Inside Yelp's SOA Infrastructure

Or: How to evaluate the ecosystem when considering PaaS’s.
Who Am I:

- Kyle Anderson
- Operations Team at Yelp
- Trying **not** to build PaaS’s since forever
Yelp’s Mission:
Connecting people with great local businesses.
Yelp Stats:
As of Q3 2015

89M
79M
71%
32
What Is the Purpose of this Talk?

A. **Inform** you of Yelp’s SOA infrastructure
B. **Persuade** you to use Yelp’s Code
C. **Inspire** you to take ideas from Yelp’s SOA journey
D. **Promote** Mesos/Docker/Kubernetes/etc
Let’s Start From Scratch

Dev

@hipsterhacker
@ExpertBeginner1
@imadveloper

Ops

@sadoperator
@sadserver
Let’s Ship Some Code

Devs

@hipster
hacker

@Expert
Beginner

@imad
veloper

Ops

@sadoperator

@sadserver
What is the problem then, exactly?

- **Empowering** developers to ship code faster is a competitive advantage
  - Somehow this didn’t used to be the case?
- This whole “devops” thing seems to hint that Ops should help Devs get their code out
  - This implies some sort of **tooling**
- **Automation** is the key of course
  - Which leads to needing some sort “PaaS”
Ok Fine. What Does This Have to Do With SOA? (microservices?)

I wonder how many organizations that say they're "doing DevOps" are actually building a bespoke PaaS. And how many of those realize it.
As an Operations guy, my primary responsibility is empowering developers to get their code running healthily in production.

Use whatever words you want to describe that. (devops, PaaS, bespoke, etc)

- Kyle Anderson
Face it: Everyone Builds A Bespoke PaaS to Some Degree

- How *Bespoke* does it have to be?
- How much open-source technology can you *reuse*?
- Should you *buy* instead of *build*?
Options:

- There is no shortage of tools out there
- How do you figure out what you want to use, if anything?

What the hell have you built.

- Did you just pick things at random?
- Why is Redis talking to MongoDB?
- Why do you even use MongoDB?

Goddamnit  Nevermind  FREE
More Help:

Talk: Beyond ad-hoc automation: to structured platforms

Speaker: Bridget Kromhout

Track: Containers in Practice

Location: Ballroom A

Duration: 2:55pm - 3:45pm
Then what did Yelp do?

1. **Survey** what you have
2. Map out what you **want**
3. Write as **little** code as possible!!
1. Survey What You Have

This was not Yelp’s first SOA rodeo. We already were using these components:

- Airbnb’s SmartStack
  Nerve / Synapse
  (service discovery)

- sensu
  (monitoring)

- docker
  (testing in dev only)

- (CI/CD Pipelines)
2. Map Out What You Want

- **Declarative** config files that define how each service is deployed (*soa-configs*)
- **Automatic** resource scheduling and contained services (*cgroups/docker*)
- **Consistent** environments (*containers*)
- **Visibility** into their deploys and control their workflow (*pipelines*)
3. Write As Little Code As Possible
Stage: Acceptance
We are going to deploy some sort of PaaS

Even if you are going to use the most full-featured turn-key PaaS, you are still going to end up adding **SOME** modifications on top.
Warning:

Opinions Ahead
Choosing Principles

PaaSTA Principles

These are a list of the principles that we think make PaaSTA special, and also opinionated. If you don’t share these opinions, then PaaSTA is probably not for you.

This document is similar, but not exactly the same as the 12factor site for Heroku. The principles behind the infrastructure do influence how the apps are deployed. The technical document for the exact contract an app must meet to run on PaaSTA is documented in the PaaSTA Contract.

Principles

1. **Declarative** is better than **imperative**

There is a subtle difference between these two approaches to configuring a particular app in a theoretical PaaS:

<table>
<thead>
<tr>
<th>Declarative</th>
<th>Imperative</th>
</tr>
</thead>
<tbody>
<tr>
<td>* each operation is done once</td>
<td>* sequence of inputs</td>
</tr>
<tr>
<td>* sequence of inputs</td>
<td>* sequence of commands</td>
</tr>
<tr>
<td>* sequence of commands</td>
<td>* sequence of commands</td>
</tr>
<tr>
<td>* sequence of commands</td>
<td>* sequence of commands</td>
</tr>
</tbody>
</table>
Building Your (Yelp’s) Dream PaaS:
SOA-Configs

When you accept you are going to build your own PaaS, you get to pick how you want it to work.

We wanted a git repo of service definitions. We called it “soa-configs”
Building Your (Yelp’s) Dream PaaS: SOA-Configs
SOA-Configs: A great foundation for a SOA

- **Declarative** control for how your services are defined
- **Git** for rollbacks, audits, access control, code review
- Yaml files are pretty **easy** to use
- **Easy distribution** (not a database)
Creating A Service

kwa@dev13-devc:~$ paasta fsm --yelpsoa-config-root ~/Projects/yelpsoa-configs -
team operations
Service name? qcon
Smartstack proxy_port? 20994
One line description of this service? "A service for Qcon"
Link to a reference doc? http://qconsf.com

With My Noodly Appendage I Have Written Configs For

qcon

Customize Them If It Makes You Happy -- http://y/paasta For Details
Remember To Add, Commit, And Push When You're Done:
How Do You Do Service Discovery in a Dynamic World?

- Yelp uses Airbnb’s SmartStack for service discovery
- Works on Mesos, Puppet, Custom SOA, etc.
- Not tied to a particular PaaS
How Do You Do Service Discovery in a Dynamic World?

docker container 1

- lo 127.0.0.1
- eth0 169.254.14.17

docker container 2

- lo 127.0.0.1
- eth0 169.254.14.18

haproxy

- docker0 169.254.1.1
- lo 127.0.0.1
- lo:0 169.254.255.254
- eth0 10.0.1.2
What Next? You need a scheduler

- Humans just don’t cut it for scheduling things on boxes
- Yelp wanted a production scheduler that was active and could contain things (docker support)
- The answer for us was Apache Mesos
What is Mesos?

- If your datacenters were an operating system, Mesos would be the kernel.
- Knows about resources, schedulers, agents, and pools
What is Mesos?

<table>
<thead>
<tr>
<th>Active Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
</tr>
<tr>
<td>gitlab-ci-11-960d746df29eda902a7512be886d6a9e164c7e8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Completed Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
</tr>
<tr>
<td>gitlab-ci-10-22d22647256638995896388a84de9e7419d58c3</td>
</tr>
</tbody>
</table>

Cluster: thunderstorm
Server: 147.32.232.88:5050
Built: 3 weeks ago by root
Started: yesterday

LOG
Slaves
Activated: 6
Deactivated: 0
Tasks
Staged: 22
Started: 0
Finished: 10
Killed: 0
Failed: 2
Lost: 0
Resources
CPUs          Mem
Total: 48      60.5 GB
Used: 1        563 MB
Offered: 0      0 B
Idle: 47       59.9 GB
Why did Yelp Pick Mesos?

- We are in it for the long haul.
- Mesos has seen large-scale production use at Twitter since 2010.
- Infrastructure agnostic (not AWS-specific)
- Mesos has an upgrade path!

http://mesos.apache.org/documentation/latest/upgrades/
Why Did Yelp Pick Marathon?

- Marathon is a framework (think plugin) for Mesos that supervises apps to make sure they are healthy (like upstart or supervisord).
- Yelp picked Marathon because it was **un-opinionated** and did one thing well.
Why Did Yelp Pick Marathon?
It’s Not Done Till It’s Shipped Monitored

- We use Sensu to monitor our services and alert authors when they are not healthy
- Services are monitored by default (authors have to opt out)
It’s Not Done Till It’s Shipped
Service Status

```
@dev13-devc:~$ paasta status -s engineering_blog -c norcal-prod,nova-prod
Pipeline: https://jenkins.yelpcorp.com/view/services-engineering_blog

cluster: norcal-prod
  instance: main
    Git sha: 56283688
    State: Running - Desired state: Started
    Marathon: Healthy - up with (3/3) instances. Status: Running.
    Mesos: Healthy - (3/3) tasks in the TASK_RUNNING state.
    Smartstack:
      Name  LastCheck  LastChange  Status
      norcal-prod - Healthy - in haproxy with (3/3) total backends UP in this namespace.

cluster: nova-prod
  instance: main
    Git sha: 56283688
    State: Running - Desired state: Started
    Marathon: Healthy - up with (3/3) instances. Status: Running.
    Mesos: Healthy - (3/3) tasks in the TASK_RUNNING state.
    Smartstack:
      Name  LastCheck  LastChange  Status
      nova-prod - Healthy - in haproxy with (3/3) total backends UP in this namespace.
```
The Result: PaaSTA

- PaaSTA is the name of “glue” of all these pieces into a coherent set of tooling for developers.
- Enforces these opinions about how things should be deployed
- Provides tools for inspecting and deploying services
Deploying A Service

Build Pipeline

- Pipeline #65
  - #65 services-devops
  - Nov 16, 2015 1:56:06 PM
  - 1 min 9 sec
- Pipeline #64
  - #64 services-devops
  - Nov 15, 2015 1:59:08 PM
  - 1 min 13 sec
- Pipeline #63
  - #63 services-devops
  - Nov 15, 2015 1:56:05 PM
  - 1 min 9 sec
- Pipeline #62
  - #62 services-devops
  - Nov 13, 2015 1:56:06 PM
  - 1 min 9 sec
- Pipeline #61
  - #61 services-devops
  - Nov 14, 2015 1:56:03 PM
  - 1 min 5 sec
- Pipeline #60
  - #60 services-devops
  - Nov 14, 2015 1:56:03 PM
  - 1 min 5 sec
- Pipeline #59
  - #59 services-devops
  - Nov 13, 2015 1:56:03 PM
  - 1 min 6 sec
- Pipeline #58
  - #58 services-devops
  - Nov 13, 2015 1:56:03 PM
  - 1 min 6 sec
- Pipeline #57
  - #57 services-devops
  - Nov 16, 2015 2:06:14 PM
  - 39 sec
- Pipeline #56
  - #56 services-devops
  - Nov 15, 2015 2:06:20 PM
  - 38 sec
- Pipeline #55
  - #55 services-devops
  - Nov 15, 2015 2:06:53 PM
  - 38 sec
- Pipeline #54
  - #54 services-devops
  - Nov 12, 2015 2:06:18 PM
  - 39 sec
- Pipeline #53
  - #53 services-devops
  - Nov 12, 2015 2:06:17 PM
  - 38 sec
- Pipeline #52
  - #52 services-devops
  - Nov 12, 2015 2:06:52 PM
  - 38 sec
Service Status

<table>
<thead>
<tr>
<th>cluster: norcal-prod</th>
<th>instance: main</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Git sha: 56283688</td>
</tr>
<tr>
<td></td>
<td>State: Running - Desired state: Started</td>
</tr>
<tr>
<td></td>
<td>Marathon: Healthy - up with (3/3) instances. Status: Running.</td>
</tr>
<tr>
<td></td>
<td>Mesos: Healthy - (3/3) tasks in the TASK_RUNNING state.</td>
</tr>
<tr>
<td></td>
<td>Smartstack:</td>
</tr>
<tr>
<td></td>
<td>Name   LastCheck  LastChange  Status</td>
</tr>
<tr>
<td>norcal-prod</td>
<td>Healthy in haproxy with (3/3) total backends UP in this namespace.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>cluster: nova-prod</th>
<th>instance: main</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Git sha: 56283688</td>
</tr>
<tr>
<td></td>
<td>State: Running - Desired state: Started</td>
</tr>
<tr>
<td></td>
<td>Marathon: Healthy - up with (3/3) instances. Status: Running.</td>
</tr>
<tr>
<td></td>
<td>Mesos: Healthy - (3/3) tasks in the TASK_RUNNING state.</td>
</tr>
<tr>
<td></td>
<td>Smartstack:</td>
</tr>
<tr>
<td></td>
<td>Name   LastCheck  LastChange  Status</td>
</tr>
<tr>
<td>nova-prod</td>
<td>Healthy in haproxy with (3/3) total backends UP in this namespace.</td>
</tr>
</tbody>
</table>
Bad Service Status

Cluster: nova-devc
Instance: demo
Git sha: 8170d ell
State: Running - Desired state: Started
Mesos: Warning - (3/6) tasks in the TASK_RUNNING state.
Smartstack:
useastladevc - Critical - in haproxy with (0/3, 0%) total backends UP in this namespace.
useast1bdevc - Healthy - in haproxy with (3/3) total backends UP in this namespace.
Conclusion

• You can build the PaaS you have always dreamed of, but you have to **know** what you want first.

• Don’t deploy a tool just because everyone else is doing it, know what **problem** you are trying to solve and be deliberate.

• The parts are out there, don’t be distracted by hype.
What Is the Purpose of this Talk?

A. **Inform** you of Yelp’s SOA infrastructure
B. **Persuade** you to use Yelp’s Code
C. **Inspire** you to take ideas from Yelp’s SOA journey
D. **Promote** Mesos/Docker/Kubernetes/etc
What Is the Purpose of this Talk?

A. **Inform** you of Yelp’s SOA infrastructure
B. **Persuade** you to use Yelp’s Code
C. **Inspire** you to take ideas from Yelp’s SOA journey
D. **Promote** Mesos/Docker/Kubernetes/etc
Share in Yelp’s Principles?

- **PaaSTA**
  [https://github.com/yelp/paasta](https://github.com/yelp/paasta)  
  *(Check out the Videos and **PaaSTA Principles**!)*
  - Sensu:  
    [https://sensuapp.org/](https://sensuapp.org/)
  - Mesos:  
  - Marathon:  
    [https://mesosphere.github.io/marathon/](https://mesosphere.github.io/marathon/)
  - Smartstack:  