Docker Intro

Radu Ciorba

Spyhce

radu.ciorba@spyhce.com

May 25, 2016

Context

- started playing arround with docker in 2013
- 2014 for dev stuff + to deploy a ruby app at HackTM
- 2015 using it in automated tests + wrote a trivial init system

What is Docker?

- Container runtime.
- A toolchain to package applications and distribute them.

What's a container

- Looks like a lightweight VM (if you squint hard enought)
- Containers: Kernel level isolation (PID, users, networking, ...)
- FreeBSD Jails 2000, Solaris Zones 2004, LXC 2008

Why a new tool when we have LXC?

LXC wants to solve the problem of lightweight VMs.
Docker wants to solve the problem of application packaging and deployment.

The docker way

- Don't treat it like a VM but an Application Container.
- Only run one "App" per container.
- Have that app log to stdout, so Docker can collect the logs.

Live Demo

Persisting Data

- Containers are ephemeral.
- You can have 'VOLUMES' for a container.
- Use a dedicated DATA container and the –volumes-from flag
- Or just bind mount a volume - very useful in development

Networking

- By default docker creates a docker0 bridge and you can map ports
- You can use another container's network
- You can use the host's network

What's nice about Docker

- Building versioned images and sharing them!
- Imagine you build a version, deploy it to QA, test it there, take that exact environment (code, os, libraries) and put it in PROD.
- Imagine you could ensure your dev environment is EXACTLY the same as prod?
- Once an image is built, you don't care what's in there, you just run it.

When developing

- Bind mount your code in to an environment
- Edit locally in your env of choice
- Run the app/tests in your target env

For CI

- Run each test in it's own environment
- No more "My selenium tests require FFox X.Y but the build slaves have X.Z"

In your tests

- Docker has simple rest api
- SetUp/TearDown containers as fixtures

Docker vs Ansible/Chef/Salt/Puppet

- Docker is focused on app packaging and distribution.
- Use Ansible to prepare the nodes, pull and start your containers.

Or the big boys

- Kubernetes
- CoreOS
- Docker Swarm

Security

- Isolation is not perfect!
- Don't just run everything as root!
- Privileged containers... the name says it all.
- Running untrusted images published on the docker hub is like downloading shareware in the 90s.