Getting to DevOps with Docker
Brian (bex) Exelbierd

- Software Engineer @ Red Hat
- Various Roles in IT since 1995
  - Programmer
  - Analyst
  - Manager (Ops, Dev, Special Projects)
  - Sales Engineering
  - etc.
- Work on Project Atomic: Tools that make containers easier

$ whoami
@bexelbie

Slides URL: www.winglemeyer.org
Question Time
DevOps
What is DevOps?

- Culture, not tools
- You can’t buy DevOps
- If you’re using Docker, you’re not necessarily DevOps
- No one’s title is now DevOps

```
$ sudo dnf install DevOps
No package DevOps available.
Error: Unable to find a match.
```
Endless Possibilities: DevOps can create an infinite loop of release and feedback for all your code and deployment targets.

https://blog.appdynamics.com/devops/devops-scares-me-part-2/
So, what is DevOps?

- Collaboration of People
- Convergence of Process
- Creation & Exploitation of Tools

Ben Rockwood
Director of IT & Operations at Chef

- https://www.youtube.com/watch?v=h5E--QSBVBY
So, what is DevOps?

- Collaboration of People
- Convergence of Process
- Creation & Exploitation of Tools

Ben Rockwood
Director of IT & Operations at Chef

- [https://www.youtube.com/watch?v=h5E--QSBVBY](https://www.youtube.com/watch?v=h5E--QSBVBY)
So, what is DevOps?

- Collaboration of People
- Convergence of Process
- Creation & Exploitation of Tools

Ben Rockwood
Director of IT & Operations at Chef

- [https://www.youtube.com/watch?v=h5E--QSBVBY](https://www.youtube.com/watch?v=h5E--QSBVBY)
So, what is DevOps?

It’s about flow

- Collaboration of People
- Convergence of Process
- Creation & Exploitation of Tools

Ben Rockwood
Director of IT & Operations at Chef

- https://www.youtube.com/watch?v=h5E--QSBVBY
Why DevOps? What Problem(s) does it Solve?

● Developers
  ○ Differences in Test/Production lead to Dependency Errors [Portability]
    ■ “It works on my laptop!”
  ○ Don’t want to wait a long time for code to get to production [Deployment]
    ■ slows down feedback cycle
    ■ multiple code bases

● Operations
  ○ New Code never seems to fit into production exactly [Controlled Infrastructure]
    ■ a/k/a “You can’t just rev the httpd version you need and not tell anyone”
  ○ Scale out has led to an increase in servers to manage [Scale Out]

Project/Business Win: Faster Time to Market

Resource: Rack Space Video: https://www.youtube.com/watch?time_continue=41&v=_I94-tJlovg
Docker
What is Docker?

Docker containers wrap up a piece of software in a complete filesystem that contains everything it needs to run. (docker.com)

Remember These?

- S/370 LPARs
- AIX WPARs
- BSD Jails
- Solaris Zones
- chroot ...

Docker is a way of packaging software and accessing Linux kernel features like cgroups, namespaces, capabilities, etc.
Thinking in Docker

- Virtual Machine Lite
  - Initially people thought it was a faster VM technology
  - Containers with
    - ssh
    - init
    - daemons, etc.

- Application in a box
  - Delete ssh, daemons
  - Lots of processes with init
  - Databases + servers + ...

- Microservices
  - Like Service-Oriented Architecture (SOA)
  - Minimal unit of an application
  - Helps with scale out
What about my data? What about Configs

Pets vs. Cattle
What about my data? What about Configs

Pets vs. Cattle

Scotch vs. Beer
What about my data? What about Configs

Pets vs. Cattle  Slivovice vs. Slivovice  Scotch vs. Beer
What about my data? What about Configs

- Pets vs. Cattle
- Scotch vs. Beer
- Slivovice vs. Slivovice
What about my data? What about Configs

Pets vs. Cattle

Slivovice vs. Slivovice

Scotch vs. Beer

Images: Brian Exelbierd; itesco.cz
Docker Vocabulary

Image: An immutable read-only template of a container. This is the distributable object.

What does an image consist of?
- A tar file of the filesystem for the layer(s)
- Metadata (image name, version, etc.)

Layer: Images are made with copy on write union file systems that create layers when you make modifications. This means you can start with a base image and layer your software over the top. This also means only your changes have to be distributed.

Base Image: An image containing enough of the libraries and binaries of an OS to support running software.

Registry: A public or private store for images used for network distribution.

Container: An image that has been instantiated. The isolated run-time unit.

<table>
<thead>
<tr>
<th>add Your App</th>
</tr>
</thead>
<tbody>
<tr>
<td>add node.js</td>
</tr>
<tr>
<td>CentOS Base Image</td>
</tr>
</tbody>
</table>
Getting and Managing Images

Docker Hub (hub.docker.com) - public registry of over 100,000 different images

- 2708 apache images, non-official
- Not signed yet

Remember: Images are templates

# Search for images
$ docker search apache

# Download images
$ docker pull centos

# List all images on your machine
$ docker images

# Remove images from your machine
$ docker rmi <ID|Name>
Running and Managing Containers

Run Options of Note:

- 

    -i Keep STDIN open even when not attached
    -t Allocate a pseudo-tty
    --rm Automatically remove a container when it stops
    --name=<name> Use <name>
    -e VAR=VALUE Set environment variables
    -d Detach container and run in background
    -p <hport>:<cport> map a host port to a container port
    --help Help :)
Building Images

Dockerfile specifies build directives

FROM - A starting image (can be a base image or any other image)

RUN - execute this command in the image

EXPOSE - make a port available

ADD - Move files from the build host into the image

CMD - default command to be run when the image is started (There was no command in our example …)

MAINTAINER - metadata

FROM fedora:20

MAINTAINER http://fedoraproject.org/wiki/Cloud

RUN yum -y update && yum clean all

RUN yum -y install httpd && yum clean all

RUN echo "Apache" >> /var/www/html/index.html

EXPOSE 80

# Simple startup script to avoid some issues observed with container restart

ADD run-apache.sh /run-apache.sh

RUN chmod -v +x /run-apache.sh

CMD ["/run-apache.sh"]
Building Images

Best Practices are being developed

- https://github.com/projectatomic/container-best-practices
- https://docs.docker.com/articles/dockerfile_best-practices/

1. Old Fedora Version
2. update in container considered sub-optimal
3. Combine yum commands to reduce layers
4. Label it with meta-data
   https://github.com/projectatomic/ContainerApplication
   GenericLabels

FROM fedora:20
MAINTAINER http://fedoraproject.org/wiki/Cloud
RUN yum -y update && yum clean all
RUN yum -y install httpd && yum clean all
RUN echo "Apache" &gt; /var/www/html/index.html
EXPOSE 80
# Simple startup script to avoid some issues observed with container restart
ADD run-apache.sh /run-apache.sh
RUN chmod -v +x /run-apache.sh
CMD ["/run-apache.sh"]

LABEL VERSION="1.0"
LABEL RUN="docker run -d -p 8080:80 \${IMAGE}"
How do I link Microservices? What about my Data?

Option 1: Docker Linking

$ docker run --link DBC webserver

Creates a private networking link between the DBC (database container) and the webserver. Helpful Environment variables for ports, etc.

Option 2: Orchestration

- Kubernetes
- Mesos (Marathon)
- Docker Swarm
- ...

Option 1: Docker Volumes

$ docker run -v /webdata:/var/www apache

Make the data from the host’s /webdata available via a mount to the container.

Option 2: Volume containers

Data is mounted (--volumes-from) from another container.

Option 3: Orchestration Provider/Persistent Storage

Look at your provider, check out things like Ceph/Gluster with containers
Why DevOps? What Problem(s) does it Solve?

● Developers
  ○ Differences in Test/Production lead to Dependency Errors [Portability]
    ■ “It works on my laptop!”
  ○ Don’t want to wait a long time for code to get to production [Deployment]
    ■ slows down feedback cycle
    ■ multiple code bases

● Operations
  ○ New Code never seems to fit into production exactly [Controlled Infrastructure]
    ■ a/k/a “You can’t just rev the httpd version you need and not tell anyone”
  ○ Scale out has led to an increase in servers to manage [Scale Out]

Project/Business Win: Faster Time to Market

Resource: Rack Space Video: https://www.youtube.com/watch?time_continue=41&v=_I94-tJlovg
Portability

bexelbie@bexelbie:~$ cat /etc/fedora-release
Fedora release 22 (Twenty Two)
bexelbie@bexelbie:~$ uname -a
Linux bexelbie 4.1.6-201.fc22.x86_64 #1 SMP Fri Sep 4 17:49:24 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
bexelbie@bexelbie:~$ docker run -i -t --rm centos bash
[root@bcd983bbeb57 ~]# cat /etc/centos-release
CentOS Linux release 7.1.1503 (Core)
[root@bcd983bbeb57 ~]# uname -a
Linux bcd983bbeb57 4.1.6-201.fc22.x86_64 #1 SMP Fri Sep 4 17:49:24 UTC 2015 x86_64 x86_64 x86_64 GNU/Linux
Why DevOps? What Problem(s) does it Solve?

- **Developers**
  - Differences in Test/Production lead to Dependency Errors [Portability]
  - “It works on my laptop!”
  - Don’t want to wait a long time for code to get to production [Deployment]
    - slows down feedback cycle
    - multiple code bases

- **Operations**
  - New Code never seems to fit into production exactly [Controlled Infrastructure]
    - a/k/a “You can’t just rev the httpd version you need and not tell anyone”
  - Scale out has led to an increase in servers to manage [Scale Out]

Project/Business Win: Faster Time to Market

Resource: Rack Space Video: https://www.youtube.com/watch?time_continue=41&v=_I94-tJlovg
Deployment

- Designed for automated build
- Pushes you to a model for easy use from a git repo (Dockerfile + source)
- Jenkins/etc. already working with it
- Project Atomic’s Nulecule is formalizing multi-container application definition
Why DevOps? What Problem(s) does it Solve?

● Developers
  ○ Differences in Test/Production lead to Dependency Errors [Portability]
    ■ “It works on my laptop!”
  ○ Don’t want to wait a long time for code to get to production [Deployment]
    ■ slows down feedback cycle
    ■ multiple code bases

● Operations
  ○ New Code never seems to fit into production exactly [Controlled Infrastructure]
    ■ a/k/a “You can’t just rev the httpd version you need and not tell anyone”
  ○ Scale out has led to an increase in servers to manage [Scale Out]

Project/Business Win: Faster Time to Market

Resource: Rack Space Video: https://www.youtube.com/watch?time_continue=41&v=_l94-tJlovg
$ cat Dockerfile
FROM mycorp/node:1.0
RUN dnf install custom-node-library
ADD node-app

$ cat Dockerfile
FROM mycorp/node:1.0
RUN npm install scary_lib
ADD node-app
Why DevOps? What Problem(s) does it Solve?

- **Developers**
  - Differences in Test/Production lead to Dependency Errors [Portability]
    - “It works on my laptop!”
  - Don’t want to wait a long time for code to get to production [Deployment]
    - slows down feedback cycle
    - multiple code bases

- **Operations**
  - New Code never seems to fit into production exactly [Controlled Infrastructure]
    - a/k/a “You can’t just rev the httpd version you need and not tell anyone”
  - Scale out has led to an increase in servers to manage [Scale Out]

Project/Business Win: Faster Time to Market

Resource: Rack Space Video: https://www.youtube.com/watch?time_continue=41&v=_I94-tJlovg
Scale Out

- Fast to start and stop
- *Slivovice vs. Slivovice* means design supports scale from the start
- Orchestration providers
Thank you
Brian (bex) Exelbierd
@bexelbie
Slides: www.winglemeyer.org
## Fake Demo: Command #1

```bash
$ docker search apache
```

<table>
<thead>
<tr>
<th>INDEX</th>
<th>NAME</th>
<th>DESCRIPTION</th>
<th>STARS</th>
<th>OFFICIAL</th>
<th>AUTOMATED</th>
</tr>
</thead>
<tbody>
<tr>
<td>docker.io</td>
<td>docker.io/tomcat</td>
<td>Apache Tomcat is an op...</td>
<td>299</td>
<td></td>
<td>[OK]</td>
</tr>
<tr>
<td></td>
<td>docker.io/fedora/apache</td>
<td></td>
<td>33</td>
<td></td>
<td>[OK]</td>
</tr>
<tr>
<td></td>
<td>docker.io/eboraas/apache</td>
<td>Apache (with SSL on Debian)</td>
<td>22</td>
<td></td>
<td>[OK]</td>
</tr>
<tr>
<td></td>
<td>docker.io/bitnami/apache</td>
<td>Bitnami Apache Docker Image</td>
<td>9</td>
<td></td>
<td>[OK]</td>
</tr>
</tbody>
</table>

...
$ docker pull centos
Using default tag: latest
Trying to pull repository docker.io/library/centos ... latest: Pulling from library/centos
47d44cb6f252: Pull complete
168a69b62202: Pull complete
812e9d9d677f: Pull complete
4234bfdd88f8: Pull complete
ce20c473cd8a: Pull complete
library/centos:latest: The image you are pulling has been verified. Important: image verification is a tech preview feature and should not be relied on to provide security.
Digest: sha256:3aaab9f1297db9b013063c781cfe901e2aa6e7e334c1d1f4df12f25ce356f2e5
Status: Downloaded newer image for docker.io/centos:latest
$ docker pull centos:6.7
Trying to pull repository docker.io/library/centos ... 6.7: Pulling from library/centos
5fc6f5013018: Pull complete
8e6730e0eaef: Pull complete
b89573a5b116: Pull complete
3fba1048142f: Pull complete
47d44cb6f252: Already exists
library/centos:6.7: The image you are pulling has been verified. Important: image verification is a tech preview feature and should not be relied on to provide security.
Digest: sha256:89d9204927e3ebbe7d93fb7b07b86d2ab5502c31e9c964cb995d6d4fd1ea3039
Status: Downloaded newer image for docker.io/centos:6.7
$ docker images

<table>
<thead>
<tr>
<th>REPOSITORY</th>
<th>TAG</th>
<th>IMAGE ID</th>
<th>CREATED</th>
<th>VIRTUAL SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>docker.io/centos</td>
<td>latest</td>
<td>ce20c473cd8a</td>
<td>5 days ago</td>
<td>172.3 MB</td>
</tr>
<tr>
<td>docker.io/centos</td>
<td>centos6.7</td>
<td>3fba1048142f</td>
<td>5 days ago</td>
<td>190.6 MB</td>
</tr>
<tr>
<td>docker.io/jekyll/jekyll</td>
<td>latest</td>
<td>44d4bdcd6f669</td>
<td>5 weeks ago</td>
<td>145 MB</td>
</tr>
<tr>
<td>docker.io/redis</td>
<td>latest</td>
<td>2f2578ff984f</td>
<td>5 weeks ago</td>
<td>109.2 MB</td>
</tr>
<tr>
<td>docker.io/nginx</td>
<td>latest</td>
<td>0b354d33906d</td>
<td>5 weeks ago</td>
<td>132.8 MB</td>
</tr>
<tr>
<td>docker.io/mysql</td>
<td>latest</td>
<td>6762f304c834</td>
<td>5 weeks ago</td>
<td>283.5 MB</td>
</tr>
<tr>
<td>docker.io/fedora</td>
<td>latest</td>
<td>ded7cd95e059</td>
<td>4 months ago</td>
<td>186.5 MB</td>
</tr>
</tbody>
</table>
Fake Demo: Command #4

$ docker rmi centos:6.7
Untagged: centos:6.7
Deleted: 3fb1048142f7f89f67f2b6b11256053a3beaa280b97538dd85d51d4f0a65961
Deleted: b89573a5b116e61624906884fc48ba0cd7037a72cf1d2757c77fbd73f03c150a
Deleted: 8e6730e0eaeaf34246dd562b1eccc41ab72012a1bab74996edd4b5783bbfe71b82
Deleted: 5fc6f5013018fd5f1e84a3b5d304f03cfb81b6131ca20c968262bc60c2edb107
Fake Demo: Command #5

$ docker run -d -p 8080:80 fedora/apache
c20ee8740ab0342fcb5e9ff9c948a07b57734c692bbc57c0d7ac7b6461ec4dee

$ docker ps
CONTAINER ID   IMAGE               COMMAND                  CREATED               STATUS        PORTS               NAMES
"/run-apache.sh" 32 seconds ago Up 31 seconds 0.0.0.0:8080-80/tcp-naughty_carson

$ curl localhost:8080
Apache
## Fake Demo: Command #6

```bash
$ docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
--- -------- --------------- ------------- --------- ------
c20ee8740ab0 fedora/apache /run-apache.sh 57 seconds ago Up 56 seconds 0.0.0.0:8080->80/tcp naughty_carson
179273eba685 mysql /entrypoint.sh mysql 29 minutes ago Exited (137) 35 seconds ago some-mysql
b8110e2d14f9 53e2c71cae40dc932e4927cc5f0c938aef8e0c8d0fd1f18e568b98f7c6cde318 /bin/true 3 days ago Created cranky_goodall
b721362b0c0ba fedora /bin/bash 3 days ago Exited (0) 3 days ago mnt_test
```
Fake Demo: Command #7

$ docker stop naughty_carson
naughty_carson

$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS
PORTS NAMES

$ docker ps -a | grep -e 'carson\|CONTAINER'
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
c20ee8740ab0 fedora/apache "/run-
apache.sh" 8 minutes ago Exited (0) About a minute ago
naughty_carson

...
```
$ docker rm naughty_carson
naughty_carson

$ docker ps -a | grep -e 'carson\|CONTAINER'
CONTAINER ID    IMAGE          COMMAND       CREATED       STATUS    PORTS NAMES
NAMES

$ docker images | grep apache
docker.io/fedora/apache latest 1eff270e703a 3 months ago
649.7 MB
```
$ docker build -t fed_apache_test .
Sending build context to Docker daemon 23.55 kB
Step 0 : FROM fedora:20
    ----> 0d071bb732e1
Step 1 : MAINTAINER http://fedoraproject.org/wiki/Cloud
    ----> Running in 4f52dc14f7cf
    ----> 68c6cfc842c3
Removing intermediate container 4f52dc14f7cf
Step 2 : RUN yum -y update && yum clean all
    ----> Running in 1f4dce793c25
No packages marked for update
Cleaning repos: fedora updates
Cleaning up everything
    ----> dc61cc0c770b
Removing intermediate container 1f4dce793c25
Fake Demo: Command #9 2/3

Step 3 : RUN yum -y install httpd && yum clean all
    ---> Running in 5836318d3d9c
Resolving Dependencies
---> Running transaction check
    ---> Package httpd.x86_64 0:2.4.10-2.fc20 will be installed
   x86_64
   ...

Complete!
Cleaning repos: fedora updates
Cleaning up everything
    ---> fd611aaea307
Removing intermediate container 5836318d3d9c
    ---> Running in 3bd0cef73706
    ---> 106033d132d7
Removing intermediate container 3bd0cef73706
Fake Demo: Command #9 3/3

Step 5 : EXPOSE 80
   ---> Running in a74a8b9c8ef3
   ---> d6d79e693080
Removing intermediate container a74a8b9c8ef3
Step 6 : ADD run-apache.sh /run-apache.sh
   ---> ed5f0bd13f85
Removing intermediate container 7eb0357f7d54
Step 7 : RUN chmod -v +x /run-apache.sh
   ---> Running in be6bc6501f5d
mode of '/run-apache.sh' changed from 0664 (rw-rw-r--) to 0775 (rwxrwxr-x)
   ---> 2e2cf3065cbb
Removing intermediate container be6bc6501f5d
Step 8 : CMD /run-apache.sh
   ---> Running in 8a5259ac3e61
   ---> cd0882400d7c
Removing intermediate container 8a5259ac3e61
Successfully built cd0882400d7c
Is it Really Fast?

$ time sudo docker run -it --rm fedora sleep 5
real 0m6.200s
user 0m0.023s
sys 0m0.022s

$ time sleep 5
real 0m5.004s
user 0m0.000s
sys 0m0.001s
Thank you
Brian (bex) Exelbierd
@bexelbie
Slides: www.winglemeyer.org