Docker: A lot changed in a year

Chris Swan, CTO
@cpswan
cohesiveFT

Cloud native networking
TL;DR

Docker is important for DevOps because it tightens iteration loops

The core project developed very fast, but the ecosystem is now moving faster

They did it with Docker
Why me?
Grid, Cloud, Docker
Docker inside

VNS3 Core Components

- Proxy
- Reverse Proxy
- Content Caching
- Load Balancing
- IDS
- Custom Container

- Router
- Switch
- Firewall
- VPN Concentrator
- Protocol Redistributor
- Dynamic & Scriptable SDN
This is a DevOps track, so why do we care about Docker?
John Boyd’s OODA loop
In the beginning
0.1.0 (2013-03-23)

Used LXC

Registry to push/pull images
Where were we a year ago?
0.6.6 (2013-11-06)

Runtime
- Ensure container name on register
- Fix regression in /etc/hosts
- Add lock around write operations in graph
- Check if port is valid
- Fix restart runtime error with ghost container networking
- Add some more colors and animals to increase the pool of generated names
- Fix issues in docker inspect
- Escape apparmor confinement
- Set environment variables using a file.
- Prevent docker insert to erase something
- Prevent DNS server conflicts in CreateBrigdelface
- Validate bind mounts on the server side
- Use parent image config in docker build
- Fix regression in /etc/hosts

Client
- Add -P flag to publish all exposed ports
- Add -notrunc and -q flags to docker history
- Fix docker commit, tag and import usage
- Add stars, trusted builds and library flags in docker search
- Fix docker logs with tty

RemoteAPI
- Make /events API send headers immediately
- Do not split last column docker top
- Add size to history

Other
- Contrib: Desktop integration. Firefox usecase.
- Dockerfile: bump to go1.2rc3
We still didn’t have

• A regular release cycle
• Compatibility with standard Linux distros
• Links
• Naming

Much of that came in 0.7.0 towards the end of Nov 2013
The watershed release
0.9.0 (2014-03-10)

Libcontainer

(LXC gets thrown under the bus)
0.9.1 (2014-03-24)

Ubuntu 14.04

Centos 7

Etc.
And then came DockerCon
1.0.0 (2014-06-09)

Notable features since 0.12.0 (2014-06-05)

• Production support
libcontainer
libchan
Since 1.0.0

exec

create
Shocker, Flocker, Clocker
Shocker

The issue

http://stealth.openwall.net/xSports/shocker.c

The response

https://news.ycombinator.com/item?id=7910117
Flocker moves a database from node1 to node2, re-routes network connections to new location.
Clocker
Composition
```yaml
fig.yml

todomvcdb:
  image: cpswan/todomvc.mysql
  expose:
    - "3306"
  volumes:
    - /data/mysql:/var/lib/mysql
todomvcapp:
  image: cpswan/todomvc.sinatra
  expose:
    - "4567"
  links:
    - todomvcdb:db
todomvcssl:
  image: cpswan/todomvc.ssl
  ports:
    - "443:443"
  links:
    - todomvcapp:app
```
geard
Panamax

CenturyLink Labs - http://www.centurylinklabs.com/
Please select one of the following options:
1) init: First time installing Panamax! - Downloads CoreOS VM and installs latest Panamax version.
2) pause: Stops Panamax
3) up: Starts Panamax
4) restart: Stops and Starts Panamax.
5) reinstall: Deletes your applications and CoreOS VM; reinstalls to latest Panamax version.
6) info: Displays version of your local panamax install.
7) check: Checks for available updates for Panamax.
8) download: Updates to latest Panamax version.
9) delete: Uninstalls Panamax, deletes applications and CoreOS VM.
10) ssh: Ssh into Panamax CoreOS VM.
11) help: Show this help
12) quit

copyright 2014
PaaS caught Docker
Docker based PaaS

DEIS

progrium / dokku

Docker powered mini-Heroku in around 100 lines of Bash

Flynn
Docker in PaaS

Decker = Docker + Cloud Foundry
PaaS == Opinionated?
Docker might be the anti PaaS

```
FROM ubuntu:12.04
MAINTAINER cpswan
# Add universe repository to /etc/apt/sources.list
# we need it for nginx
RUN sed -i 's/main/\main universe/\n/etc/apt/sources.list'
# Update repos so that changes can take effect
RUN apt-get update
# Install nginx
RUN apt-get install -y nginx
# Turn off daemon mode
RUN echo "\ndaemon off;" >> /etc/nginx/nginx.conf
# Meetup customisation
RUN sed -i 's/nginx/\'Chicago Docker Meetup'/\n/usr/share/nginx/www/index.html'
# Expose web server
EXPOSE 80
# Run nginx
CMD /usr/sbin/nginx
```
PaaS == Operationalised?
Orchestration
Everybody has one...

Brad Rydzewski
@bradrydzewski

what I learned at #dockercon: everyone is building their own orchestration platform. seriously. everyone
Kubernetes is the one to watch
This week…

Google Container Engine (Alpha)

Run Docker containers on Google Cloud Platform, powered by Kubernetes. Container Engine takes care of provisioning and maintaining the underlying virtual machine cluster, scaling your application, and operational logistics like logging, monitoring, and health management.
… not containerised cloud (yet)

Pricing

Container Engine uses Google Compute Engine instances for nodes in the cluster, including the cluster master (created automatically for every cluster) and the number of worker nodes you specify. You will be billed for those instances according to Compute Engine's pricing, until the clusters are deleted. During this alpha phase, there is no additional charge for Container Engine on top of the Compute Engine pricing.
So why does it matter?
Build, Ship > Run?

Docker Hub

Bottled software

Image credit CC Julio Morales [https://www.flickr.com/photos/phooleo/4703216251](https://www.flickr.com/photos/phooleo/4703216251)
There’s still a bunch of work to be done
Containers don’t contain

With thanks to Dan Walsh @rhatdan
Watch his DockerCon 2014 presentation at http://is.gd/dcrhdw
The manifest problem
Each active line creates a layer
An image binds layers together

- Base OS
- Sources
- Update repos
- Install nginx
- Mod nginx.conf
- Mod index.html
The image is the unit of deployment

Nginx example
What version of nginx is that?

Nginx example
What version of OpenSSL installed?

Nginx example

?
and which bash?

Nginx example
My Docker wish list
Feb 2014

Docker CLI

Disk quotas

Route propagation
Nov 2014

Docker CLI
Disk quotas
Finer life cycle control
If you want a better Docker

Learn Go
And another thing...
Windows

Windows 10
Governance

1. Mature tooling/CI process so project can scale (more)

2. Dealing with long outstanding pull requests

3. Clear separation between Docker the company and Docker the project (and clarification of MS relationship)

4. Clear articulation of roadmap
Maintainers

Solomon Hykes
@solomonstre

A maintainer is a contributor of decisions rather than code
How did we come so far so fast?
They used Docker
Wrapping up
TL;DR

Docker is important for DevOps because it tightens iteration loops

The core project developed very fast, but the ecosystem is now moving faster

They did it with Docker
Questions?

Chicago, US
ContactMe@cohesiveft.com
📞 +1 888 444 3962