TAMING A DINOSAUR WITH JAVASCRIPT AND DOCKER

Linux Users’ Group of Davis

Dennis Stevense
Preparing the Next Generation of Business Leaders

...whose entrepreneurship and innovation are impacting the world
The UC Davis School of Education is transforming education through preparation of the best education leaders, researchers, and teachers in the country. We are committed to eliminating inequities in schooling and creating learning opportunities for diverse learners using the power of knowledge and the promise of education.
24% of children live in poverty. We're changing that.

Sacramento County
all relationships, all genders, all ages, all people, all the time
What we do

- We empower communication and connect communities
  - “5-phase process”
  - “Standard Model,” our own Drupal 6 flavor/distribution
- We host about 200 sites
- We update all sites monthly as part of our SLA
Our hosting stack

- “LEMP stack”
  - CentOS
  - Varnish
  - NGINX
  - PHP-FPM
  - MySQL (Percona)
- Drupal 6
THE PROBLEM
Taming a Drupal 6 codebase

https://www.youtube.com/watch?v=_dcojg8MFPo
Digital Deployment “Standard Model”

- Our Drupal 6 flavor/distribution
  - Drupal 6 released in 2008
  - Drupal 6 EOL in 2016
  - Only minimal unit tests
- Used to run about 200 sites
  - All sites updated monthly as part of our SLA
  - Each site has a unique theme
Updating 200 sites from Drupal 6 to 8

- Commitment to quality and client confidence
- Digital Deployment-specific features
  - Functional regression testing
- Site-specific themes
  - Visual regression testing
Testing requirements

- Fast test resets
  - Because test independence
- Consistent screenshots
  - Because visual regression testing is key
- Framework and stack agnostic
  - Because we’re upgrading the framework and stack
- Continuous integration
  - Because humans
OUR SOLUTION

JavaScript and Docker to the rescue

https://www.youtube.com/watch?v=AHAVptS87rs
Operating-system-level virtualization

- Virtualize without running a separate kernel
- File system isolation
- Copy-on-write
- System resource limits
- Network isolation
- ...

Docker

libcontainer

libvirt

lxc

systemd-nspawn

Linux

cgroups

namespaces

netlink

selinux

netfilter

capabilities

apparmor

https://blog.docker.com/2014/03/docker-0-9-introducing-execution-drivers-and-libcontainer/
Docker

- Docker **container**
  - Light-weight equivalent of a virtual machine

- Docker **image**
  - File system snapshot to start new containers
  - Built from a **Dockerfile**
  - Pulled from Docker Hub
Docker Compose

- Define multi-container Docker application
  - docker-compose.yml
Fast test resets

OUR SOLUTION

- Build Docker images for all the things
  - Site files
  - MySQL w/ site database preloaded
  - NGINX/PHP-FPM
- Run with Docker Compose
- Re-create Docker containers for every test
FROM alpine:3.4

RUN apk add --update curl build-base pcre pcre-dev zlib zlib-dev openssl openssl-dev && \
  cd /root && \
  curl -L http://nginx.org/download/nginx-1.8.0.tar.gz | tar -xz && \
  cd nginx-* && \
  ./configure \
    --prefix=/etc/nginx \
    --sbin-path=/usr/sbin/nginx \
    --conf-path=/etc/nginx/nginx.conf \
    --pid-path=/var/run/nginx.pid \
    --error-log-path=/var/log/nginx/error.log \
    --http-log-path=/var/log/nginx/access.log && \
  make && \
  make install && \
  cd .. && \
  rm -rf nginx-* && \
  apk del curl build-base pcre-dev zlib-dev openssl-dev && \
  rm -rf /var/cache/apk/*

RUN ln -sf /dev/stdout /var/log/nginx/access.log && \
    ln -sf /dev/stderr /var/log/nginx/error.log

COPY nginx.conf /etc/nginx/

EXPOSE 80 81

CMD ["nginx", ":g", ":daemon off;"]
version: '2'

services:
  # This image has the site data in /var/data, and we create a data-only
  # container from it that we can throw away to reset quickly.
  data:
    image: $DATA_IMAGE
    volumes:
      - /var/data/files
      - /var/data/modules
      - /var/data/themes
  # This image has MySQL with the site database pre-loaded, so we can reset
  # quickly by re-creating the container.
  db:
    image: $DB_IMAGE
  php:
    image: 597943232345.dkr.ecr.us-west-2.amazonaws.com/model/php:lightcrest-20161115
    environment:
      - XHPROF_ENABLE
    volumes:
      - /:/var/www:ro
      - /docker/sites-default:/var/www/sites/default:ro
    volumes_from:
      - data
    links:
      - db
  nginx:
    image: 597943232345.dkr.ecr.us-west-2.amazonaws.com/model/nginx:lightcrest-20161115
    volumes:
      - /:/var/www:ro
      - /docker/sites-default:/var/www/sites/default:ro
    volumes_from:
      - data
    links:
Consistent screenshots

- Selenium Docker images
  - https://hub.docker.com/r/selenium/
  - Includes Selenium Server
  - Includes browser
  - Consistent font rendering, form fields, etc.
Framework and stack agnostic

- Node.js / JavaScript
- Mocha test framework
  - `$ npm test`
- Control arbitrary Docker containers
  - `child_process`
  - `stream`
- Selenium
  - `selenium-webdriver`
Continuous integration

- **CircleCI** automatically tests subset of sites every time we push to **GitHub**
- has none for robust-webform (46.57s)
- has none for classes-library (43.90s)
- has none for embed-images (43.11s)
- has none for headings-waterfall (23.87s)
- has none for accordions (20.00s)
- has none for photo-gallery (31.55s)
- has none for file-attachments (23.14s)
- has none for various-rtl-elements (24.93s)
- has none for directory (31.62s)
- has none for search-results (29.95s)

helenorine
post editing
- lets me create a new post (30.736ms)
- lets me do other stuff (18.651ms)

visual regression
- has none for homepage (11.800ms)

ccahn
post editing
- lets me create a new post (24.815ms)
- lets me do other stuff (6.704ms)

visual regression
- has none for homepage (8.873ms)
- has none for ca-state-resources (33.17ms)

reeh
post editing
- lets me create a new post (29.807ms)
- lets me do other stuff (8.740ms)

visual regression
- has none for homepage (18.937ms)
- has none for transparency-and-public-information (29.44ms)
- has none for performance (27.814ms)
- has none for purchase-service-data (29.74ms)

35 passing (11ms)
HTTP-based microservice for taking screenshots of sites using Selenium. — Edit

<table>
<thead>
<tr>
<th>File</th>
<th>Action</th>
<th>Status</th>
<th>Last updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>.gitignore</td>
<td>Initial commit</td>
<td></td>
<td>8 hours ago</td>
</tr>
<tr>
<td>Dockerfile</td>
<td>Initial commit</td>
<td></td>
<td>8 hours ago</td>
</tr>
<tr>
<td>LICENSE</td>
<td>Create LICENSE</td>
<td></td>
<td>8 hours ago</td>
</tr>
<tr>
<td>README.md</td>
<td>Update README.md</td>
<td></td>
<td>an hour ago</td>
</tr>
<tr>
<td>app.js</td>
<td>Initial commit</td>
<td></td>
<td>8 hours ago</td>
</tr>
<tr>
<td>docker-compose.yml</td>
<td>Initial commit</td>
<td></td>
<td>8 hours ago</td>
</tr>
<tr>
<td>package.json</td>
<td>Initial commit</td>
<td></td>
<td>8 hours ago</td>
</tr>
</tbody>
</table>

**camera**

This is a simple webservice for taking screenshots of sites using a real browser using Selenium. It’s meant to be a building block for setting up your own visual regression testing. Features:

- Automatically restarts the browser when necessary for basic error recovery.
- Takes screenshots by hiding scrollbars, scaling images consistently using nearest neighbor, and taking into account browser chrome/UI dimensions.
- Attempts to resize the browser window height to fit the webpage.

Currently it’s provided as a Docker image that depends on the Selenium Docker image, but in the future we’d like to extract some of the browser control logic into a node package so it can more easily be incorporated in existing JavaScript-based test suites.

**Getting started**
git clone https://github.com/digitaldeployment/camera.git
cd camera/
docker-compose up -d