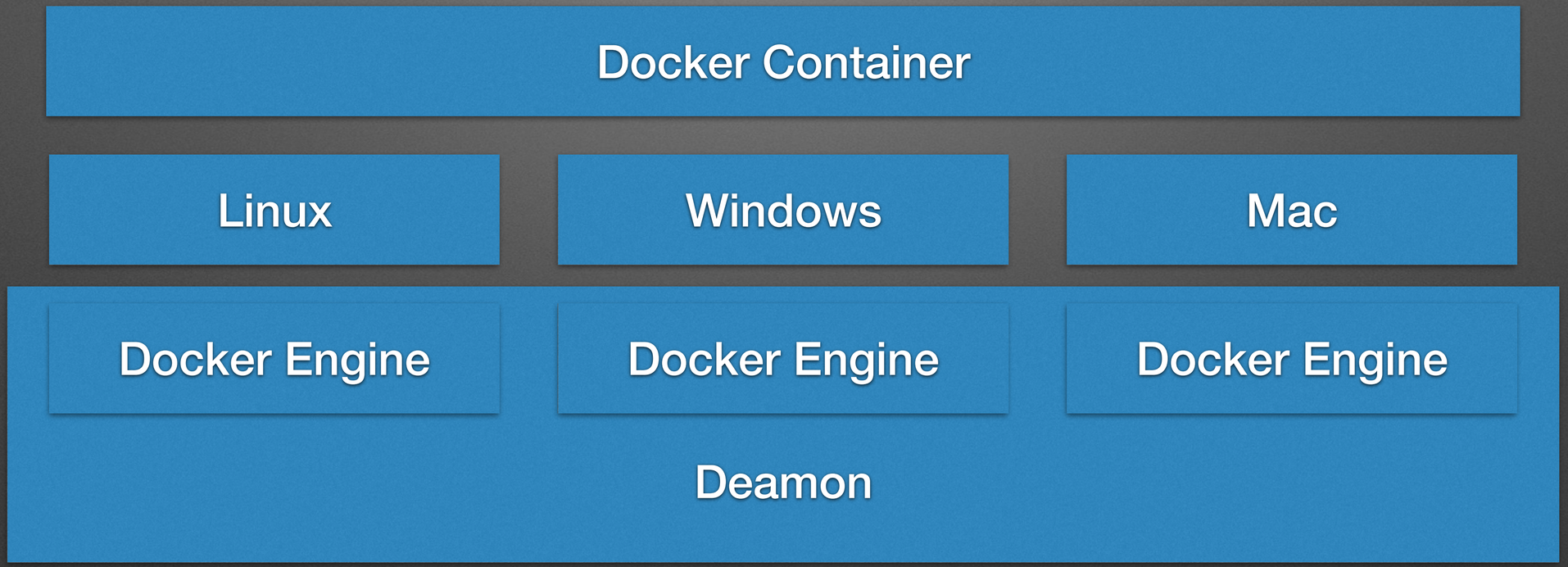
What to expect here

- get an idea of what Docker is
- first experience with Docker
- managing containers through the CLI

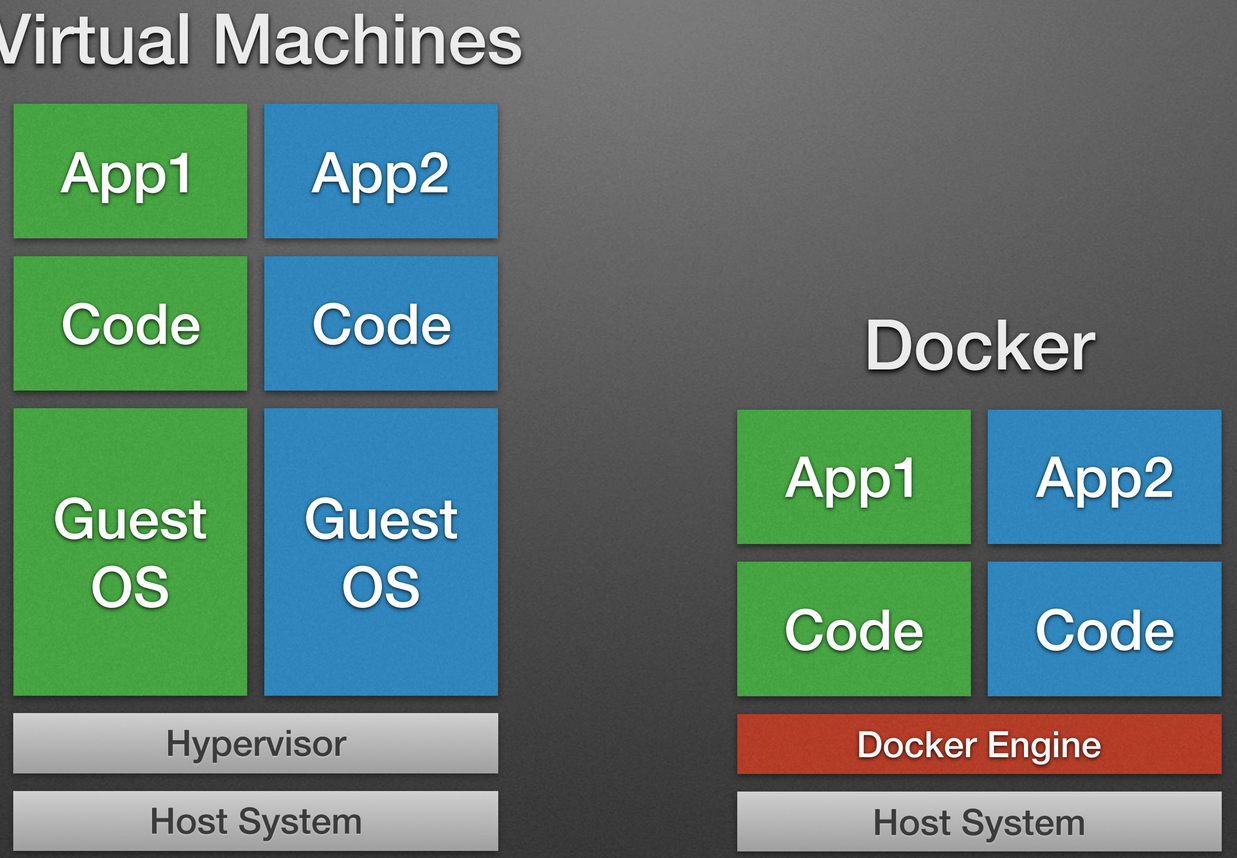
What is Docker?

- standard object
- can contain Linux based software
- can be moved around
- can be combined with other containers
- self-contained object
- own duty

Where is a container running?



Virtual Machine VS Docker

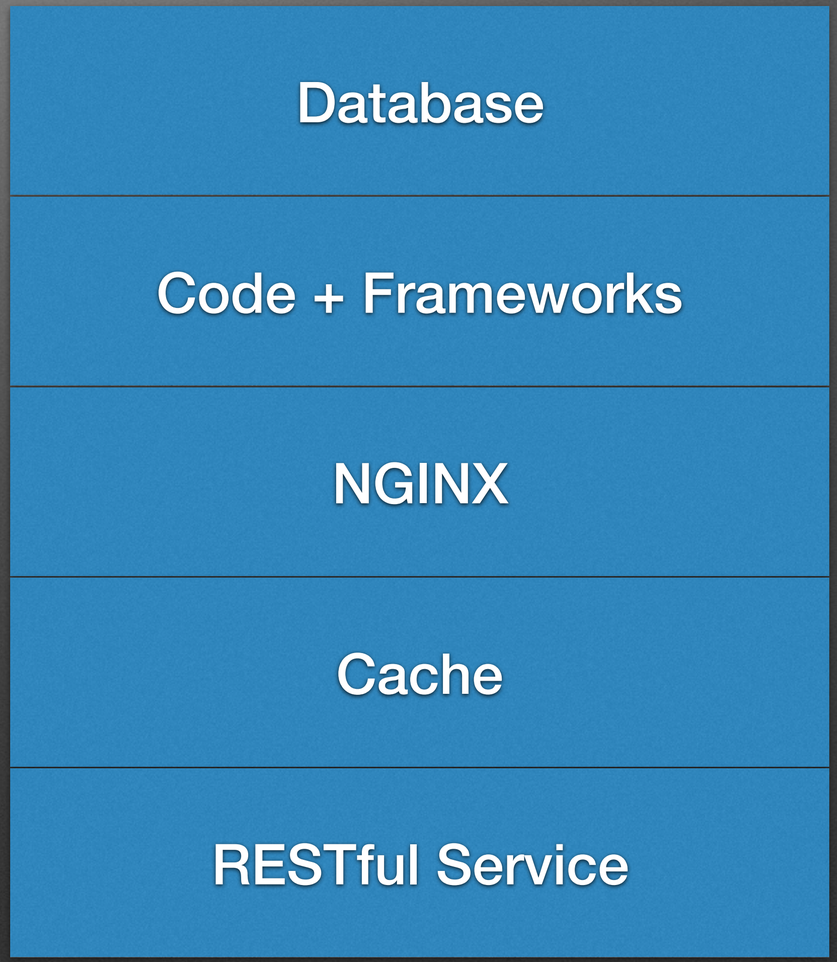


boring tech info

- Docker is a virtualization on Linux
- Isolation
  - cgroups
  - namespaces
  - AuFS Filesystem

Benefits of Docker

- package only what you need, less overhead
- combine what you need
- swap when needed
- uniform CI Deploy, Dev, Test, …, Deploy
- Docker eco system



Resources

- hub.docker.com
- docker.com
- docs.docker.com

Hand On

Installation

- Windows 10 Pro, Edu, Enterprise or Mac OS X Intel Yosemite
  Beta (Win / Mac) (new) >v1.12
  https://docs.docker.com/engine/installation/

- with Toolbox (Win x64 / Mac) (old) v1.11
  https://docs.docker.com/v1.11/

...

```
$ docker load --input hello-world.tar
$ docker load --input debian.tar
$ docker load --input dotnet.tar
```

Images

$ docker pull hello-world
$ docker run hello-world
$ docker run --help
$ docker images
$ docker ps
$ docker ps -a
HINT: to reference a container only the first individual characters are needed
$ docker rm XXX
$ docker rmi XXX

running a container

```
$ docker run -it debian bin/bash
# apt-get update
Ctrl+p, Ctrl+q
$ docker ps
$ docker attach XXX
Ctrl+p, Ctrl+q
$ docker ps
$ docker stop XXX
$ docker ps
$ docker ps -a
$ docker start XXX
$ docker ps
$ docker rm XXX
```

Images, Containers, Volumes

Images are like blueprints
- collection of changes
- read-only

volumes are files from the local file system

a container is an instance of an image

Container and Volumes

$ docker run -it -v /tmp/test debian bin/bash
Ctrl+p, Ctrl+q
$ docker ps
$ docker inspect XXX

$ docker stop XXX
$ docker rm XXX

Container and Volumes

$ mkdir jumpstart
$ cd jumpstart

$ docker run -it -v $(pwd):/tmp/test/ -w „/tmp/test“ debian bin/bash
# touch LeaveMeHere.txt
Ctrl+p, Ctrl+q
$ ls
$ docker ps
$ docker stop XXX
$ docker rm XXX
$ ls

Container and Volumes

```
$ mkdir jumpstart
$ cd jumpstart

$ docker run -it debian bin/bash
# cd /tmp
Ctrl+p, Ctrl+q
$ docker cp LeaveMeHere.txt XXX:/tmp
$ docker attach XXX
# ls
```

Dockerfile

- describes how to create a container
- run initial commands
- export ports

Dockerfile Components

- FROM
- MAINTAINER
- RUN
- COPY
- ENTRYPOINT
- WORKDIR
- EXPOSE
- ENV
- VOLUME

Dockerfile Example

```
FROM node:latest
RUN mkdir /src
RUN npm install nodemon@dev -g
WORKDIR /src
EXPOSE 3000
```

Create a Container from a Dockerfile

```
$ code DOCKERFILE

FROM debian:latest
RUN mkdir /src
WORKDIR /src

$ docker build -t myDebian:latest .
$ docker run -it myDebian bin/bash
# pwd
```

Container Linking

- legacy linking
- bridge network

Legacy Linking

```
$ cd locationOfTheUserServiceDemo
$ docker build -t userservice .
$ cd locationOfTheWebsite
$ docker build -t website .
$ docker run -d --name userService userservice
$ docker run -it -p 5000:5000 --link
userService:userService website
```

Bridge Network

```
$ docker network create --driver bridge network_one
$ docker run -d --net=network_one --name userService
userservice
$ docker run -it -p 5000:5000 --net=network_one website
```

Container Composing

- Docker Compose
- create multiple container
- create network bridges
- start with one command

docker-compose.yml

```
version: '2'
services:
  userservice:
    build:
      context: UserSerive/
    networks:
      - service-hub
  website:
    build:
      context: Website
    networks:
      - service-hub
networks:
  service-hub:
    driver: bridge
```

Bridge Network

```
$ docker-compose --help
$ docker-compose build
$ docker-compose up
# Ctrl+C
$ docker-compose up -d
$ docker-compose ps
$ docker-compose logs
$ docker-compose down
$ docker ps -a
```

```
Run 'docker COMMAND --help' for more information on a command.
Tims-MacBook-Pro:test tim$ docker run -it debian /bin/bash
root@7dbaa1b85034:/# apt-get update
Get:1 http://security.debian.org jessie/updates InRelease [63.1 kB]
Get:2 http://security.debian.org jessie/updates/main amd64 Packages [373 kB]
Ign http://httpredir.debian.org jessie InRelease
Get:3 http://httpredir.debian.org jessie-updates InRelease [142 kB]
Get:4 http://httpredir.debian.org jessie Release.gpg [2373 B]
Get:5 http://httpredir.debian.org jessie Release [148 kB]
Get:6 http://httpredir.debian.org jessie-updates/main amd64 Packages [17.6 kB]
Get:7 http://httpredir.debian.org jessie/main amd64 Packages [9032 kB]
Fetched 9778 kB in 5s (1703 kB/s)
Reading package lists... Done
root@7dbaa1b85034:/# apt-get install figlet
Reading package lists... Done
Building dependency tree... Done
The following NEW packages will be installed:
  figlet
0 upgraded, 1 newly installed, 0 to remove and 2 not upgraded.
Need to get 189 kB of archives.
After this operation, 748 kB of additional disk space will be used.
Get:1 http://httpredir.debian.org/debian/ jessie/main figlet amd64 2.2.5-2 [189 kB]
Fetched 189 kB in 0s (205 kB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package figlet.
(Reading database ... 7548 files and directories currently installed.)
Preparing to unpack .../figlet_2.2.5-2_amd64.deb ...
Unpacking figlet (2.2.5-2) ...
Setting up figlet (2.2.5-2) ...
update-alternatives: using /usr/bin/figlet-figlet to provide /usr/bin/figlet (figlet) in auto mode
root@7dbaa1b85034:/# figlet THANK YOU
 _____ _   _    _    _   _ _  __   __   __ ___  _   _
|_   _| | | |  / \  | \ | | |/ /  \ \ / /  / _ \| | | |
  | | | |_| | / _ \ |  \| | ' /    \ V /  | | | | | | |
  | | |  _  |/ ___ \| |\  | . \     | |   | |_| | |_| |
  |_| |_| |_/_/   \_\_| \_|_|\_\    |_|    \___/ \___/

root@7dbaa1b85034:/#
```