CONTINUOUS DEPLOYMENT WITH SINGULARITY

Large Scale Mission-Critical Service and Job Deployment

Gregory Chomatas
@gchomatas
PAAS TEAM

Implement & maintain:
- the deploy & build tools
- the PAAS platform (mesos clusters)
- load balancer tools
- logging infrastructure

Boston: Whitney Sorenson, Tom Petr, Tim Finley

Dublin: Gregory Chomatas, Kieran Manning
AN ESSENTIAL SINGULARITY $\exp(1/Z)$
THE WAY TO MESOS

Speed wins -> Speed Product Development

Increase change rate -> Remove Friction + Reduce size, cost, risk of change:

small teams, high trust, low process freedom and responsibility culture

micro services
libs & cross cutting APIs to simplify coding
automate deployment by tooling
SOME FACTS & NUMBERS

3-4 person teams
several micro-services & jobs per team (full operation)
1 or more services per dev

All QA in MESOS / Part of PROD with plan to move all

400 deploys / day - 843 Deployable Items:
219 WEB SERVICES (long running with an API)
246 WORKERS (long running no API)
205 SCHEDULED JOBS (CRON schedule)
173 ON-DEMAND
QA Environment

pre-mesos:
400 small & medium size servers (c1.xlarge)

post-mesos:
20 big servers (c3.8xlarge)
WHY SINGULARITY

almost no framework 1 year ago
get a consistent, unified API for all deployable items

mission critical / strategic tool - important to control:
priority and delivery of bug fixes
features and integrations
the overall roadmap

have the resources to implement & maintain a highly complex piece of software
name: MDS_All_Item_Types_In_One_Config
buildName: MesosDeployIntegrationTestsProject
type: procfile

owners:
- user@hubspot.com

appRoot: /mesos-deploy-test-srv1/v1
loadBalancers:
  - test

env:
  all:
    JOB_JAR: TestJob.jar

procfile:
  webService:
    cmd: java $JVM_DEFAULT_OPTS -jar TestService.jar server $CONFIG_YAML
    instances: 2
    cpus: 2
    memory: 1024
    numRetriesOnFailure: 5
  scheduledJob:
    cmd: java $JVM_DEFAULT_OPTS -jar $JOB_JAR -testjob
    schedule: '*/3 * * * *'
    numRetriesOnFailure: 5
DEPLOY WITH HUBSPOT PAAS

1. User adds Deploy Config in project
2. User pushes project to code repo
3.1. Deploy Config is registered to DMR
3.2. Project build is triggered
4.1. Project Artifacts are stored in Artifact Repo
4.2. Build info is stored in DMR
5. User initiates a deploy
6. Deployer uses MDS API to post a Deploy Request
7. MDS registers + deploys deployable items using Singularity API. Polis Singularity for Task Completion and sends deploy status to deployer
8. Singularity creates mesos tasks
9. Upon Deploy Success or Failure MDS writes deploy metadata to DMR
SINGULARITY SCHEDULER

A DEPLOY-CENTRIC REST API TO:

- register deployable items
- execute their deploys
- view sandbox files
- get metadata / historical data
SINGULARITY SCHEDULER

Advanced features:

Health Checking at the process and the service endpoint level

Automatic cool-down of repeatedly failing services

Load balancing of service instances (LB API)

Automatic Rollback of failed deploys

Reconciliation of LOST tasks

Decommissioning of Slaves & Racks
Log Rotation

Task Sandbox Cleanup

Graceful Task Killing with configurable timeout

Environment Setup

Task Runner Script
ADVANCED SLAVE SERVICES

Log Watcher : Tail & Stream Logs

S3 uploader : Archive logs with AWS S3 Service

Executor Cleanup : Clean failed executor tasks

OOM Killer : replace the default memory limit checking supported by Linux Kernel CGROUPS
KEY SINGULARITY ABSTRACTIONS

SINGULARITY REQUEST OBJECT

```json
{
    "id": "TestService",
    "owners": [ 
        "feature_x_team@mycompany.com",
        "developer@mycompany.com"
    ],
    "daemon": true,
    "instances": 3,
    "rackSensitive": true,
    "loadBalanced": true
}
```
SINGULARITY DEPLOY OBJECT

RESOURCES: Memory, CPUs, network ports
HEALTH CHECKS: Timeouts and URLs
LOAD BALANCING of web service instances (LB groups, api base path)
EXECUTOR INFORMATION: execution environment, executable artifacts, configuration files, command to execute, executor to use, etc.
{  
  "requestId": "MDS_TestService",
  "id": "71_7",
  "customExecutorCmd": "../singularity-executor",
  "resources": {
    "cpus": 1,
    "memoryMb": 896,
    "numPorts": 3
  },
  "env": {
    "DEPLOY_MEM": "768",
    "JVM_MAX_HEAP": "384m"
  }
}
"executorData": {
    "cmd": "java -Xmx$JVM_MAX_HEAP -jar .../TestService.jar server $CONFIG_YAML",
    "embeddedArtifacts": [
        {
            "name": "rawDeployConfig",
            "filename": "TestService.yaml",
            "content": "bmFtZT..."
        }
    ],
    "externalArtifacts": [],
    "s3Artifacts": [
        {
            "name": "executableSlug",
            "filename": "TestService.tar.gz",
            "md5sum": "313be85c5979a1c652ec93e305eb25e9",
            "filesize": 81055833,
            "s3Bucket": "hubspot.com",
            "s3ObjectKey": "build_artifacts/.../TestService.tar.gz"
        }
    ]
}
SINGULARITY API

MANAGE DEPLOYABLE ITEMS

ENDPOINT: /requests

register / update / unregister an item
get info about an item
list items in  active | paused | cool-down state
run / restart / pause / un-pause an item
SINGULARITY API

DEPLOY THE DEPLOYABLE ITEMS

ENDPOINT: /deloys

- deploy an already registered item
- cancel a pending deploy
MANAGE DEPLOYABLE ITEM INSTANCES (TASKS)

ENDPOINT: /tasks

- get the list of all scheduled tasks (not yet active)
- get scheduled tasks for a specific item
- list tasks in active | cleaning | lbcleanup state
- info about a specific task
- active tasks in a slave
- Kill a task
SINGULARITY API

Historical Information about deployable items & their tasks

ENDPOINT: /history

- a single task history
- tasks that have run in the past
- all previous item updates
- search for historical items by item id
- all item deploys
- a specific item deploy
LIST & DOWNLOAD FILES IN ACTIVE TASK SANDBOX

ENDPOINT: /sandbox

- list all task files
- read file chunks
- download a file
Cluster STATE Information

ENDPOINT: /state

```json
{
    "activeTasks": 567,
    "activeRequests": 843,
    "cooldownRequests": 1,
    "scheduledTasks": 142,
    "pendingRequests": 0,
    "lbCleanupTasks": 1,
    "activeSlaves": 21,
    "deadSlaves": 0,
    "decommissioningSlaves": 0,
    "activeRacks": 3,
    "deadRacks": 0,
    "futureTasks": 142,
    "maxTaskLag": 0,
    "overProvisionedRequests": 0,
    "underProvisionedRequests": 0,
    "allRequests": 844
}
```
# Singularity UI - Global Cluster Status

## Hosts

<table>
<thead>
<tr>
<th>Hostname</th>
<th>Driver status</th>
<th>Uptime</th>
<th>Time since last offer</th>
</tr>
</thead>
<tbody>
<tr>
<td>wayma.io9f.hubspot-networks.net</td>
<td>Driver not started</td>
<td>43 minutes</td>
<td>–</td>
</tr>
<tr>
<td>benjen.io9f.hubspot-networks.net</td>
<td>Driver not started</td>
<td>43 minutes</td>
<td>–</td>
</tr>
<tr>
<td>joel.io9f.hubspot-networks.net</td>
<td>Driver running</td>
<td>43 minutes</td>
<td>a few seconds</td>
</tr>
</tbody>
</table>

## Requests

<table>
<thead>
<tr>
<th>Status</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>844</td>
</tr>
<tr>
<td>Active</td>
<td>844</td>
</tr>
<tr>
<td>Cool Dow</td>
<td>0</td>
</tr>
<tr>
<td>Paused</td>
<td>0</td>
</tr>
<tr>
<td>Pending</td>
<td>0</td>
</tr>
<tr>
<td>Cleaning</td>
<td>0</td>
</tr>
</tbody>
</table>

## Tasks

<table>
<thead>
<tr>
<th>Status</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>536</td>
</tr>
<tr>
<td>Overdue</td>
<td>0</td>
</tr>
<tr>
<td>Future</td>
<td>173</td>
</tr>
<tr>
<td>Cleaning</td>
<td>0</td>
</tr>
</tbody>
</table>

## Racks

<table>
<thead>
<tr>
<th>Status</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>3</td>
</tr>
<tr>
<td>Dead</td>
<td>0</td>
</tr>
<tr>
<td>Decommissioning</td>
<td>0</td>
</tr>
</tbody>
</table>

## Slaves

<table>
<thead>
<tr>
<th>Status</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>21</td>
</tr>
<tr>
<td>Dead</td>
<td>0</td>
</tr>
<tr>
<td>Decommissioning</td>
<td>0</td>
</tr>
</tbody>
</table>
## SINGULARITY UI - DEPLOYABLE ITEM LIST

<table>
<thead>
<tr>
<th>Name</th>
<th>State</th>
<th>Instances</th>
<th>Deploy ID</th>
<th>Deploy user</th>
<th>Deploy time</th>
<th>Schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDS_All_Item_Types_In_One_Config-onDemand</td>
<td>Active</td>
<td>1</td>
<td>78_1</td>
<td>gchomatias</td>
<td>2 minutes ago</td>
<td></td>
</tr>
<tr>
<td>MDS_All_Item_Types_In_One_Config-worker</td>
<td>Active</td>
<td>2</td>
<td>78_1</td>
<td>gchomatias</td>
<td>2 minutes ago</td>
<td></td>
</tr>
<tr>
<td>MDS_All_Item_Types_In_One_Config-scheduledJob</td>
<td>Active</td>
<td>1</td>
<td>78_1</td>
<td>gchomatias</td>
<td>2 minutes ago</td>
<td></td>
</tr>
<tr>
<td>MDS_All_Item_Types_In_One_Config-webService</td>
<td>Active</td>
<td>2</td>
<td>71_7</td>
<td>gchomatias</td>
<td>6 days ago</td>
<td></td>
</tr>
<tr>
<td>MDS_TestService_Travelview_Enabled-web</td>
<td>Active</td>
<td>1</td>
<td>63_1m</td>
<td>gchomatias</td>
<td>10 days ago</td>
<td></td>
</tr>
<tr>
<td>MDS_TestJob-testjob</td>
<td>Active</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Running instances

<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>Deploy ID</th>
<th>Started</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDS_All_Item.Types_In_One_Config-webService</td>
<td>Task running</td>
<td>78_2</td>
<td>11 minutes ago</td>
<td>11 minutes ago</td>
</tr>
<tr>
<td>MDS_All_Item.Types İn-One_Config-webService</td>
<td>Task running</td>
<td>78_2</td>
<td>11 minutes ago</td>
<td>11 minutes ago</td>
</tr>
</tbody>
</table>

### Task history

<table>
<thead>
<tr>
<th>Name</th>
<th>Last state</th>
<th>Deploy ID</th>
<th>Started</th>
<th>Updated</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDS_All_Item.Types İn-One_Config-webService</td>
<td>Task killed</td>
<td>78_1</td>
<td>18 minutes ago</td>
<td>11 minutes ago</td>
</tr>
<tr>
<td>MDS_All_Item.Types İn-One_Config-webService</td>
<td>Task killed</td>
<td>78_1</td>
<td>18 minutes ago</td>
<td>11 minutes ago</td>
</tr>
</tbody>
</table>

### Deploy history

<table>
<thead>
<tr>
<th>Deploy ID</th>
<th>Status</th>
<th>User</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>78_2</td>
<td>Succeeded</td>
<td>gchomatas</td>
<td>11 minutes ago (7 Aug 2014 23:00)</td>
</tr>
<tr>
<td>78_1</td>
<td>Succeeded</td>
<td>gchomatas</td>
<td>18 minutes ago (7 Aug 2014 22:54)</td>
</tr>
</tbody>
</table>

### Request history

<table>
<thead>
<tr>
<th>State</th>
<th>User</th>
<th>Created</th>
</tr>
</thead>
<tbody>
<tr>
<td>Created</td>
<td>gchomatas</td>
<td>18 minutes ago</td>
</tr>
</tbody>
</table>
SINGULARITY UI - DEPLOYABLE ITEM TASK

Request MDS All Item Types In One Config-webService

MDS_All_Item_Types_In_One_Config-webService-78_2-140778235...

Task running as of 15 minutes ago (11 Aug 2014 19:39) — Task running process bash - (pid: 473)

History

<table>
<thead>
<tr>
<th>Status</th>
<th>Message</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task starting</td>
<td>Staging files... (executor pid: 566)</td>
<td>15 minutes ago (11 Aug 2014 19:39)</td>
</tr>
<tr>
<td>Task running</td>
<td>Task running process bash - (pid: 473)</td>
<td>15 minutes ago (11 Aug 2014 19:39)</td>
</tr>
</tbody>
</table>

Files

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Last modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>app/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>bin/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>conf/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>logs/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MDS_All_Item_Types_In_One_Config.yaml</td>
<td>1 KB</td>
<td>11 Aug 2014 19:39</td>
</tr>
<tr>
<td>deploy.env</td>
<td>2 KB</td>
<td>11 Aug 2014 19:39</td>
</tr>
<tr>
<td>executor.bash.log</td>
<td>433 B</td>
<td>11 Aug 2014 19:39</td>
</tr>
<tr>
<td>executor.java.log</td>
<td>5 KB</td>
<td>11 Aug 2014 19:39</td>
</tr>
<tr>
<td>runner.sh</td>
<td>1 KB</td>
<td>11 Aug 2014 19:39</td>
</tr>
<tr>
<td>service.log</td>
<td>112 KB</td>
<td>11 Aug 2014 19:53</td>
</tr>
</tbody>
</table>
### SINGULARITY UI - HISTORICAL TASK

#### MDS_All_Item_Types_In_One_Config-webService-78_2-140778234...

Task killed as of 22 minutes ago (11 Aug 2014 19:39) — Task killed. Process exited gracefully with code 143

### History

<table>
<thead>
<tr>
<th>Status</th>
<th>Message</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task</td>
<td>Staging files... (executor pid: 8873)</td>
<td>22 minutes ago (11 Aug 2014 19:39)</td>
</tr>
<tr>
<td>Task</td>
<td>Task running process bash - (pid: 3608)</td>
<td>22 minutes ago (11 Aug 2014 19:39)</td>
</tr>
<tr>
<td>Task</td>
<td>SCALING_DOWN</td>
<td>22 minutes ago (11 Aug 2014 19:39)</td>
</tr>
<tr>
<td>Task</td>
<td>Task killed. Process exited gracefully with code 143</td>
<td>22 minutes ago (11 Aug 2014 19:39)</td>
</tr>
</tbody>
</table>

### Files

#### MDS_All_Item_Types_In_One_Config-webService-78_2-140778234160...

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Last modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>.cmd/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>.log/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>logs/</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MDS_All_Item_Types_In_One_Config-yaml</td>
<td>1 KB</td>
<td>11 Aug 2014 19:39</td>
</tr>
<tr>
<td>executor bash.log</td>
<td>423 B</td>
<td>11 Aug 2014 19:39</td>
</tr>
<tr>
<td>tail_of_finished_service.log</td>
<td>21 KB</td>
<td>11 Aug 2014 19:39</td>
</tr>
<tr>
<td>logfolder.status</td>
<td>758 B</td>
<td>11 Aug 2014 19:39</td>
</tr>
<tr>
<td>display.env</td>
<td>2 KB</td>
<td>11 Aug 2014 19:39</td>
</tr>
<tr>
<td>runner.sh</td>
<td>1 KB</td>
<td>11 Aug 2014 19:39</td>
</tr>
<tr>
<td>service.log</td>
<td>6 B</td>
<td>11 Aug 2014 19:39</td>
</tr>
<tr>
<td>executorjava.log</td>
<td>16 KB</td>
<td>11 Aug 2014 19:39</td>
</tr>
</tbody>
</table>

### S3 Logs

#### MDS_All_Item_Types_In_One_Config-webService-78_5-1407782541880-0-elia_us_east...

<table>
<thead>
<tr>
<th>Log file</th>
<th>Size</th>
<th>Last modified</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>53 KB</td>
<td>11 Aug 2014 19:41</td>
</tr>
</tbody>
</table>
## Racks

### Active racks

<table>
<thead>
<tr>
<th>ID</th>
<th>Uptime</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>us_east_1b</td>
<td>19 hours (8/8/2014 9:08pm)</td>
<td>Decommission</td>
</tr>
<tr>
<td>us_east_1a</td>
<td>19 hours (8/8/2014 9:08pm)</td>
<td>Decommission</td>
</tr>
<tr>
<td>us_east_1e</td>
<td>19 hours (8/8/2014 9:08pm)</td>
<td>Decommission</td>
</tr>
</tbody>
</table>

## Slaves

### Active slaves

<table>
<thead>
<tr>
<th>ID</th>
<th>Rack</th>
<th>Host</th>
<th>Uptime</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>20140724-180527-3834914056-5050-7632-29</td>
<td>us_east_1b</td>
<td>darklyn</td>
<td>19 hours (8/8/2014 9:08pm)</td>
<td>Decommission</td>
</tr>
<tr>
<td>20140724-180527-3834914056-5050-7632-27</td>
<td>us_east_1e</td>
<td>lorne</td>
<td>19 hours (8/8/2014 9:08pm)</td>
<td>Decommission</td>
</tr>
<tr>
<td>20140724-180527-3834914056-5050-7632-37</td>
<td>us_east_1b</td>
<td>dany</td>
<td>19 hours (8/8/2014 9:08pm)</td>
<td>Decommission</td>
</tr>
<tr>
<td>20140724-180527-3834914056-5050-7632-28</td>
<td>us_east_1a</td>
<td>blackhair</td>
<td>19 hours (8/8/2014 9:08pm)</td>
<td>Decommission</td>
</tr>
<tr>
<td>20140724-180527-3834914056-5050-7632-25</td>
<td>us_east_1b</td>
<td>zhoe</td>
<td>19 hours (8/8/2014 9:08pm)</td>
<td>Decommission</td>
</tr>
</tbody>
</table>
DEVELOP WITH SINGULARITY

java 7
guice
dropwizard
(jersey, jackson, liquibase)
maven
backbone
nodejs
brunch
ROADMAP / NEW FEATURES

Enhance Job Scheduler

Support deploy of Docker containers

Add advanced slave affinity algorithms to support data locality for Big Data Analysis tasks

Open source Deployer
(A simplified version of Deploy Metadata Registry + Mesos Deploy Service + Deployer UI)
http://getsingularity.com/
https://github.com/HubSpot/Singularity

https://github.com/HubSpot/Singularity/blob/master/Docs/Singularity_API_Reference.md
https://github.com/HubSpot/Singularity/blob/master/Docs/Singularity_Local_Setup_For_Testing.md
https://mesosphere.io/resources/mesos-case-study-hubspot/