

Inside Yelp's SOA Infrastructure

Or: How to evaluate the ecosystem when
considering PaaS's.



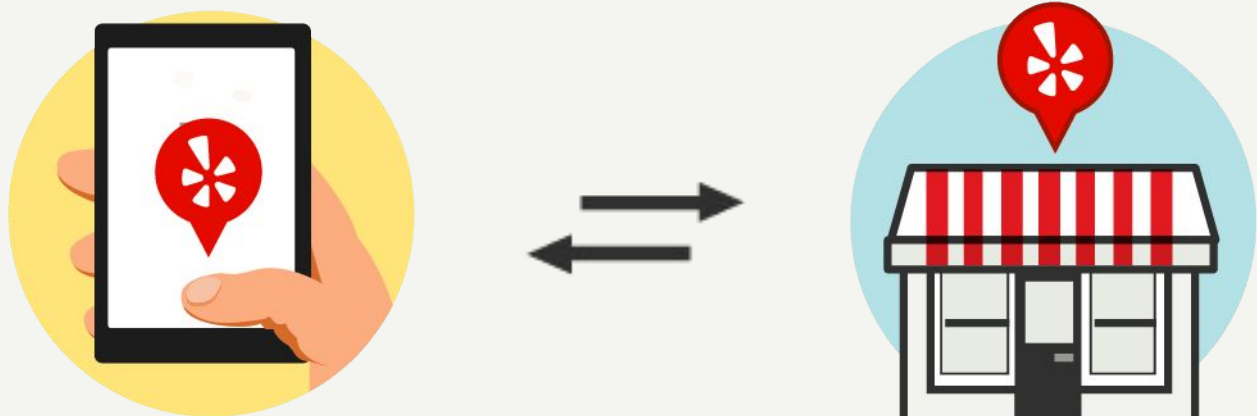
Who Am I:

- Kyle Anderson
- Operations Team at Yelp
- Trying **not** to build PaaS's since forever



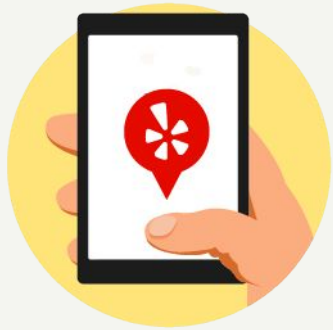
Yelp's Mission:

Connecting people with great local businesses.

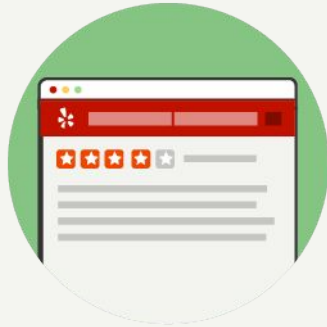


Yelp Stats:

As of Q3 2015



89M



79M



71%



32



What Is the Purpose of this Talk?

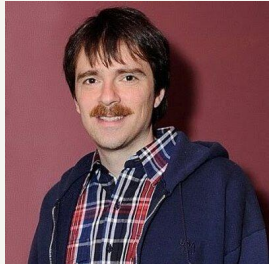
- A. **Inform** you of Yelp's SOA infrastructure
- B. **Persuade** you to use Yelp's Code
- C. **Inspire** you to take ideas from Yelp's SOA journey
- D. **Promote** Mesos/Docker/Kubernetes/etc



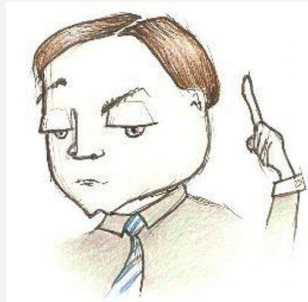
Let's Start From Scratch

Devs

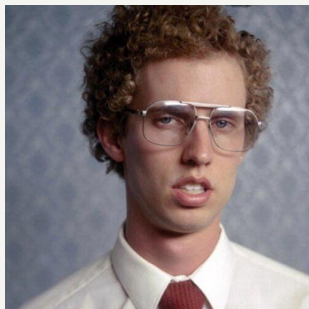
@hipsterhacker



@ExpertBeginner1

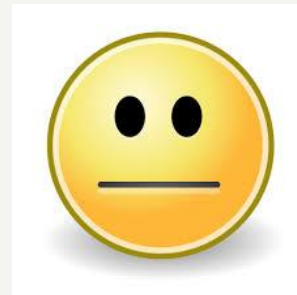


@imadveloper



Ops

@sadoperator



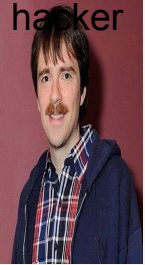
@sadserver



Let's Ship Some Code

Devs

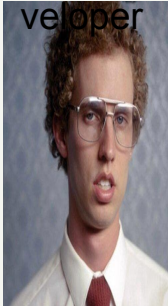
@hipster
hacker



@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?)



Mark Imbriaco
@markimbriaco



 Follow

I wonder how many organizations that say they're "doing DevOps" are actually building a bespoke PaaS. And how many of those realize it.

RETWEETS

102

LIKES

77



9:27 PM - 28 Sep 2014



What are we doing then?



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**



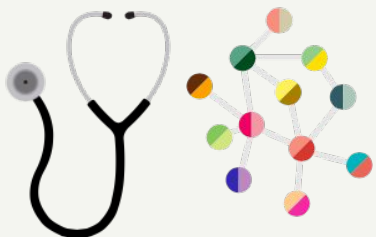
Then what did Yelp do?

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



1. Survey What You Have

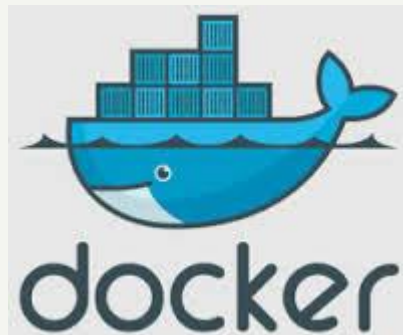
This was not Yelp's first SOA rodeo. We already were using these components:



Airbnb's **SmartStack**
Nerve / Synapse
(service discovery)



(monitoring)



(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

PaaS Principles

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

This document is similar, but not exactly the same as the [12 factor](#) 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 PaaS is documented in the [PaaS 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:

Declarative	Imperative
<pre>function deploy() { // ... }</pre>	<pre>function deploy() { // ... }</pre>

Table Of Contents

[PaaS Principles](#)

- [Principles](#)
 - [1. Declarative is better than imperative](#)
 - [2. Git is a pretty good control plane](#)
 - [3. Services should be owned and monitored](#)

This Page

[Show Source](#)

Quick search

Go

Enter search terms or a module, class or function name.

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 $ tree -t
```

```
backend
├── marathon.yaml
├── monitoring.yaml
├── service.yaml
└── chronos.yaml

web
├── service.yaml
├── marathon.yaml
├── monitoring.yaml
└── smartstack.yaml

app
├── marathon.yaml
├── monitoring.yaml
├── service.yaml
└── deploy.yaml

README.md
```

```
3 directories, 13 files
```

```
~ > soa-configs $
```

```
~ > yelpsoa-configs > devops master + $ cat monitoring.yaml
```

```
---
team: operations
page: true
notification_email: devops@yelp.com
```

```
~ > yelpsoa-configs > devops master + $ cat marathon.yaml
```

```
---
demo:
  cpus: 0.1
  instances: 6
  mem: 500
```

```
~ > yelpsoa-configs > devops master + $
```



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

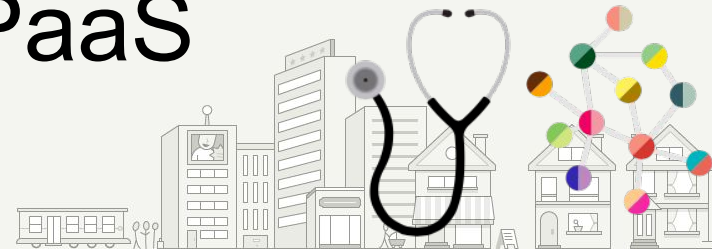
qcon

Customize Them If It Makes You Happy -- <http://y/paasta> For Details
Remember To Add, Commit, And Push When You're Done:

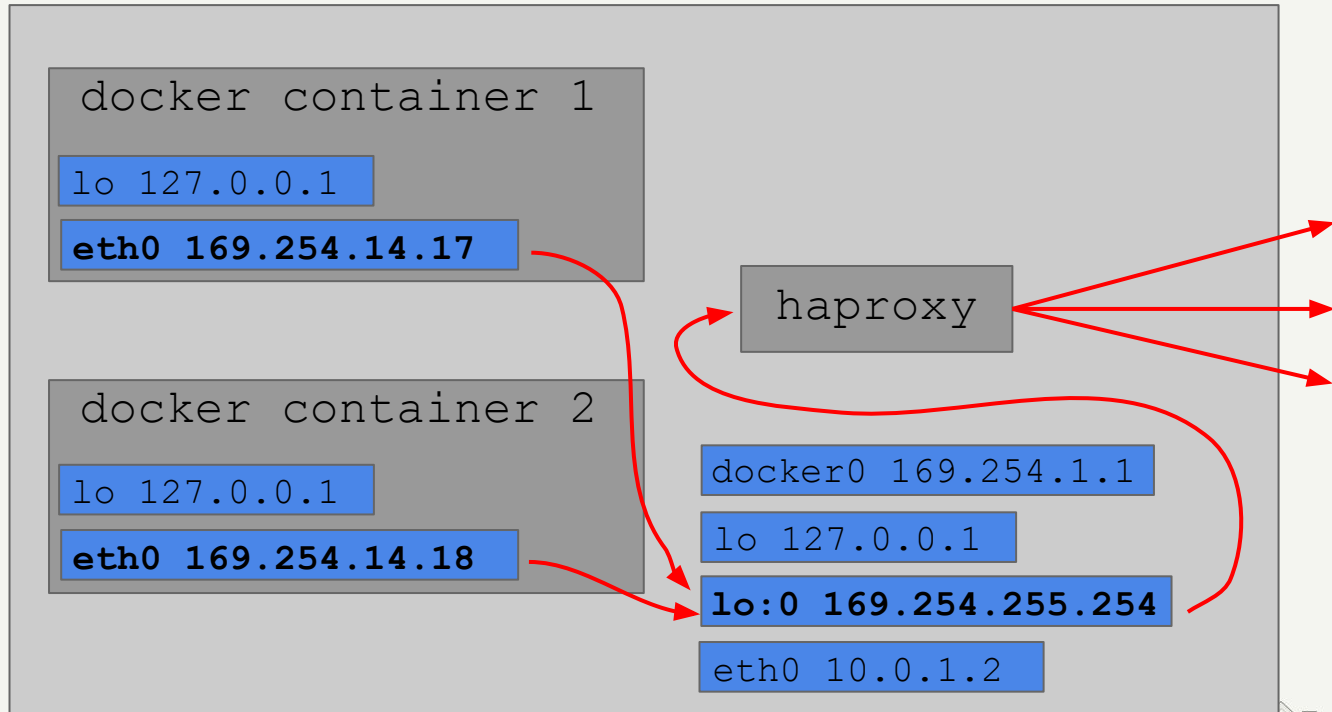


How Do You Do Service Discovery in a Dynamic World?

- 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?



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



MESOS

What is Mesos?

Mesos Frameworks Slaves Offers thunderstorm

Master 201402042334-1491607699-5050-25070

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	19
Killed	0
Failed	2
Lost	0

Resources

	CPU's	Mem
Total	48	60.5 GB
Used	1	563 MB
Offered	0	0 B
Idle	47	59.9 GB

Active Tasks


Find...

ID	Name	State	Started ▼	Host	
gitlab-ci-11-bd3df5d7a685b19a4b13642afbcc7f148dc3539d	task gitlab-ci-11-bd3df5d7a685b19a4b13642afbcc7f148dc3539d	RUNNING	44 years ago	at.fit.cvut.cz	Sandbox

Completed Tasks

Find...

ID	Name	State	Started ▼	Stopped	Host
gitlab-ci-10-d60d746df29eda902a7512be88c66ae9b164c7e8	task gitlab-ci-10-d60d746df29eda902a7512be88c66ae9b164c7e8	FINISHED	5 hours ago	5 hours ago	storm2.fit.cvut.cz
gitlab-ci-10-cc749a0a5e6941e8696a8cc66901843fdeba5e3b	task gitlab-ci-10-cc749a0a5e6941e8696a8cc66901843fdeba5e3b	FINISHED	5 hours ago	5 hours ago	storm2.fit.cvut.cz
gitlab-ci-8-e9036dab994b4f913e8cc9980ac7d0e4c5523038	task gitlab-ci-8-e9036dab994b4f913e8cc9980ac7d0e4c5523038	FINISHED	9 hours ago	9 hours ago	storm2.fit.cvut.cz
gitlab-ci-8-e9036dab994b4f913e8cc9980ac7d0e4c5523038	task gitlab-ci-8-e9036dab994b4f913e8cc9980ac7d0e4c5523038	FINISHED	44 years ago	44 years ago	at.fit.cvut.cz
gitlab-ci-10-d60d746df29eda902a7512be88c66ae9b164c7e8	task gitlab-ci-10-d60d746df29eda902a7512be88c66ae9b164c7e8	FINISHED	44 years ago	44 years ago	at.fit.cvut.cz
gitlab-ci-12-d29d22647256638995896388a84de9e7419d58c3	task gitlab-ci-12-d29d22647256638995896388a84de9e7419d58c3	FINISHED	44 years ago	44 years ago	datablab.fit.cvut.cz



Why did Yelp Pick Mesos?

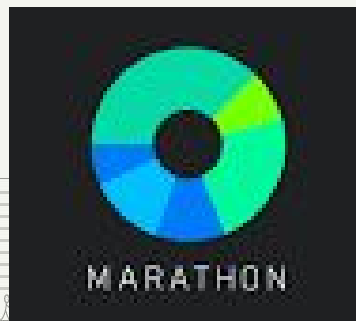
- We are in it for the long haul.
- Mesos has seen large-scale production use at Twitter since 2010.
- Infrastructure agnostic (not AWS-specific)
- **Mesos has an upgrade path!**

<http://mesos.apache.org/documentation/latest/upgrades/>

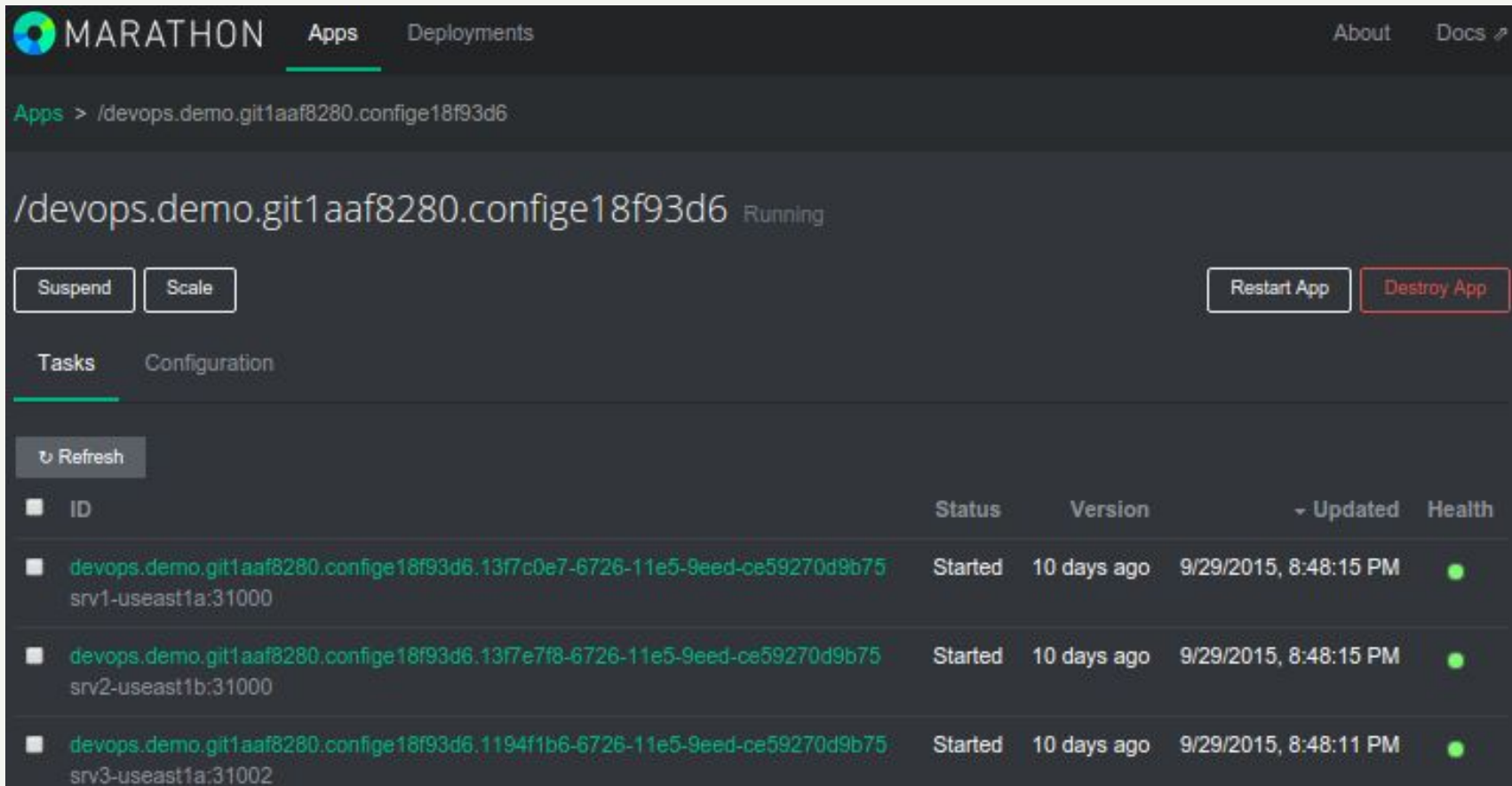


Why Did Yelp Pick Marathon?

- 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.

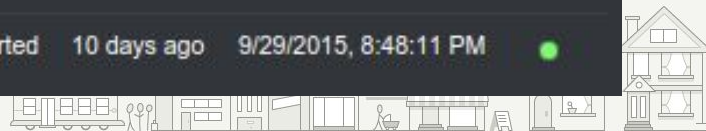


Why Did Yelp Pick Marathon?



The screenshot displays the Marathon web interface. At the top, there's a navigation bar with 'MARATHON' logo, 'Apps' (selected), 'Deployments', 'About', and 'Docs'. Below the navigation, the breadcrumb path is 'Apps > /devops.demo.git1aaf8280.config18f93d6'. The main content area shows the application path '/devops.demo.git1aaf8280.config18f93d6' in a 'Running' state. There are four action buttons: 'Suspend', 'Scale', 'Restart App', and 'Destroy App'. Below these are tabs for 'Tasks' (selected) and 'Configuration'. A 'Refresh' button is located above a table of tasks. The table has columns for 'ID', 'Status', 'Version', 'Updated', and 'Health'. Three tasks are listed, all with a 'Started' status and a green health indicator.

ID	Status	Version	Updated	Health
devops.demo.git1aaf8280.config18f93d6.13f7c0e7-6726-11e5-9eed-ce59270d9b75 srv1-useast1a:31000	Started	10 days ago	9/29/2015, 8:48:15 PM	●
devops.demo.git1aaf8280.config18f93d6.13f7e7f8-6726-11e5-9eed-ce59270d9b75 srv2-useast1b:31000	Started	10 days ago	9/29/2015, 8:48:15 PM	●
devops.demo.git1aaf8280.config18f93d6.1194f1b6-6726-11e5-9eed-ce59270d9b75 srv3-useast1a:31002	Started	10 days ago	9/29/2015, 8:48:11 PM	●



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 Monitored

The screenshot displays the Uchiwa Events monitoring interface. The page title is "Events" and it includes a search bar and a "Show 50" dropdown. The main content is a table with the following columns: Client, Check, Output, ID, and Time. The table lists numerous events, with one row highlighted in blue. The interface also features a sidebar with navigation icons and a top navigation bar with the Uchiwa logo and various system icons.

Client	Check	Output	ID	Time
Client 1	Check 1	Output 1	4154	2014-09-29 14:18:16
Client 1	Check 1	Output 1	4155	2014-09-29 14:18:02
Client 1	Check 1	Output 1	1587	2014-09-29 14:18:34
Client 1	Check 1	Output 1	1596	2014-09-29 14:18:00
Client 1	Check 1	Output 1	1587	2014-09-29 14:18:34
Client 1	Check 1	Output 1	1596	2014-09-29 14:18:00
Client 1	Check 1	Output 1	1	2014-09-24 17:03:53
Client 1	Check 1	Output 1	9703	2014-09-29 14:18:05
Client 1	Check 1	Output 1	9703	2014-09-29 14:18:00
Client 1	Check 1	Output 1	13952	2014-09-29 14:18:44
Client 1	Check 1	Output 1	13948	2014-09-29 14:18:54
Client 1	Check 1	Output 1	13951	2014-09-29 14:18:50
Client 1	Check 1	Output 1	13951	2014-09-29 14:18:44
Client 1	Check 1	Output 1	13968	2014-09-29 14:18:50
Client 1	Check 1	Output 1	1587	2014-09-29 14:18:18
Client 1	Check 1	Output 1	205	2014-09-29 14:18:16
Client 1	Check 1	Output 1	4155	2014-09-29 14:18:08
Client 1	Check 1	Output 1	4155	2014-09-29 14:18:01
Client 1	Check 1	Output 1	4155	2014-09-29 14:18:18
Client 1	Check 1	Output 1	3	2014-09-26 14:50:40
Client 1	Check 1	Output 1	2	2014-09-26 14:35:59
Client 1	Check 1	Output 1	1	2014-09-26 12:51:48
Client 1	Check 1	Output 1	4154	2014-09-29 14:18:23
Client 1	Check 1	Output 1	3	2014-09-26 14:25:44
Client 1	Check 1	Output 1	1	2014-09-26 12:19:06
Client 1	Check 1	Output 1	3	2014-09-26 14:41:47
Client 1	Check 1	Output 1	1587	2014-09-29 14:18:20
Client 1	Check 1	Output 1	1596	2014-09-29 14:18:45
Client 1	Check 1	Output 1	1596	2014-09-29 14:18:44
Client 1	Check 1	Output 1	1587	2014-09-29 14:18:18
Client 1	Check 1	Output 1	2	2014-09-26 13:40:02
Client 1	Check 1	Output 1	1	2014-09-26 12:40:27
Client 1	Check 1	Output 1	9703	2014-09-29 14:18:11
Client 1	Check 1	Output 1	9703	2014-09-29 14:18:48
Client 1	Check 1	Output 1	3	2014-09-26 14:50:40
Client 1	Check 1	Output 1	2	2014-09-26 14:35:59
Client 1	Check 1	Output 1	1	2014-09-26 12:51:48
Client 1	Check 1	Output 1	2	2014-09-26 14:40:35
Client 1	Check 1	Output 1	3	2014-09-26 14:25:44
Client 1	Check 1	Output 1	1	2014-09-26 12:19:06
Client 1	Check 1	Output 1	3	2014-09-26 14:41:47
Client 1	Check 1	Output 1	1587	2014-09-29 14:18:20
Client 1	Check 1	Output 1	1596	2014-09-29 14:18:45
Client 1	Check 1	Output 1	1596	2014-09-29 14:18:44
Client 1	Check 1	Output 1	2	2014-09-26 14:35:59
Client 1	Check 1	Output 1	1	2014-09-26 13:40:02
Client 1	Check 1	Output 1	1	2014-09-29 14:18:11
Client 1	Check 1	Output 1	1	2014-09-29 14:18:11
Client 1	Check 1	Output 1	1	2014-09-29 14:18:11
Client 1	Check 1	Output 1	1	2014-09-29 14:18:11
Client 1	Check 1	Output 1	68	2014-09-29 13:52:16



Service Status

```
kwa@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.

kwa@dev13-devc:~$
```



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

Jenkins ▾ services-devops ▸

Build Pipeline



Service Status

```
kwa@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.

kwa@dev13-devc:~$
```



Bad Service Status

```
cluster: nova-devc
instance: demo
Git sha: 8170de11
State: Running - Desired state: Started
Marathon: Warning - up with (3/6) instances. Status: Running.
Mesos: Warning - (3/6) tasks in the TASK_RUNNING state.
Smartstack:
  useastldevc - Critical - in haproxy with (0/3, 0%) total backends UP in this namespace.
  useastlbdevc - 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>

(Check out the Videos and [PaaSTA Principles!](#))

- Sensu:

<https://sensuapp.org/>

- Mesos:

<http://mesos.apache.org/>

- Marathon:

<https://mesosphere.github.io/marathon/>

- Smartstack: <http://nerds.airbnb.com/smartstack-service-discovery-cloud/>

