Introduction to Docker

About me

Frederik Mogensen

Software Pilot at Trifork

Focus on Docker, orchestration and ci/cd

Agenda

Containers are NOT VMs

Working with Docker (Build, Ship, Run)

Container Architecture

But Why?

Multi-container applications

- Docker Compose
- Docker Swarm

Getting started

Q & A

Containers are not VMs

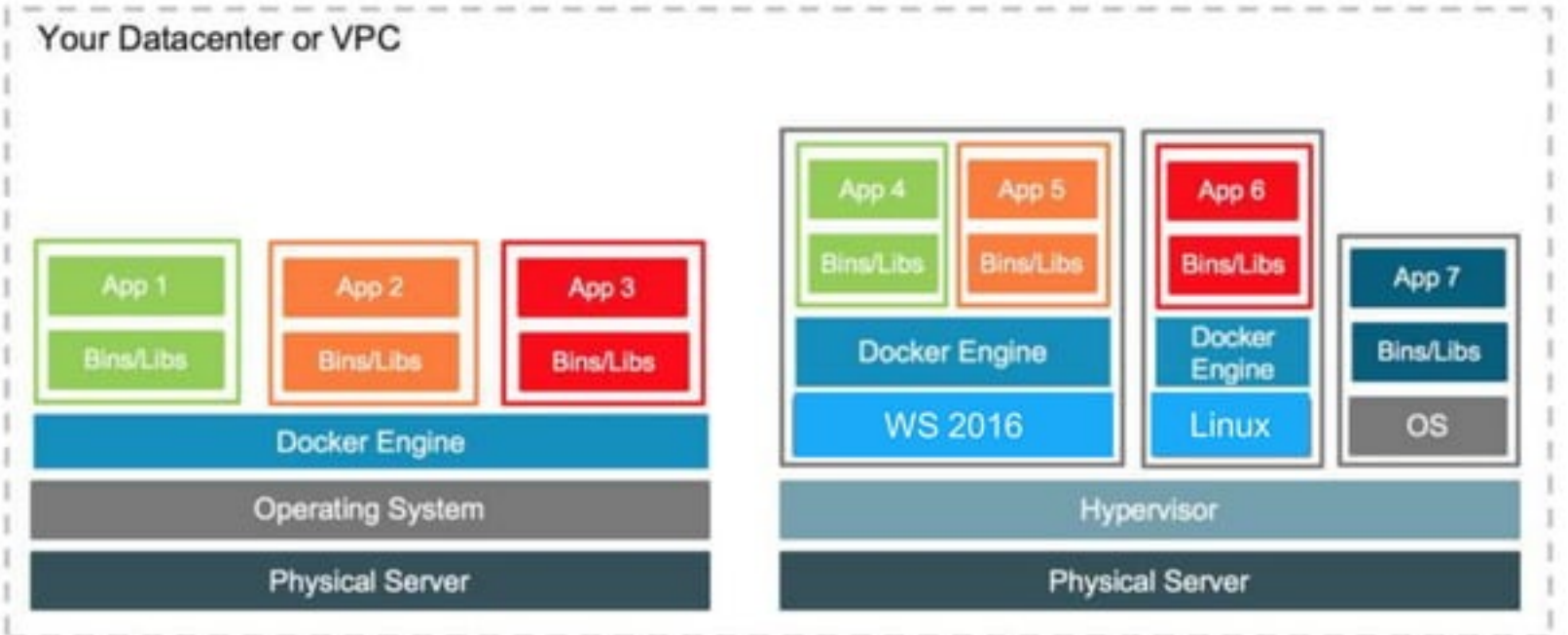
Docker containers are NOT VMs

- Easy connection to make
- Fundamentally different architectures
- Fundamentally different benefits

VMs

Containers

They're different, not mutually exclusive



Build, Ship, and Run

Build, Ship, Run, Any App Anywhere

Some Docker vocabulary

Docker Image

The basis of a Docker container. Represents a full application

Docker Container

The standard unit in which the application service resides and executes

Docker Engine

Creates, ships and runs Docker containers deployable on a physical or virtual, host locally, in a datacenter or cloud service provider

Registry Service (Docker Hub or Docker Trusted Registry)

Cloud or server based storage and distribution service for your images

Basic Docker Commands

```
$ docker pull mikegcoleman/catweb:latest

$ docker images

$ docker run -d -p 5000:5000 --name catweb mikegcoleman/catweb:latest

$ docker ps

$ docker stop catweb (or <container id>)

$ docker rm catweb (or <container id>)

$ docker rmi mikegcoleman/catweb:latest (or <image id>)
```

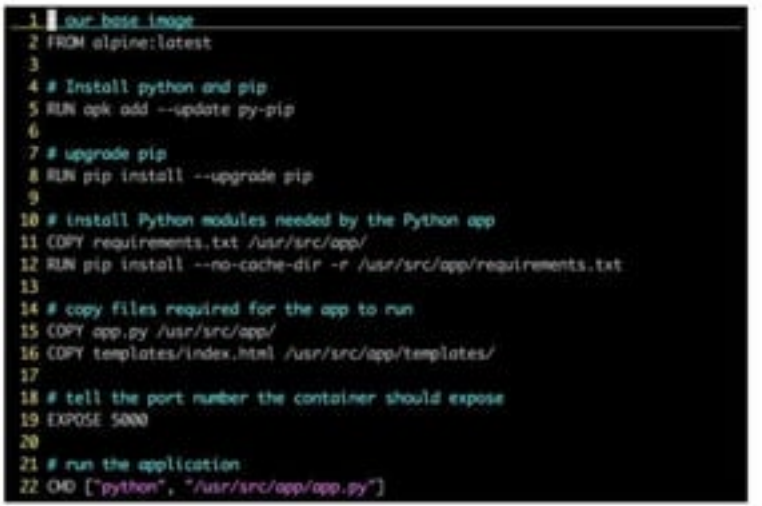
Dockerfile – Linux Example

```
# our base image
FROM alpine:latest

# Install python and pip
RUN apk add --update py-pip

# upgrade pip
RUN pip install --upgrade pip

# install Python modules needed by the Python app
COPY requirements.txt /usr/src/app/
RUN pip install --no-cache-dir -r /usr/src/app/requirements.txt

# copy files required for the app to run
COPY app.py /usr/src/app/
COPY templates/index.html /usr/src/app/templates/

# tell the port number the container should expose
EXPOSE 5000

# run the application
CMD ["python", "/usr/src/app/app.py"]
```

- Instructions on how to build a Docker image
- Looks very similar to “native” commands
- Important to optimize your Dockerfile

Basic Docker Commands

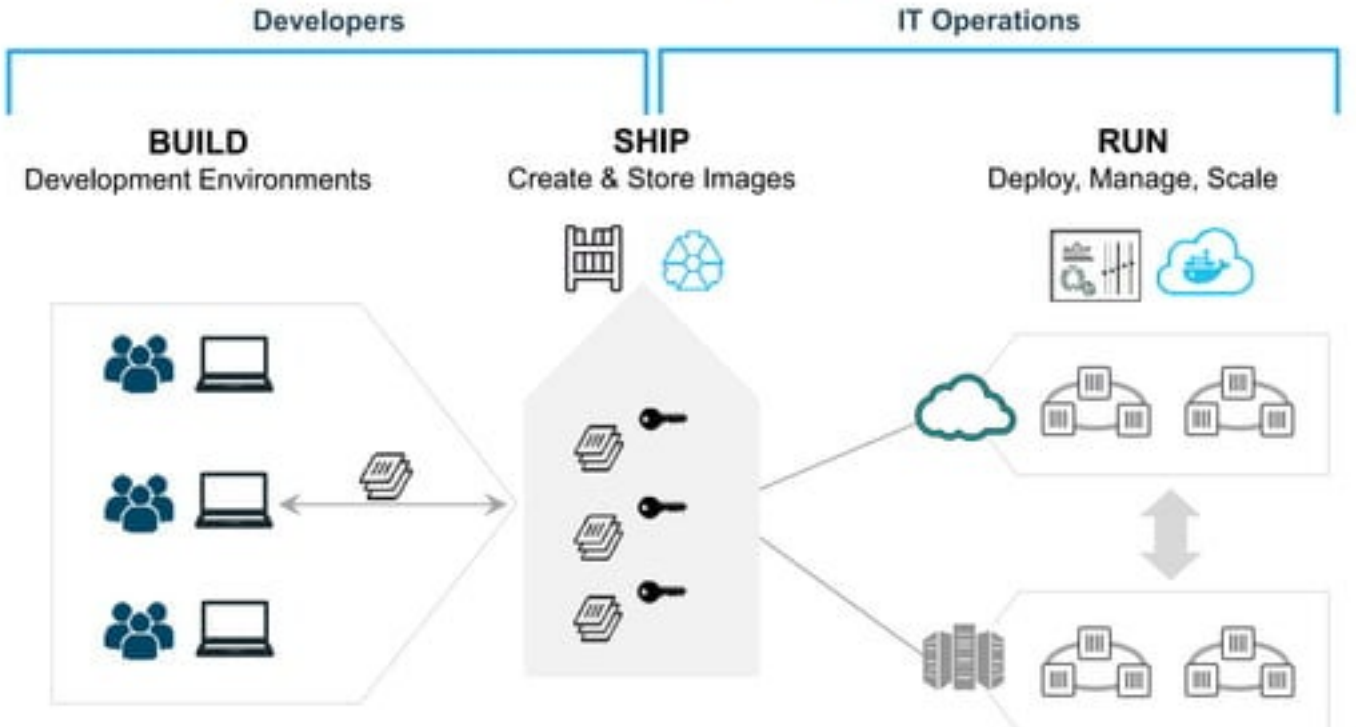
```
$ docker build -t mikegcoleman/catweb:2.0 .
$ docker push mikegcoleman/catweb:2.0
```



```
1  our base image
2  FROM alpine:latest
3
4  # Install python and pip
5  RUN apk add --update py-pip
6
7  # upgrade pip
8  RUN pip install --upgrade pip
9
10 # install Python modules needed by the Python app
11 COPY requirements.txt /usr/src/app/
12 RUN pip install --no-cache-dir -r /usr/src/app/requirements.txt
13
14 # copy files required for the app to run
15 COPY app.py /usr/src/app/
16 COPY templates/index.html /usr/src/app/templates/
17
18 # tell the port number the container should expose
19 EXPOSE 5000
20
21 # run the application
22 CMD ["python", "/usr/src/app/app.py"]
```

docker

Put it all together: Build, Ship, Run Workflow



docker

Docker Container Architecture

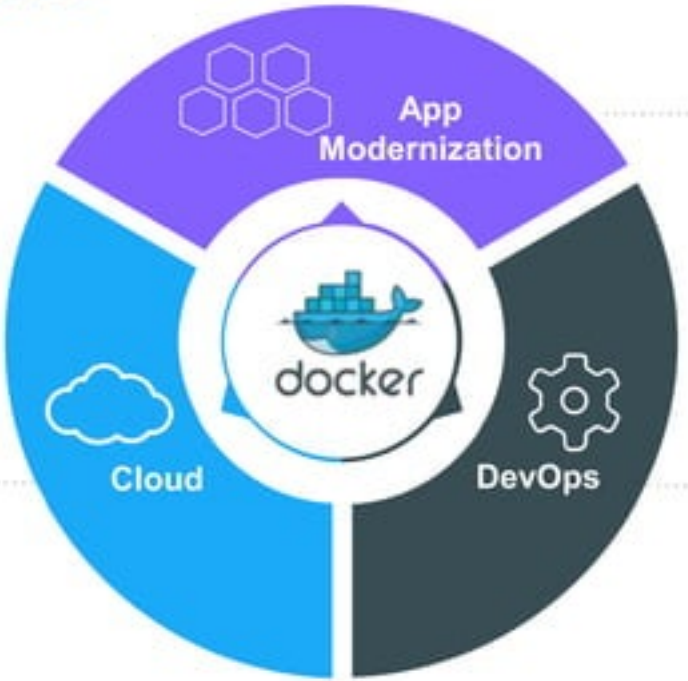
Image Layers

What about data persistence?

- Volumes allow you to specify a directory in the container that exists outside of the docker file system structure
- Can be used to share (and persist) data between containers
- Directory persists after the container is deleted
  - Unless you explicitly delete it
- Can be created in a Dockerfile or via CLI

But, Why?

Enterprises are looking to Docker for critical transformations



3 out 4

Top initiatives revolve around applications

80%

Docker is central to cloud strategy

44%

Looking to adopt DevOps

docker

State of App development Survey: Q1 2016

Docker delivers speed, flexibility and savings

Agility

13X

More software releases

65%

Reduction in developer onboarding time

Portability

41%

Move workloads across private/public clouds

Eliminate

"works on my machine" issues

Control

62%

Report reduction in MTTR

10X

Cost reduction in maintaining existing applications

docker

State of App development Survey: Q1 2016, Cornell University case study

One platform delivers one journey for all applications

1. **Containerize Legacy Applications**
   Lift and shift for portability and efficiency

2. **Transform Legacy to Microservices**
   Look for shared services to transform

3. **Accelerate New Applications**
   Greenfield innovation

Docker Compose

Defining and running multi-container Docker applications

Multiple container application in Docker

```
$ docker pull mysql

$ docker pull wordpress

$ docker run -d --name=db -e MYSQL_ROOT_PASSWORD=root mysql

$ docker run --name=wp -p 8000:80 --link db:db \
      -e WORDPRESS_DB_HOST=db \
      -e WORDPRESS_DB_PASSWORD=root wordpress
```

Docker Compose - YAML

```
$ docker pull mysql

$ docker pull wordpress

$ docker run -d --name=db
        -e MYSQL_ROOT_PASSWORD=root mysql

$ docker run --name=wp  \
        -p 8000:80 \
        --link db:db \
        -e WORDPRESS_DB_HOST=db \
        -e WORDPRESS_DB_PASSWORD=root  \
        wordpress
```

```
version: '2'
services:
  db:
   image: mysql
   environment:
    MYSQL_ROOT_PASSWORD: root
  wp:
   depends_on:
    - db
   image: wordpress
   ports:
    - "8000:80"
   environment:
    WORDPRESS_DB_HOST: db
    WORDPRESS_DB_PASSWORD: root
```

docker

Docker Compose - YAML

```
$ docker-compose up

$ docker-compose ps

$ docker-compose stop
```

```
version: '2'
services:
  db:
   image: mysql
   environment:
    MYSQL_ROOT_PASSWORD: root
  wp:
   depends_on:
    - db
   image: wordpress
   ports:
    - "8000:80"
   environment:
    WORDPRESS_DB_HOST: db
    WORDPRESS_DB_PASSWORD: root
```

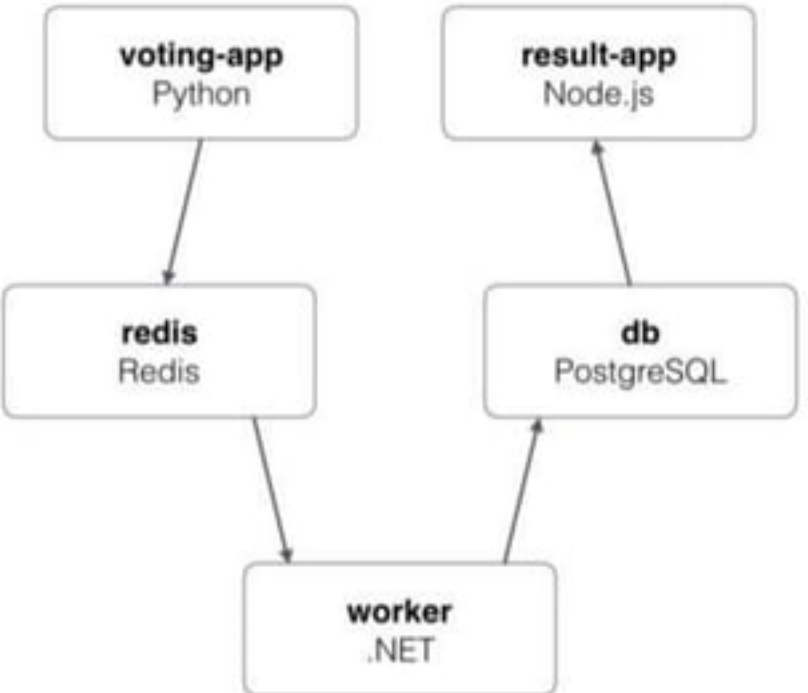
Docker Swarm

Setting up a basic docker cluster

Docker Swarm mode

```
// http://play-with-docker.com
$ docker swarm init --advertise-addr eth0

// join nodes
$ docker swarm join ...
$ docker node ls
```

The Vote Application



docker

Docker Swarm mode

```
// download stack definition
$ curl -O https://raw.githubusercontent.com/docker/example-voting-app/master/docker-stack.yml

// Spin up cluster
$ docker stack deploy -c docker-stack.yml vote
```

Getting started!

Docker on Linux

- Create a Linux VM (or use physical), and install Docker
  - Requires kernel 3.10
- Stable builds
  - `curl -sSL https://get.docker.com/ | sh`
- Test and experimental builds
  - `curl -sSL https://test.docker.com/ | sh`
  - `curl -sSL https://experimental.docker.com/ | sh`
- Can also manually install (see docs)

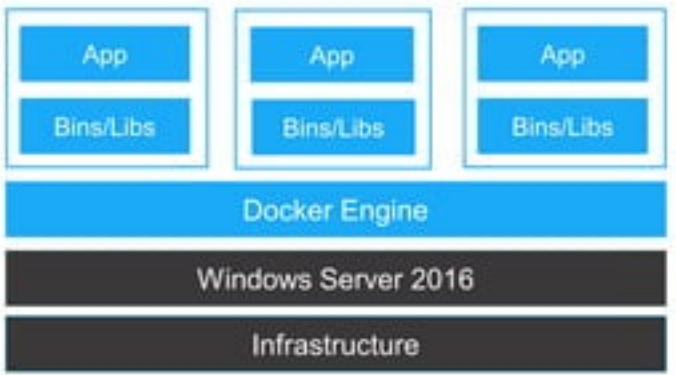
Docker for Windows / Mac

- Currently in public beta
- Easy to install: Get up and running on Docker in minutes
- Leverages Hyper-V (Windows) or xhyv (Mac)
  - Docker for Windows requires Windows Pro 10, Enterprise, or Education
- Full API / CLI compatibility
- OS integration for increased stability and speed

Docker for Azure / AWS

- Easily deploy Docker 1.12 Swarm clusters (Linux)
- Scale up and down easily
- Integrate with underlying platform (i.e. load balancers)

Docker + Windows Server = Windows Containers



- Native Windows containers powered by Docker Engine
- Windows kernel engineered with new primitives to support containers
- Deep integration with 2+ years of engineering collaboration in Docker Engine and Windows Server
- Microsoft is top 5 Docker open source project contributor and a Docker maintainer

docker

Walk, Jog, Run

Walk:

- Setup your preferred Docker environment
- Fire up some prebuilt images (nginx, hello-world, mikegcoleman/catweb)

Jog:

- Pick a well documented solution (Wordpress, Jenkins, etc)
- Build it for yourself (blogs are your friend)

Run:

- Extend one your Walk solution or Dockerize an existing project
- Build your own Dockerfiles
- Experiment with Docker Compose and Swarm Mode

Where to go from here?

https://github.com/docker/labs
https://prakhar.me/docker-curriculum/
https://europe-2017.dockercon.com/

Thank You.

Questions?

docker