Mesos + Singularity: PaaS automation for mortals

Gregory Chomatas @gchomatas

PaaS team

HubSpot
120 meters: My shortest travel to a Conference
Those who can, do, the others philosophise...
Really?
Optionality* with Mesos

Invest in a Mesos-powered PaaS and keep doing what you love most; building your product

* Optionality is the property of asymmetric upside (preferably unlimited) with corresponding limited downside (preferably tiny)
The underlying culture & structure

12-factor apps

.net monolith to microservices

Small, autonomous teams with end-to-end ownership - no ops
High productivity

~100 engineers

800+ components that can be updated/scaled independently

QA: ~400 small to medium AWS machines

PROD: ~750 medium to large AWS machines

Source: Martin Fowler
http://martinfowler.com/bliki/MicroservicePremium.html
But for how long...

1. Develop locally

2. Provision QA aws instance
3. Deploy via local Python script

4. Provision PROD aws instance
5. Deploy via local Python script

6. Repeat 4 & 5 to scale

... 

10. Repeat 4&5 at 4am to replace hw
Statically partitioning the cluster is inefficient
The cost of flexibility & asynchronicity

High operational overhead
Poor utilisation & elasticity
Higher rate of failures
Redress the balance with a Mesos-based PaaS

Abstract away machines
Homogenous environment
Scale out in seconds
Centralized deployables registry

Sept 2013: Our First Mesos Cluster
To Boldly go...to Singularity
Singularity: do more with a single scheduler

Great UI & HTTP API
Native Docker Support
Health Checks
Load Balancing API
Log Maintenance

Singularity

Long Running
Batch Scheduling
Manual Running

Web Service
Worker
Scheduled Job
on-demand job
run-once job

Mesos Cluster

Oct 2013: Start building Singularity
Singularity: do even more...

Security / artifact signature verification

Agent & Rack maintenance

Webhooks

Auto-rollback

Email Notifications

Executor cleanup
The PaaS Stack

BUILD
Jenkins

DEPLOY
Orion

RUN
Singularity
The Build / Deploy cycle

- GitHub repository
- S3
- Buildpack runner
- Deployment
The Deployer - Dry run

QA ORION deploy of ORION_NestedMetaConfig

ORION_NestedMetaConfig

ORION_TestWorker

31 (active in QA) → 33 (latest build)

worker

ORION_MetaConfig

ORION_All_Item_Types_In_One_Config

31 (active in QA) → 33 (latest build)

webService
scheduledJob
worker
onDemand
run-once

ORION_TestService

31 (active in QA) → 33 (latest build)

web

ORION_TestJob

31 (active in QA) → 33 (latest build)

testjob
longcron

ORION_All_Item_Types_In_One_Config

build name
OrionTests

build number
31 → 33

repo
HubSpot/OrionTests

branch
master

committer
gchomatas@hubspot.com

commit sha
bb521846 → 1b8af26b [compare]

commit message
Changed config to get a new commit/build

committed
a minute ago
The Deployer - Deploying

QA ORION deploy of ORION_NestedMetaConfig

Started: Thursday, October 1st 2015, 6:53:27 pm,
Deploy Time: 29 seconds
Last Update: 1 seconds ago
Deploys: WAITING DEPLOYS TO FINISH

ORION_NestedMetaConfig — waiting child deploys to finish (••)
  ORION_MetaConfig — waiting child deploys to finish (••)
    ORION_TestService — waiting child deploys to finish (••)
      31 (active in QA) ▸ 33 (latest build)
        web — active ✔
    ORION_All_Item_Types_In_One_Config — waiting child deploys to finish (••)
      31 (active in QA) ▸ 33 (latest build)
        worker — will not deploy this item is paused ⚠
        webService — deployed waiting activation (••)
        scheduledJob — active ✔
        run-once — deployed waiting activation (••)
        onDemand — active ✔
    ORION_TestJob — child deploys finished ✔
      31 (active in QA) ▸ 33 (latest build)
        longcron — active ✔
        testjob — active ✔
    ORION_TestWorker — waiting child deploys to finish (••)
      31 (active in QA) ▸ 33 (latest build)
        worker — deployed waiting activation (••)
Singularity

Deployable view
Singularity Task view

### History

<table>
<thead>
<tr>
<th>Status</th>
<th>Message</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task launched</td>
<td>---</td>
<td>3 hours ago (5 Oct 2015 15:37)</td>
</tr>
<tr>
<td>Task starting</td>
<td>Staging files... (executor pid: 88473)</td>
<td>3 hours ago (5 Oct 2015 15:37)</td>
</tr>
<tr>
<td>Task running</td>
<td>Task running process bash - (pid: 88654)</td>
<td>3 hours ago (5 Oct 2015 15:37)</td>
</tr>
</tbody>
</table>

### Files

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Last modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>root / ORION_TrainService.yml</td>
<td>1.67 KB</td>
<td>5 Oct 2015 15:37</td>
</tr>
<tr>
<td>executor.jar.log</td>
<td>7.93 KB</td>
<td>5 Oct 2015 15:37</td>
</tr>
<tr>
<td>deploy.env</td>
<td>2.7 KB</td>
<td>5 Oct 2015 15:37</td>
</tr>
<tr>
<td>runner.sh</td>
<td>1.86 KB</td>
<td>5 Oct 2015 15:37</td>
</tr>
<tr>
<td>service.log</td>
<td>2.5 KB</td>
<td>5 Oct 2015 15:37</td>
</tr>
<tr>
<td>executor.bash.log</td>
<td>888 B</td>
<td>5 Oct 2015 15:37</td>
</tr>
</tbody>
</table>

S3 Logs
Singularity file tailing

Running java8 with JVM arguments: 

```
INFO [2015-10-01 17:53:37,579] org.eclipse.jetty.setuid.SetUIDListener: Opened TestService@53e3a87a{HTTP/1.1}{0.0.0.0}
INFO [2015-10-01 17:53:37,781] io.dropwizard.jersey.DropwizardResourceConfig: The following paths were found for the
```

```
GET /procfileitems/{procfileItemId} (com.hubspot.deploy.testservice.resources.ProcfileResource)
```
Singularity Health Checks & Resources

**Healthchecks**

Beginning on **Task running**, hit `/admin/healthcheck` with a 5 second timeout every 5 second(s) until **HTTP 200** is received, or 120 seconds have elapsed.

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>Duration</th>
<th>Status</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Oct 2015</td>
<td>137ms</td>
<td><strong>HTTP 200</strong></td>
<td>{&quot;TestServiceHealthCheck&quot;:{&quot;healthy&quot;:true,&quot;message&quot;:&quot;Test service is healthy. Free memory is 1930MB&quot;,&quot;deadlocks&quot;:{&quot;healthy&quot;:true}}}</td>
</tr>
</tbody>
</table>

**Resource Usage**

- **Memory (rss vs limit)**
  - 4.46 GB / 4.49 GB

- **CPU Usage**
  - 0.25 used / 12 allocated CPUs

- **Memory (anon)**
  - 3.43 GB

- **Memory (file)**
  - 1.02 GB

- **Memory (mapped file)**
  - 36 KB
**Singularity All Deployables view**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>State</th>
<th>Deploy time</th>
</tr>
</thead>
<tbody>
<tr>
<td>ORION_TestService-web</td>
<td>Service</td>
<td>Active</td>
<td>4 minutes ago (5 Oct 2015 19:02)</td>
</tr>
<tr>
<td>SignalsGatingService-web</td>
<td>Service</td>
<td>Active</td>
<td>4 minutes ago (5 Oct 2015 19:02)</td>
</tr>
<tr>
<td>LeadinFormProcessorKafka-consume</td>
<td>Worker</td>
<td>Active</td>
<td>8 minutes ago (5 Oct 2015 18:58)</td>
</tr>
<tr>
<td>LeadinJobs-sendWeeklyEmails</td>
<td>Scheduled</td>
<td>Active</td>
<td>8 minutes ago (5 Oct 2015 18:58)</td>
</tr>
<tr>
<td>LeadinDatabaseMigration-db migrate</td>
<td>On demand</td>
<td>Active</td>
<td>8 minutes ago (5 Oct 2015 18:58)</td>
</tr>
<tr>
<td>LeadinJobs-formSubmissionEmails</td>
<td>Worker</td>
<td>Active</td>
<td>8 minutes ago (5 Oct 2015 18:58)</td>
</tr>
</tbody>
</table>
Singularity
Cluster Status

Requests
- 1766 active (97%)
- 50 paused (3%)
- 9 cooling down (0%)
- 0 pending
- 0 cleaning
- 0 Over Provisioned
- 0 Under Provisioned
1825 Total

Tasks
- 1003 active (77%)
- 285 scheduled (23%)
- 0 overdue
- 0 cleaning
- 0 load balancer cleanup
Max Task Lag: (no lag)
1298 Total

Racks
- 3 Active Racks
- 0 Decommissioning Racks
- 0 Inactive Racks

Slaves
- 18 Active Slaves
- 0 Decommissioning Slaves
- 0 Inactive Slaves
- 0 Unknown Slaves

Deploys
- 0 Active Deploys

Singularity scheduler instances

<table>
<thead>
<tr>
<th>Hostname</th>
<th>Driver status</th>
<th>Uptime</th>
<th>Time since last offer</th>
</tr>
</thead>
<tbody>
<tr>
<td>rozal.iad02.hubspot-networks.net</td>
<td>Driver not started</td>
<td>18 minutes</td>
<td>—</td>
</tr>
<tr>
<td>bigpidgeon.iad02.hubspot-networks.net</td>
<td>Driver not started</td>
<td>18 minutes</td>
<td>—</td>
</tr>
<tr>
<td>ojofrio.iad02.hubspot-networks.net</td>
<td>Driver running</td>
<td>18 minutes</td>
<td>a few seconds</td>
</tr>
</tbody>
</table>
Singularity

Cluster Maintenance view

---

### Slaves

#### Active

<table>
<thead>
<tr>
<th>ID</th>
<th>State</th>
<th>Since</th>
<th>Hash</th>
<th>Host</th>
<th>Uptime</th>
<th>Archival Roy</th>
</tr>
</thead>
<tbody>
<tr>
<td>23:02:18-02:00-00-01-0299-0-9</td>
<td>Active</td>
<td>23 Sep 2019 12:10</td>
<td>uo-549</td>
<td>red-water.lab2.ubc.ca</td>
<td>2 days</td>
<td></td>
</tr>
<tr>
<td>23:02:18-02:00-00-01-0299-0-9</td>
<td>Active</td>
<td>23 Sep 2019 11:51</td>
<td>uo-549</td>
<td>red-water.lab2.ubc.ca</td>
<td>2 days</td>
<td></td>
</tr>
<tr>
<td>23:02:18-02:00-00-01-0299-0-9</td>
<td>Active</td>
<td>23 Sep 2019 10:13</td>
<td>uo-549</td>
<td>red-water.lab2.ubc.ca</td>
<td>2 days</td>
<td></td>
</tr>
</tbody>
</table>

#### Frozen

No slaves

#### Decommissioning

No slaves

---

### Inactive

<table>
<thead>
<tr>
<th>ID</th>
<th>State</th>
<th>Since</th>
<th>Hash</th>
<th>Host</th>
<th>Uptime</th>
<th>Archival Roy</th>
</tr>
</thead>
<tbody>
<tr>
<td>23:02:18-02:00-00-01-0299-0-9</td>
<td>Deactive</td>
<td>23 Sep 2019 14:44</td>
<td>uo-549</td>
<td>red-water.lab2.ubc.ca</td>
<td>6 days</td>
<td></td>
</tr>
</tbody>
</table>
Migration to Mesos - Timeline

- **Start building Orion deployer**
- **Start migration of QA environment**
- **Complete migration of QA environment**
  - **Start PROD**
- **Amazon Rebootpocalypse**
- **Ghidorah**
  - **QA Load Balancers in Mesos**
- **99% migration of PROD environment**
- **QA**: 1814 deployables, 18 agents (m4.10xlarge)
- **PROD**: 1664 deployables, 134 agents (c3.8xlarge), 15204 CPUs, 8TB Ram
- **1,400,000 tasks/month**

Timeline:
- Dec 2013
- Jan 2014
- Feb
- Mar
- Apr
- May
- Jun
- Jul
- Aug
- Sep
- Oct
- Nov
- Dec
- Jan 2015
- Feb
- Mar
- Apr
- May
- Jun
- Jul
- Aug
- Sep
- Oct
- Nov
Manual Server Provisioning

- App Servers
- Mesos Slaves

Singularity first deploy

PROD migrated

Server provisioning UI usage

PROD migrated
1800+ deployables

~300 deploys / day

10 minutes from git

Push to production
I want persistent IMAP connections to 200k+ inboxes (Jul 2015)

I NEED MORE POWER, SCOTTY

I JUST CANNOT DO IT! I D DON'T HAVE THE POOWER!
CHALLENGES AHEAD
Migration Issues

Stateful Services

Single Process services

Hard coded stationary hosts

Cgroups memory isolation

User resistance
Operational Issues

All eggs in one basket

Mesos / Framework issues (pingback port)

Failures (Zookeeper, Mesos, Singularity)

Cluster Maintenance

Missing features
Maintenance
Phased rollout of new Kernel, Instance types

Rolling upgrade of instance basic SW with puppet vars

Rolling upgrade of master/agent process with ansible

Local testing on docker cluster

Roll out at **infra-QA** then **product-QA** and last to **Production** cluster

Deploy tools deploy themselves but maintain command line alternative with fabric
Optionality with Mesos@HubSpot

- Blazar: Massive Builds in Mesos
- Baragon: Tasks Load Balancer Manager
- Ghidorah: Load Balancers in Mesos
- Mesos Spark Cluster
- Singularity
The sweet spot

Source: Mark Leslie (http://firstround.com/review/The-Arc-of-Company-Life-and-How-to-Prolong-It/)
Invest early in a deploy & build infrastructure

Dedicate 1-2 engineers to experiment on a Mesos powered PaaS

Try Singularity today!

github.com/HubSpot/Singularity
Useful links

HubSpot Blog: How We Built Our Stack For Shipping at Scale

Blazar: An out-of-this world build system!

Baragon: Load Balancer API