CONTINUOUS INTEGRATION, DELIVERY, AND DEPLOYMENT WITH DOCKER AND DC/OS

Velocity 2016, Amsterdam, Day 1
http://301.sh/velocity-2016-ams-day1
A BIT ABOUT MYSELF …

Background
- Mesosphere
- MapR
- Research & development

What?
- DC/OS
- Docker, Kubernetes
- Serverless
- Big & fast data

github.com/mhausenblas
twitter.com/mhausenblas
mhausenblas.info
YOUR GOALS

- Building CI/CD pipelines with Docker
- Best practices of using Docker
- Efficient use of Docker
- How to properly use Docker
- Security
- At scale
- In production/bootstrap
- Team roles and responsibilities
- Basic understanding
- Distributed apps/services
• Quick poll: which OS are you using?

• Docker installed? Git installed? GitHub account?

• Docker Hub account? (or alternative registry)

• DAY 1: container bootcamp, all about manual build & deploy

• DAY 2: container operations, all about automated CI/CD

• Get https://github.com/mesosphere/training/
MOTIVATION
SEE THAT SHIT, KIDS?

THAT SHIT’S THE FUCKING FUTURE.

CONTAINER CAMP
containers are not VMs

containers are

lightweight (startup time, runtime, footprint)

for app-level dependency management

process groups on steroids
TERMINOLOGY

RUNTIME
The Docker engine runs on Linux machines, manages containers images.

IMAGE
A container image is a set of file system layers that can be shipped around as a unit and is used as a template for creating containers.

REPOSITORY
A docker image repository is a service that manages and store docker images and image layers.
CONTAINER 101

DOCKER

https://articles.microservices.com-containers-are-a-lie-2521afda1f81
Hands-on ...

• resource isolation
• resource consumption throttling
• resource consumption accounting
• images/copy-on-write filesystem(s)
namespaces for resource isolation

CLONE_NEWNS (Linux 2.4.19) filesystem mount points

CLONE_NEWUTS (Linux 2.6.19) hostname & NIS domain name

CLONE_NEWIPC (Linux 2.6.19) interprocess communication

CLONE_NEWPID (Linux 2.6.24) process ID space

CLONE_NEWNET (Linux 2.6.29) network devices, routing tables, etc.

CLONE_NEWUSER (Linux 3.8) user & group ID space
cgroups for resource consumption throttling & accounting

CONFIG_CGROUP_SCHED (Linux 2.6.24)
CONFIG_CGROUP_CPUACCT (Linux 2.6.24)
CONFIG_CPUESETS (Linux 2.6.24)
CONFIG_MEMCG (Linux 2.6.25)
CONFIG_CGROUP_DEVICE (Linux 2.6.26)
CONFIG_CGROUP_FREEZER (Linux 2.6.28)
CONFIG_CGROUP_NET_CLASSID (Linux 2.6.29)
CONFIG_BLK_CGROUP (Linux 2.6.33)
CONFIG_CGROUP_PERF (Linux 2.6.39)
CONFIG_CGROUP_NET_PRIO (Linux 3.3)
CONFIG_CGROUP_HUGETLB (Linux 3.5)
CONFIG_CGROUP_PIDS (Linux 4.3)
CoW (copy-on-write filesystems)

OverlayFS

AUFS

btrfs

ZFS

etc.

containerz.info
CONTAINER 101

IMAGES

docs.docker.com/engine/userguide/storagedriver/imagesandcontainers/
Hands-on ...

GOOD PRACTICES AND ANTPATTERNS

FROM NOW ON, WE WILL REFER TO ALL OF OUR PROBLEMS AS OPPORTUNITIES.

ONE OF YOUR IDIOT SPAWN WAS PLAYING WITH THE OVEN AND BURNED DOWN YOUR HOUSE.

CAMPING OPPORTUNITY?
- Docker Hub
  https://hub.docker.com/
- Google Cloud
  https://cloud.google.com/tools/container-registry/
- AWS
  https://aws.amazon.com/ecr/
- CoreOS
  https://quay.io/
- SUSE Portus
  http://port.us.org/
- JFrog Artifactory
  https://www.jfrog.com/artifactory/
- Run your own
  https://docs.docker.com/registry/deploying/
Rather than baking credentials into images …

$ docker run -d -e API_TOKEN=SECRET somedatabase:1.0

$ docker run -d -v $(pwd):/fsecret:/fsecret:ro somedatabase:1.0

Even better: use key-value in-memory stores such as Square's KeyWhiz, HashiCorp's Vault, or Crypt
- Follow the basic **Dockerfile best practices**
- Security/auditing
  - [https://github.com/docker/docker-bench-security](https://github.com/docker/docker-bench-security)
  - [https://github.com/coreos/clair](https://github.com/coreos/clair)
- Think of footprint (base image)
- Use labels! [http://label-schema.org](http://label-schema.org)
- One process per container (if possible)
- Use dedicated user (not root)

RUN groupadd -r myservice && useradd -r -g myservice myservice
● DON’T put a monolith in a container
● DON’T log in the container (ship it)
● DON’T use/depend on :latest tag
Hands-on …

https://github.com/mesosphere/training/blob/master/velocity-training-11-2016/DAY1/container-103.md
(TIME-PERMITTING)  
ADVANCED  
CONTAINER  
STUFF
EXAMPLES

- Take an existing app from your organisation and containerize it
- Create a new containerized microservice from scratch
- Have a look at MARVIN and try to hack it
- Try breaking out of a container and take over control of the host
- registry, labels (see also this tool)
• log shipping (slides will be incl. in DAY 2) manually
  http://www.fluentd.org/guides/recipes/docker-logging
  https://technologyconversations.com/2015/05/18/centralized-system-and-docker-logging-with-elk-stack/

• roles and responsibilities
Hands-on ...
Q & A

• @mhausenblas

• mhausenblas.info

• michael@dcos.io