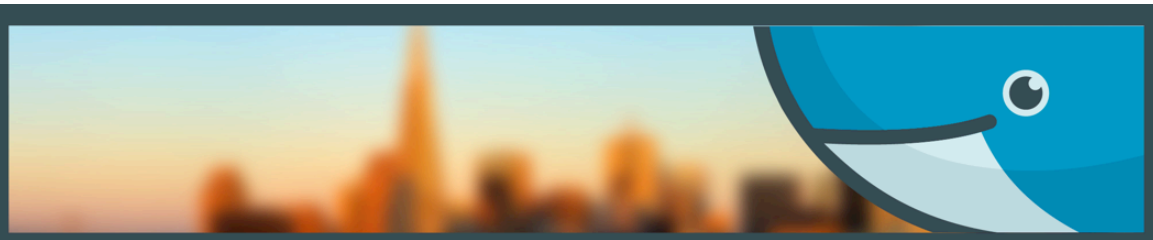


Scaling New Services

From Container Creation to Automated Deployment

@dontrebootme @brainscott #dockercon

dockercon 15

SF JUNE 22-23

Introduction

Who we are

What we do

Why *Docker*

Where we are ***going***


Break it down

chef

chef

the components

```
# ZK params for mesos
default['bi_mesos']['zk_hosts'] = 'zk://localhost:2181'
default['bi_mesos']['zk_path'] = '/mesos'
default['bi_mesos']['zk_url'] = "#{node['bi_mesos']['zk_host

default['bi_mesos']['ip'] = "#{node['ipaddress']}"

# Brought in for mesos cookbook usage
default['mesos']['version'] = '0.22.1'
#default['mesos']['master']['bin'] = '/usr/local/sbin/mesos-
default['mesos']['master']['flags']['cluster'] = 'MyMesosClu
default['mesos']['master']['flags']['quorum'] = '1'
default['mesos']['master']['flags']['zk'] = "#{node['bi_meso
#default['mesos']['slave']['bin'] = '/usr/local/sbin/mesos-s
default['mesos']['slave']['flags']['containerizers'] = 'dock
default['mesos']['slave']['flags']['docker_remove_delay'] =
default['mesos']['slave']['flags']['isolation'] = 'cgroups/c
default['mesos']['slave']['flags']['master'] = "#{node['bi_m

# Mesosphere Marathon
default['bi_mesos']['marathon']['version'] = 'v0.8.2'
default['bi_mesos']['marathon']['image'] = 'mesosphere/marat
default['bi_mesos']['marathon']['zk_path'] = "/marathon"
default['bi_mesos']['marathon']['command'] = "--master #{nod
  " --zk #{node['bi_mesos']['zk_hosts']}#{node['bi_mesos']['
  " --hostname #{node['fqdn']}" \
  " --max_tasks_per_offer 10" \
  " --http_port 80" \
  " --logging_level warn"

# Airbnb Chronos
default['bi_mesos']['chronos']['version'] = '2.3.2-mesos0.21
default['bi_mesos']['chronos']['image'] = 'tomaskral/chronos
```

chef configures

mesos, mesos frameworks, haproxy and consul

Chef the initial containers

```
marathon_version = node['bi_mesos']['marathon']['version']
marathon_image = node['bi_mesos']['marathon']['image']
marathon_command = node['bi_mesos']['marathon']['command']

# Pull the image
docker_image marathon_image do
  tag marathon_version
  # Since we are version pinning and not pulling :latest, pull_if_missing should be sufficient
  action :pull_if_missing
end

# Run it
docker_container 'marathon' do
  container_name 'mesos_marathon'
  net 'host'
  image "#{marathon_image}:#{marathon_version}"
  detach true
  command "#{marathon_command}"
end
```

docker

packaging

and deployment of our services

```
FROM gliderlabs/alpine
MAINTAINER  Patrick OConnor <patrick@dontreboot.me>

# Install nginx, make it work well with docker, and get it ready to run
RUN apk-install nginx && echo "daemon off;" >> /etc/nginx/nginx.conf \
 && mkdir -p /tmp/nginx/client-body

# Bring in the baymax
COPY html /usr/html
COPY start_nginx.sh /

# Expose port 80
EXPOSE 80

# Run nginx
CMD /start_nginx.sh
```

dockerfile

exists in the application repo

application build

includes docker build/push

mesos

cluster scheduler

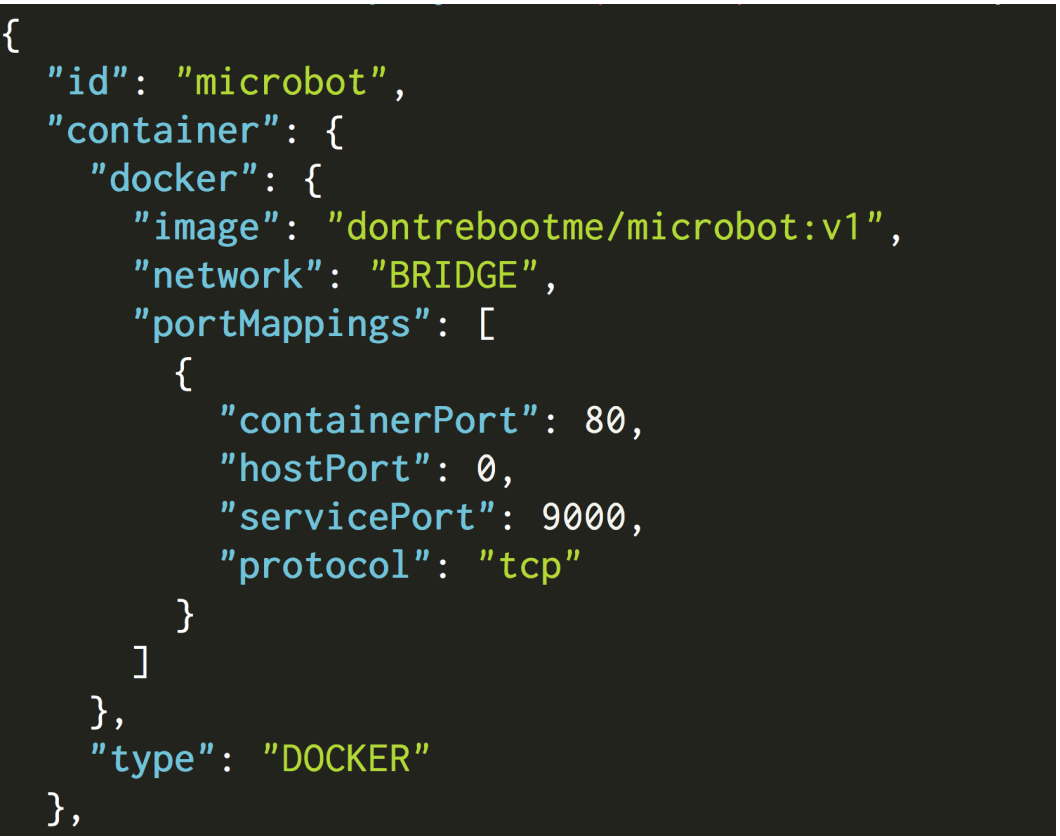
manage

resources and tasks

marathon

docker on mesos

can **describe**
docker...



deployments

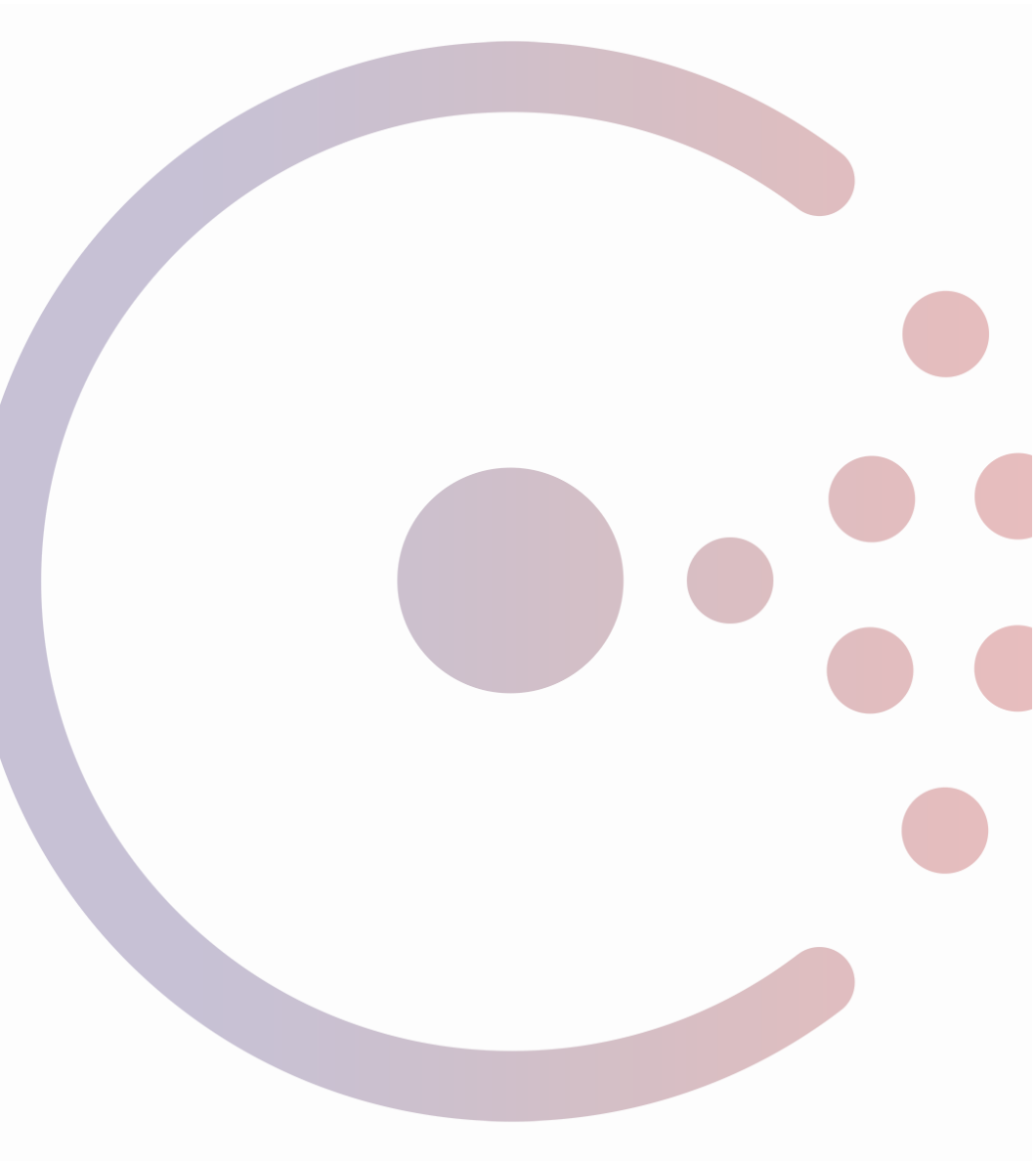
container **health**

check

```
    "type": "DOCKER"
  },
  "healthChecks": [
    {
      "protocol": "HTTP",
      "path": "/"
    }
  ],
  "cpus": 0.01,
  "mem": 32,
```

upgrade strategy

```
  "cpus": 0.01,
  "mem": 32,
  "instances": 100,
  "upgradeStrategy": {
    "minimumHealthCapacity": 0.9,
    "maximumOverCapacity": 0.1
  }
}
```



consul

distributed
key/value storage

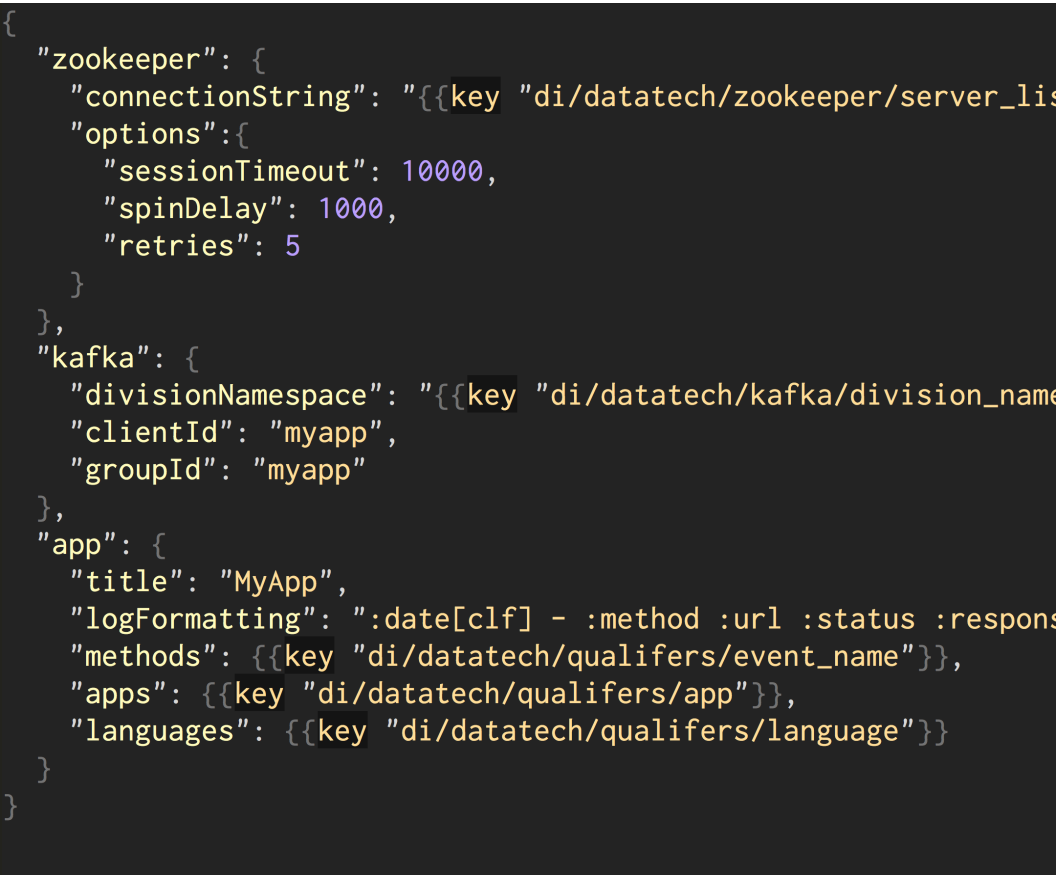
enables containers
to get environment specific
configuratiation

seeding consul data

```
#####
# KAFKA KVPs
#####
curl -X PUT -d "${ckafka_brokers}" http://${cserver}:8500/v1/kv/di/datatech/kafka/metadata_broker_list
curl -X PUT -d '0' http://${cserver}:8500/v1/kv/di/datatech/kafka/request_required_acks
curl -X PUT -d 'async' http://${cserver}:8500/v1/kv/di/datatech/kafka/producer_type
curl -X PUT -d 'gzip' http://${cserver}:8500/v1/kv/di/datatech/kafka/compression_codec
curl -X PUT -d '500' http://${cserver}:8500/v1/kv/di/datatech/kafka/batch_num_messages
curl -X PUT -d '5000' http://${cserver}:8500/v1/kv/di/datatech/kafka/queue_buffering_max_ms
curl -X PUT -d '' http://${cserver}:8500/v1/kv/di/datatech/kafka/division_namespace
curl -X PUT -d "${kafka_metadata_fetch_timeout_ms}" http://${cserver}:8500/v1/kv/di/datatech/kafka/metad
ch_timeout_ms

#####
# ZOOKEEPER KVPs
#####
curl -X PUT -d "${czk_server_list}" http://${cserver}:8500/v1/kv/di/datatech/zookeeper/server_list

#####
# HADOOP KVPs
#####
curl -X PUT -d "${hadoop_cm}" http://${cserver}:8500/v1/kv/di/datatech/hadoop/cm
curl -X PUT -d "${hadoop_nn}" http://${cserver}:8500/v1/kv/di/datatech/hadoop/nn
```



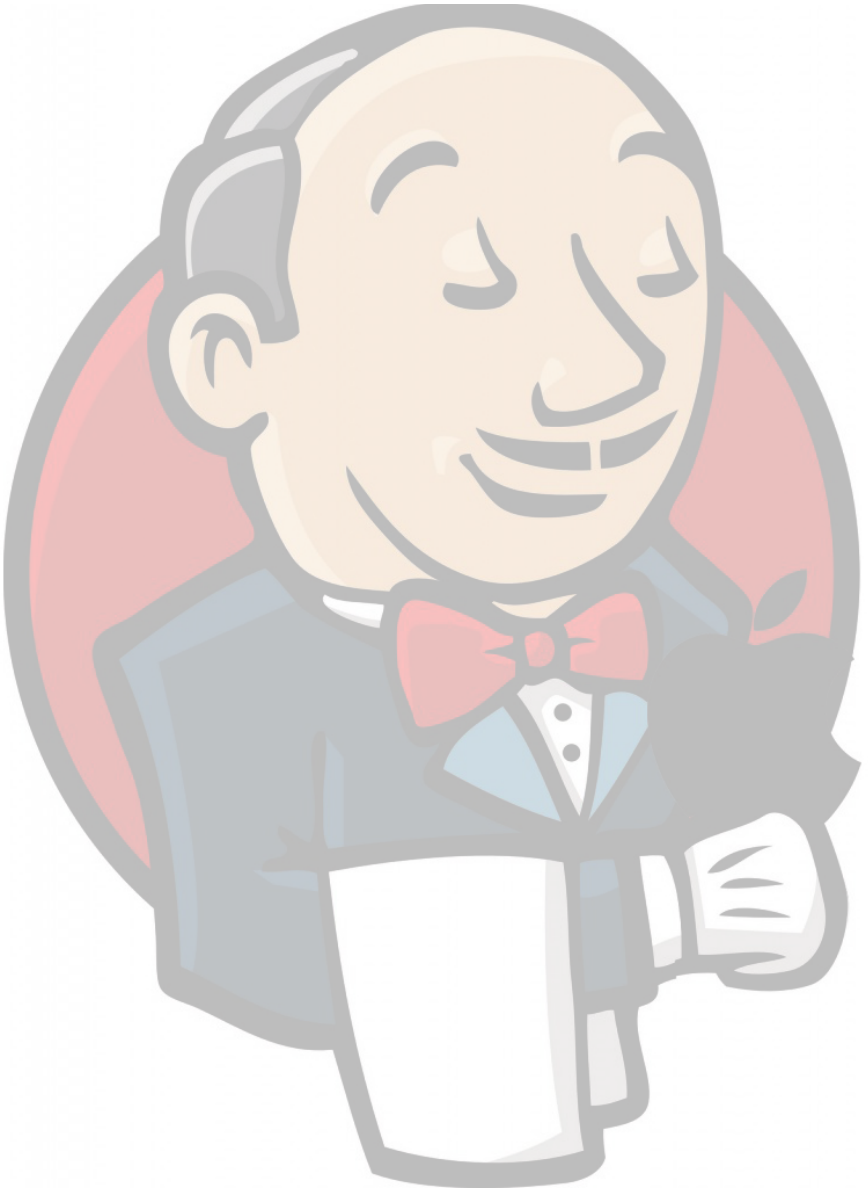
```
{
  "zookeeper": {
    "connectionString": "{{key "di/datatech/zookeeper/server_lis
    "options":{
      "sessionTimeout": 10000,
      "spinDelay": 1000,
      "retries": 5
    }
  },
  "kafka": {
    "divisionNamespace": "{{key "di/datatech/kafka/division_name
    "clientId": "myapp",
    "groupId": "myapp"
  },
  "app": {
    "title": "MyApp",
    "logFormatting": ":date[clf] - :method :url :status :respons
    "methods": {{key "di/datatech/qualifers/event_name"}},
    "apps": {{key "di/datatech/qualifers/app"}},
    "languages": {{key "di/datatech/qualifers/language"}}
  }
}
```

in the **container**

consul template

generated at launch

jenkins



build
push
deploy

And now...

demo

Thank you #dockercon

Patrick O'Connor

@dontrebootme
spof.io

Brian Scott

@brainscott
blog.bscott.io

dockercon 15