Introducing Huxley

Using Docker, CoreOS, and Git Hooks to Deploy Applications
Dan
Out sick :(

Lance
Docker

Linux Containers
Skeptical?
Cloud Computing
Skeptical?
99 Problems
2014 AWS Errors By Service (Top 10)

- Route 53: 24
- CloudWatch: 20
- CloudFront: 20
- ELB: 16
- RDS: 12
- Mechanical Turk: 10
- ElastiCache: 9
- Management Console: 8

Source: AWS Service Health Dashboard, 2014
But we adapt...
KEEP CALM
AND
DO YOUR JOB
Fucking Shell Scripts
Puppet/Chef
Ansible?
Docker!
No one wants that late night phone call.
Four Nines  Five Nines
Fast!  Auto-scale
Web App
“We’re big in Japan...”
What if...
We had a cluster of Docker hosts...
“We’ve already got one…”
Okay, now we need a resource broker...
Maybe we can just write one…
“We’ve already got one...”
What about private DNS for Service Discovery…?
“We’ve already got one...”
SkyDNS
SkyDNS
AWS Route53
Private DNS
CoreOS
SkyDNS
Putting it all together...
1. Define a service
2. Request deployment
3. Deploy service
4. Register DNS Endpoints
Our New Deployment Process!

1. Set up a cluster
2. Set up fleet endpoint
3. Set up a git hook server
4. Write and test Dockerfiles (and .service files).
5. Clone the bare repo to the hook server
6. Write a script—
“How is this easier, again?”
“…the three great virtues of a programmer: laziness, impatience, and hubris…”

–Larry Wall, Programming Perl
panda-cluster
panda-hook
panda-kick
These are MIT-licensed open source projects.

You can *totally* use them.
panda-cluster

- Automate the management of clusters
- Resource allocation via fleet
- Deployment via git hook server
panda-hook

- Clone a local repo to the cluster
- Subsequent pushes trigger deploys
- Deploys everything in launch directory
panda-kick

- Internal HTTP service providing authenticated access to external services
- Ex: service announcements via private DNS
- Only services on the cluster can do this
Our New Deployment Process!

1. Set up a cluster
2. Set up fleet endpoint
3. Set up a git hook server
4. Write and test Dockerfiles (and .service files)
5. Clone the bare repo to the hook server
6. Write a script—
git hook server
| git hook server | fleet endpoint |
git hook server
fleet endpoint
kick server
git push staging
fleetctl start ...
curl kick:2000 -XPOST ...
Voila! Push-based deployments!
Self-Describing Deployments
my-project/
  bin/
  launch/
  src/
  test/
my-project/
  bin/
  → launch/
  src/
  test/
Contains configuration and initialization files

```
my-project/
  bin/
  → launch/
  src/
  test/
```
my-project/
bin/
launch/
web/
    Dockerfile
    web.service
db/
    Dockerfile
    db.service
worker/
    Dockerfile
    worker.service
src/
test/
FROM base/archlinux
MAINTAINER David Harper (david@pandastrike.com)

# This Dockerfile describes a Vanilla web server.

RUN pacman -Syu --noconfirm
RUN pacman-db-upgrade
RUN pacman -S --noconfirm jre7-openjdk-headless wget vim tmux git nodejs

# Install Vanilla
#RUN git clone ssh://root@vanilla.pandastrike.com:3000/repos/orwell.git vanilla
RUN git clone https://github.com/PandaPup/vanilla.git vanilla

# Install CoffeeScript
RUN npm install -g coffee-script
# Vanila - Service File

[Unit]
Description=Spin Up Vanilla
After=docker.service

[Service]
TimeoutStartSec=0

ExecStartPre=/usr/bin/echo "=================================
ExecStartPre=/usr/bin/echo " New Service Starting"
ExecStartPre=/usr/bin/echo "=================================

# Display this service's IP addresses.
EnvironmentFile=/etc/environment
ExecStartPre=/usr/bin/echo "Public IP Address: ${COREOS_PUBLIC_IPV4}"
ExecStartPre=/usr/bin/echo "Private IP Address: ${COREOS_PRIVATE_IPV4}"

# Pull The Service's Docker Container.
ExecStartPre=/usr/bin/docker build --tag="vanilla_image" /home/core/launch/vanilla/.

ExecStart=/usr/bin/docker run --name vanilla -p 3001:80 \
    vanilla_image /bin/bash -c \
    'cd vanilla && coffee vanilla.coffee'

[Install]
WantedBy=multi-user.target
README.txt  INSTALL.txt  Dockerfile
my.service
git push staging master
So far, so good...
Auto-scaling
Better documentation
Lots of moving parts...
Awesome Command-Line Tool and Library to Manage CoreO

using latest pirate master

Final Preps for Prototype Release
Removed "app" stuff. "cluster" cmd
ported over pbx and pbx-cli
renamed package.json
Update create.json
ported over pbx and pbx-cli
Private DNS Established, No Anno
Initial commit
Added Schema For Create()
renamed api-guide.md and unit-guide.md
MOAR PANDA!
Huxley
panda-cluster
panda-hook
panda-kick
HTTP API

panda-cluster
panda-hook
panda-kick
CLI

HTTP API

panda-cluster
panda-hook
panda-kick

huxley create cluster staging
huxley init
huxley remote add staging
huxley mixins add node ha-redis
Mixins are predefined launch configurations
my-project/
  bin/
  launch/
    web/
      Dockerfile
      web.service
  src/
  test/
huxley mixin add ha-redis
my-project/
  bin/
  launch/
  web/
      Dockerfile
      web.service
      ha-redis/
          Dockerfile
          redis.@service
          config.yml
  src/
  test/
url: http://my-app.acme.com
web:
  - web-0.local
  - web-1.local
redis:
  - redis-0.local
  - redis-1.local
url: http://my-app.acme.com

web:
- web-0.local
- web-1.local

redis:
- redis-0.local
- redis-1.local
Linux containers: disruptive
Automate container deployment onto clusters.
Self-Describing Deployments
Mixins

my-project/
  bin/
  launch/
  web/
    Dockerfile
    web.service
    ha-redis/
      Dockerfile
      redis.@service
      config.yaml

crc/

test/
managing clusters
container deployment
writing mixins
Status

panda-cluster
panda-hook
panda-kick
huxley
huxley-as-a-service
Status

panda-cluster
panda-hook
panda-kick
huxley
huxley-as-a-service
Status

panda-cluster
panda-hook
panda-kick
huxley
huxley-as-a-service
Status

panda-cluster
panda-hook
panda-kick
huxley
huxley-as-a-service
Status

panda-cluster
panda-hook
panda-kick
huxley
huxley-as-a-service

alpha
alpha
alpha
development
beta in March
Open Source/MIT License

https://github.com/pandastrike/

panda-cluster
panda-hook
panda-kick
huxley
Huxley-as-a-service
https://www.pandastrike.com/huxley
Huxley-as-a-service
https://www.pandastrike.com/huxley

Join the beta!
Thank you.

dan@pandastrike.com