Cracking the Container Scale Problem with Apache Mesos

Connor Doyle
connor@mesosphere.io

Sunil Shah
sunil@mesosphere.io


We are Mesosphere

240 million monthly active users

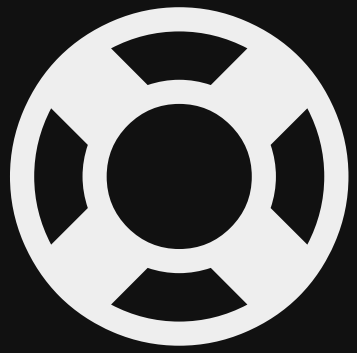
500 million tweets per day

Up to 150k tweets per second

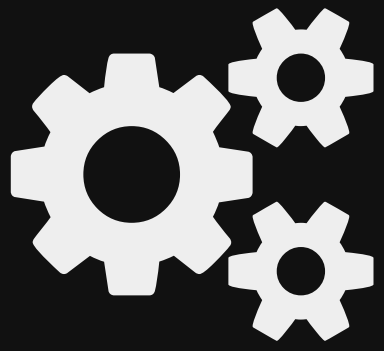
More than 100TB per day of compressed data

“Mesos is the cornerstone of our elastic compute infrastructure – it's how we build all our new services and is critical for Twitter's continued success at scale. It's one of the primary keys to our data center efficiency.” – Chris Fry, SVP of Engineering at Twitter


airbnb
ebay
vimeo
OpenTable
shopify
HubSpot
ignidata
igniting business with data
PayPal
xogito
...radical thinking...
DueDil
Atlassian
CLOUD PHYSICS
device scape
mesosphere



Mesoswhat?

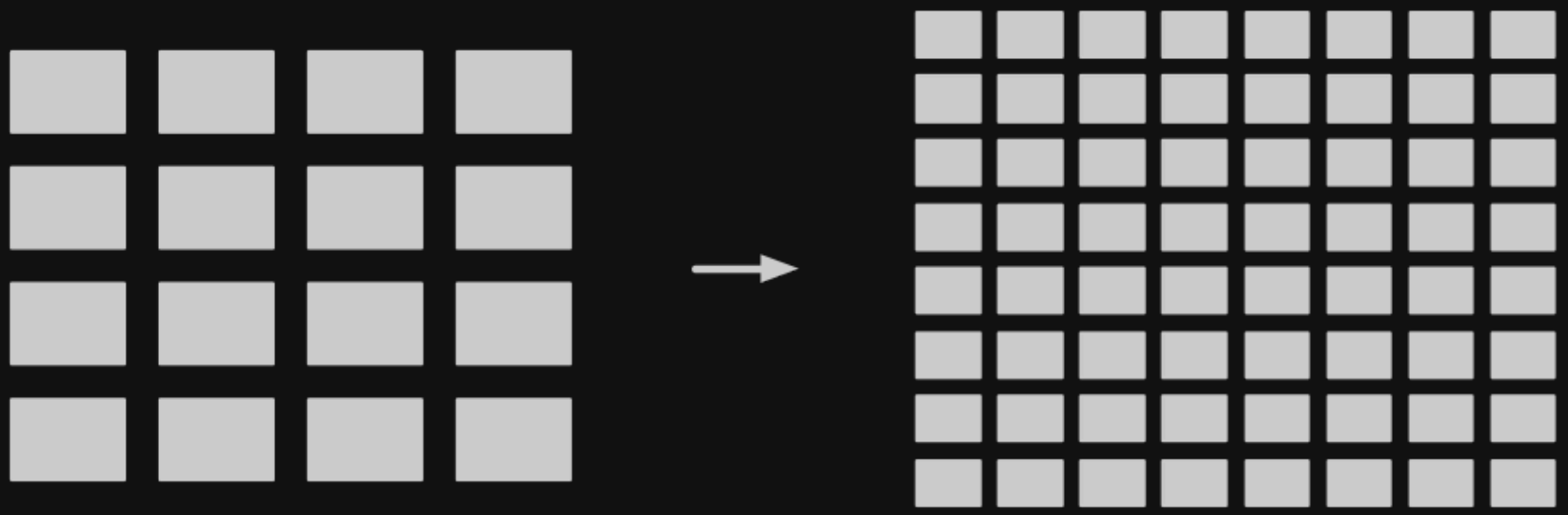


Marathon

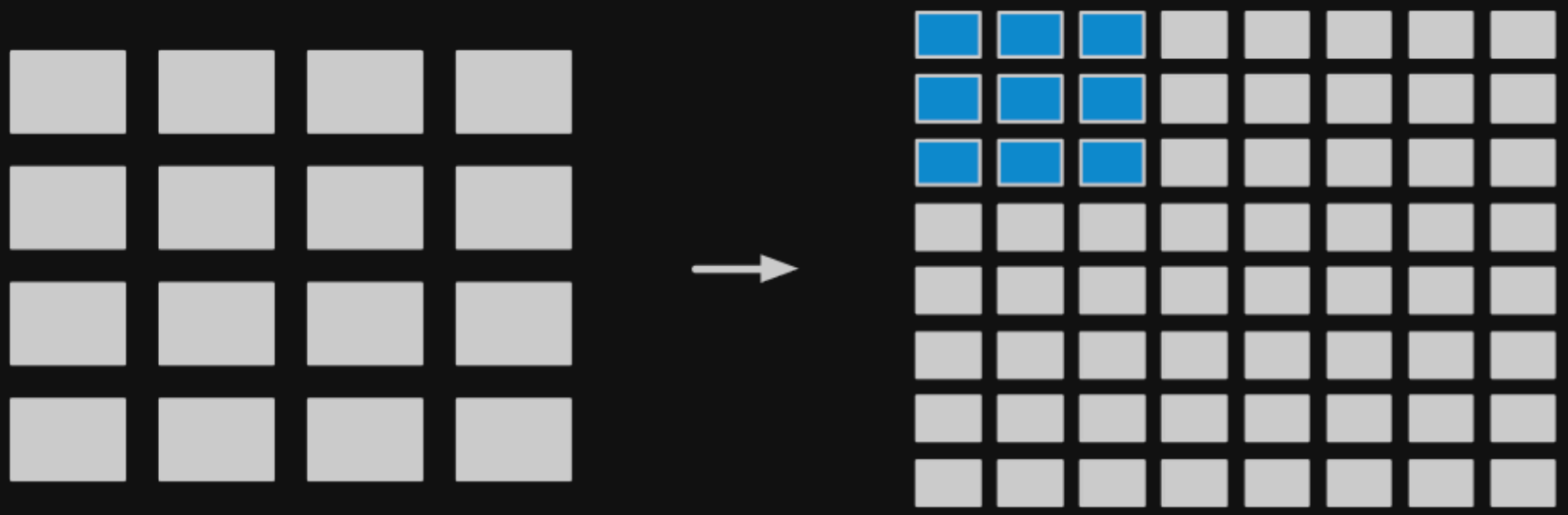
Chronos

Demo`s!

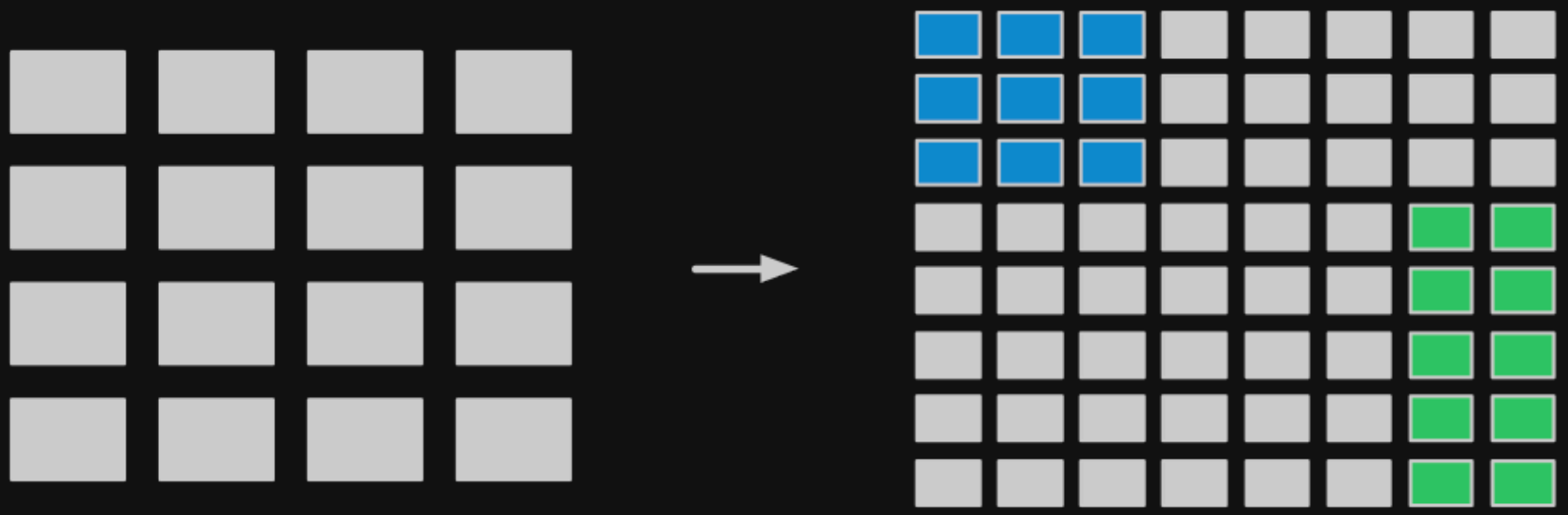
Status quo is static partitioning and use of virtual machines



Add some virtual machines



Provision Hadoop

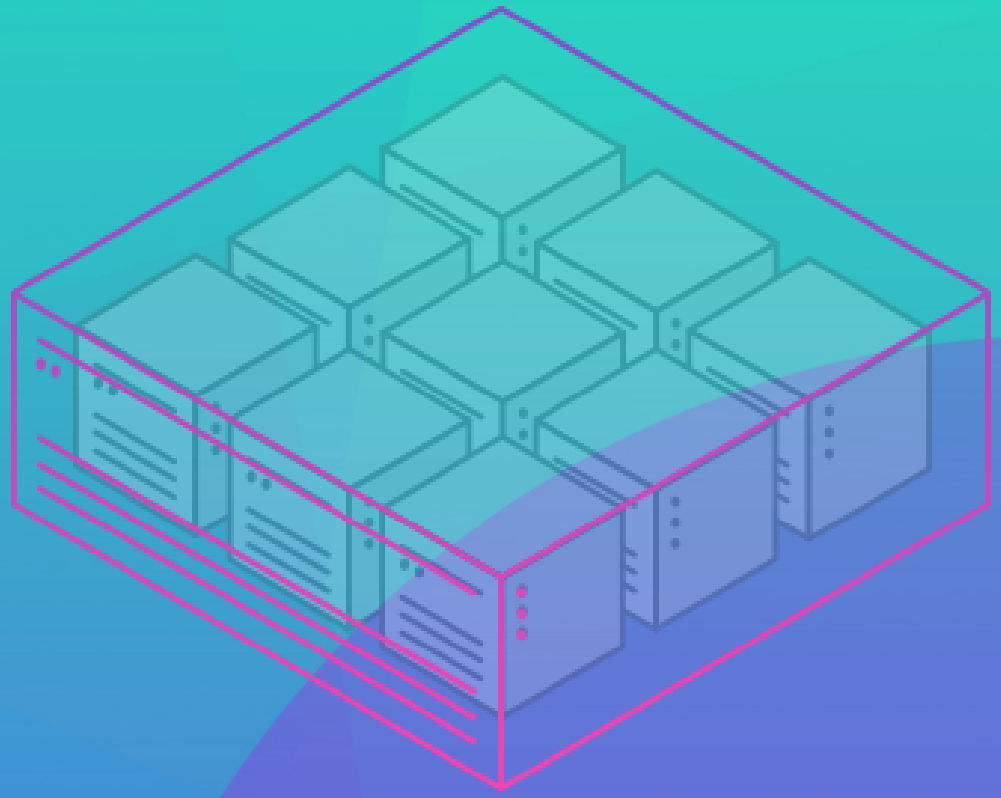


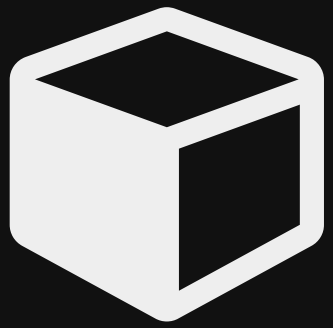
Provision a web service

Moar data, moar Hadoop

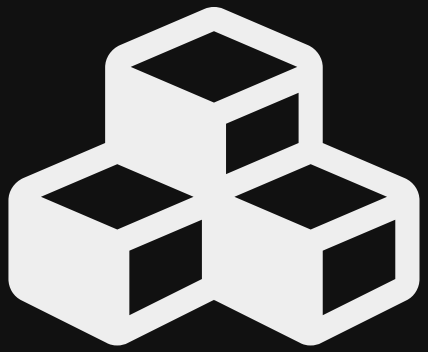
Mesos let us treat a cluster of nodes...

As one big computer



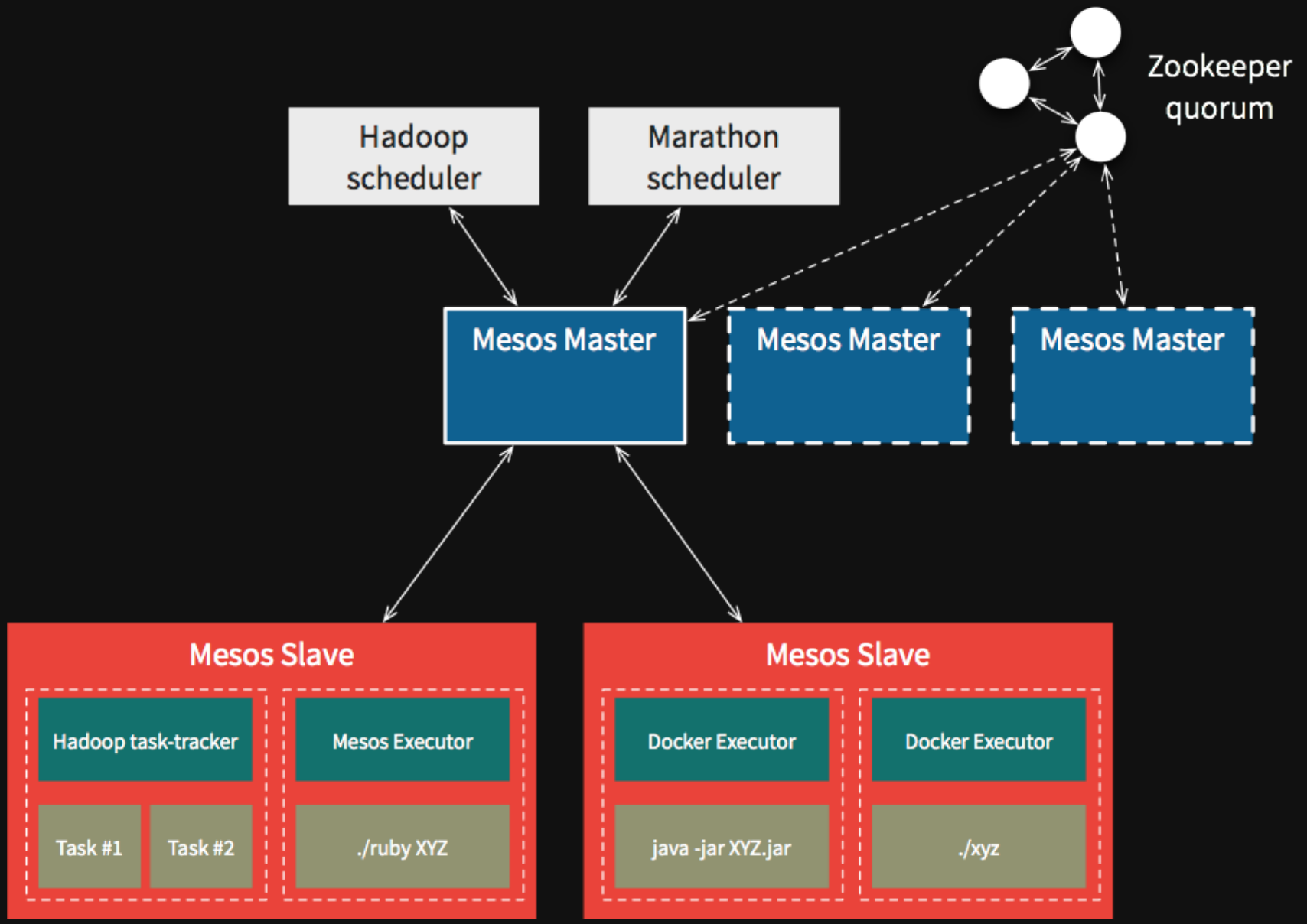


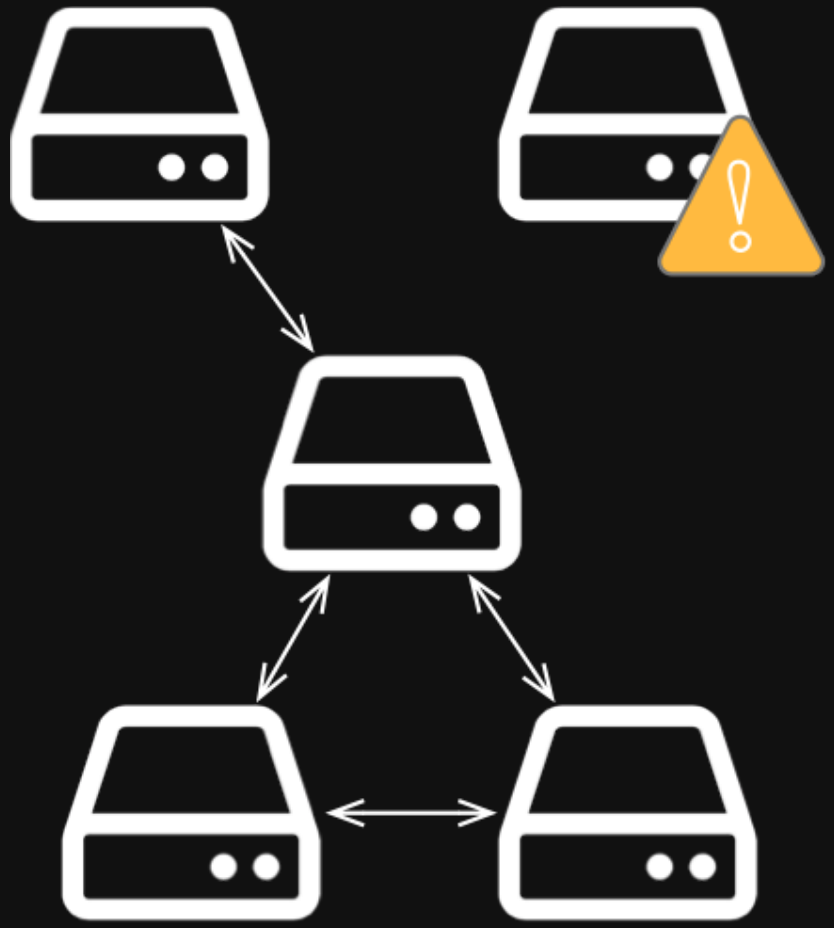
Not as individual machines



Not as VMs


But as computational resources
like cores, memory, disks, etc.

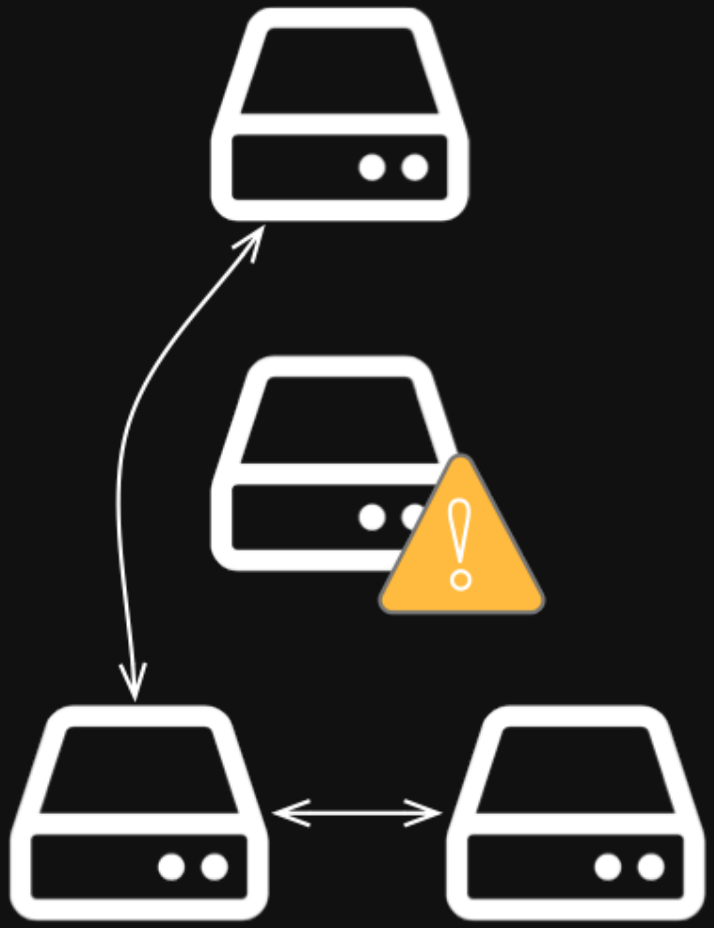

Zookeeper quorum
Hadoop scheduler
Marathon scheduler
Mesos Master
Mesos Master
Mesos Master
Mesos Slave
Hadoop task-tracker
Mesos Executor
Task #1
Task #2
./ruby XYZ
Mesos Slave
Docker Executor
Docker Executor
java -jar XYZ.jar
./xyz



Framework

Masters

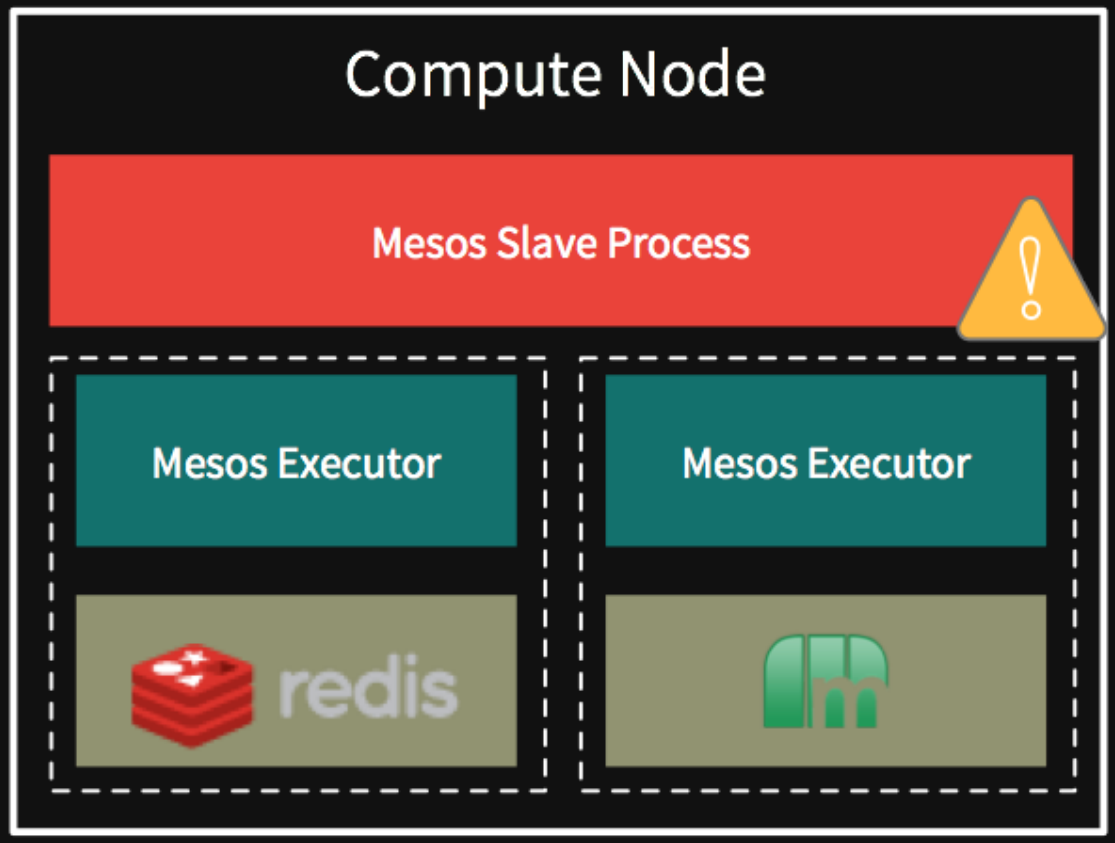
Tasks keep running!



Framework

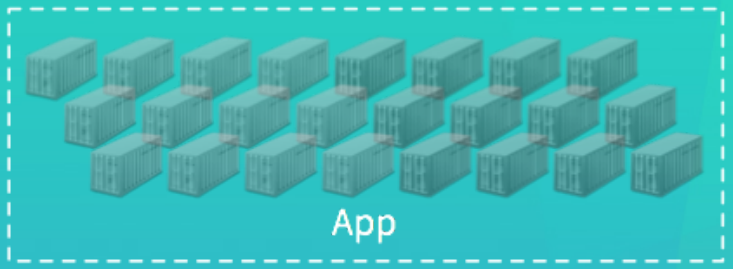
Masters

Tasks keep running!



Tasks keep running!

Mesos for all the things



Marathon | Kubernetes | Cassandra | Spark | Hadoop

Mesosphere Mesos Distribution

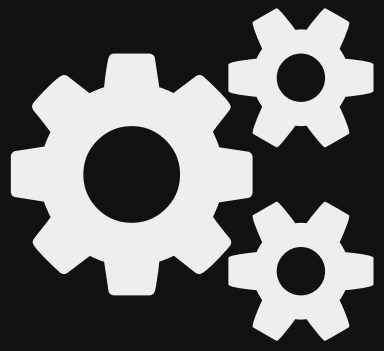
Linux | Linux | Linux | Linux | Linux | Linux | Linux | Linux | Linux | Linux

Mesos is...

- Open Source Apache project
- Cluster Resource Manager
- Scalable to 10,000s of nodes
- Fault-tolerant, battle-tested
- An SDK for distributed apps

The Mesos ecosystem is growing

Mesoswhat?



Marathon

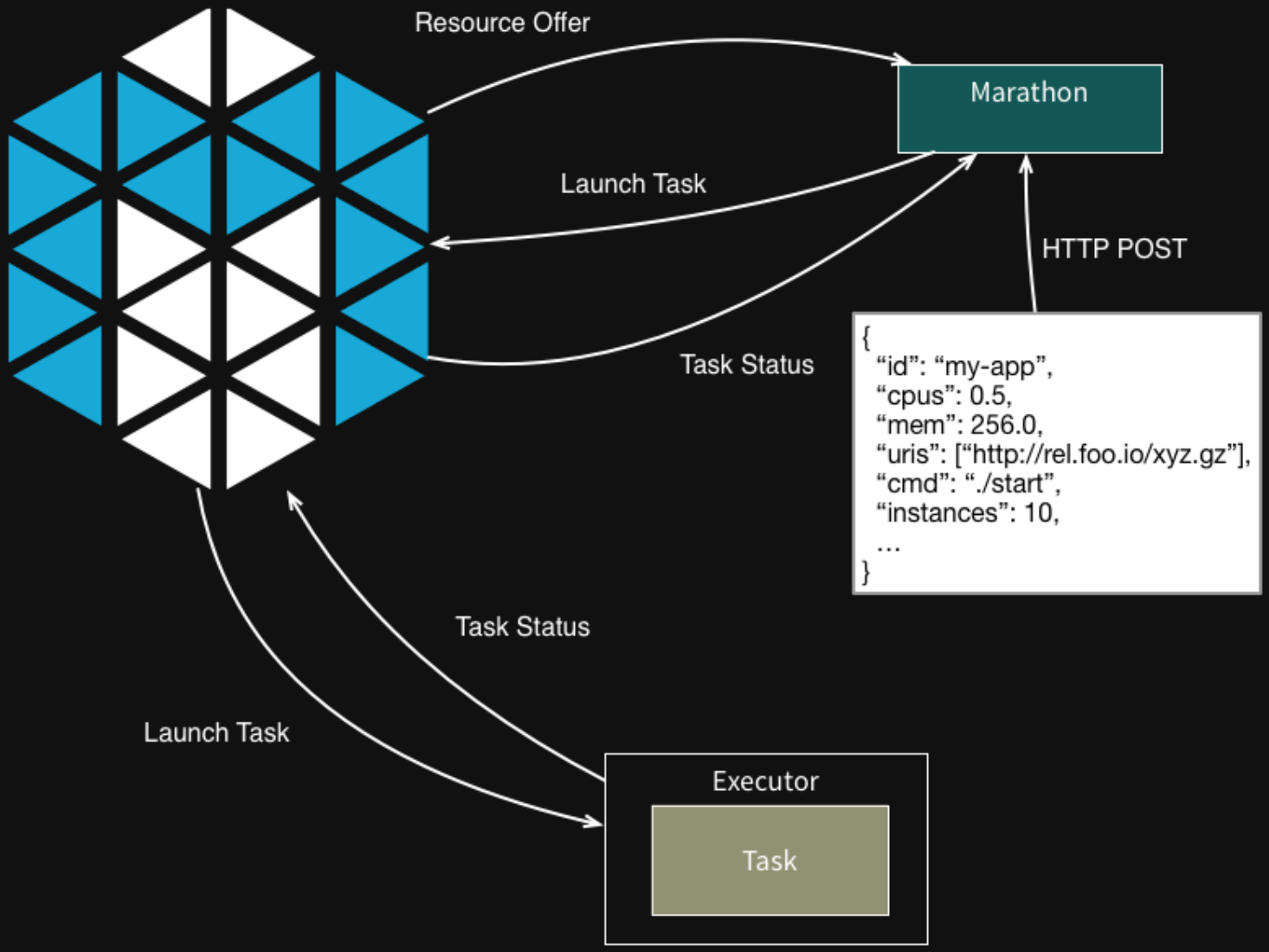
Chronos

Demo!

Say hi to Marathon

a self-serve interface to your cluster

distributed "init" for long-running services



a private fault-tolerant PaaS

a container orchestration platform

Marathon does it!

- Start, stop, scale, update apps
- Nice web interface, API
- Highly available, no SPoF
- Native Docker support
- Pluggable event bus
- Rolling deploy / restart
- Application health checks
- Artifact staging

Service Discovery

Set environment variables

Read config from device (rsync'ed to fs)

Read from K-V store

Use DNS

HAProxy works pretty well

Marathon ❤ REST

POST /v2/apps

GET /v2/apps

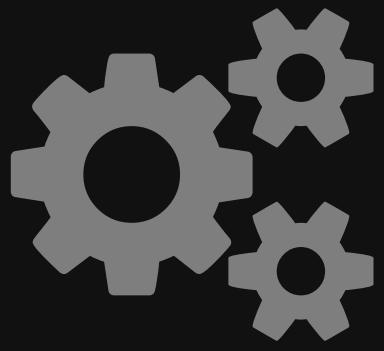
PUT /v2/apps/{appId}

GET /v2/apps/{appId}/tasks

DELETE /v2/apps/{appId}/tasks/{taskId}

...

Mesoswhat?

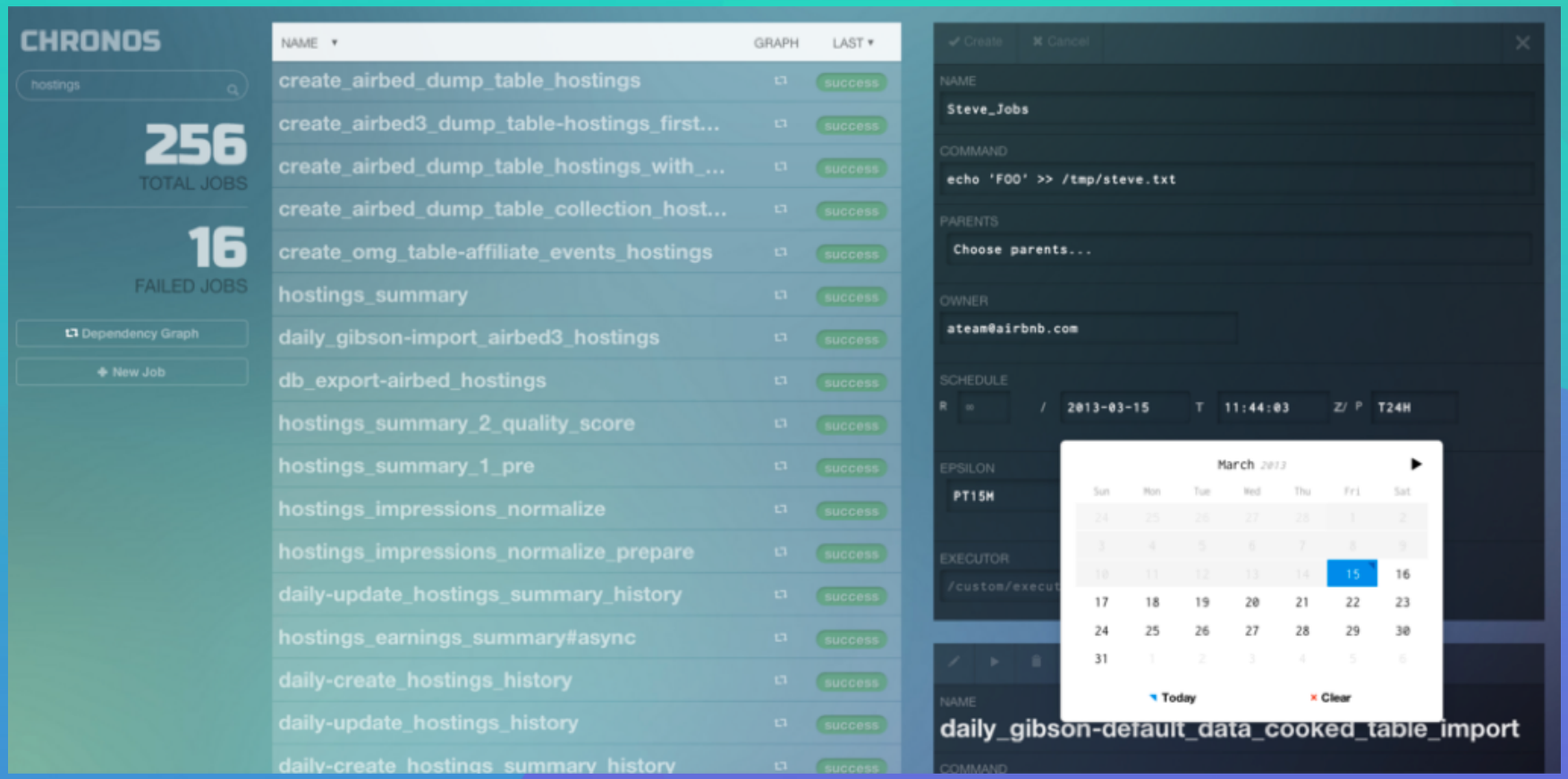


Marathon

Chronos

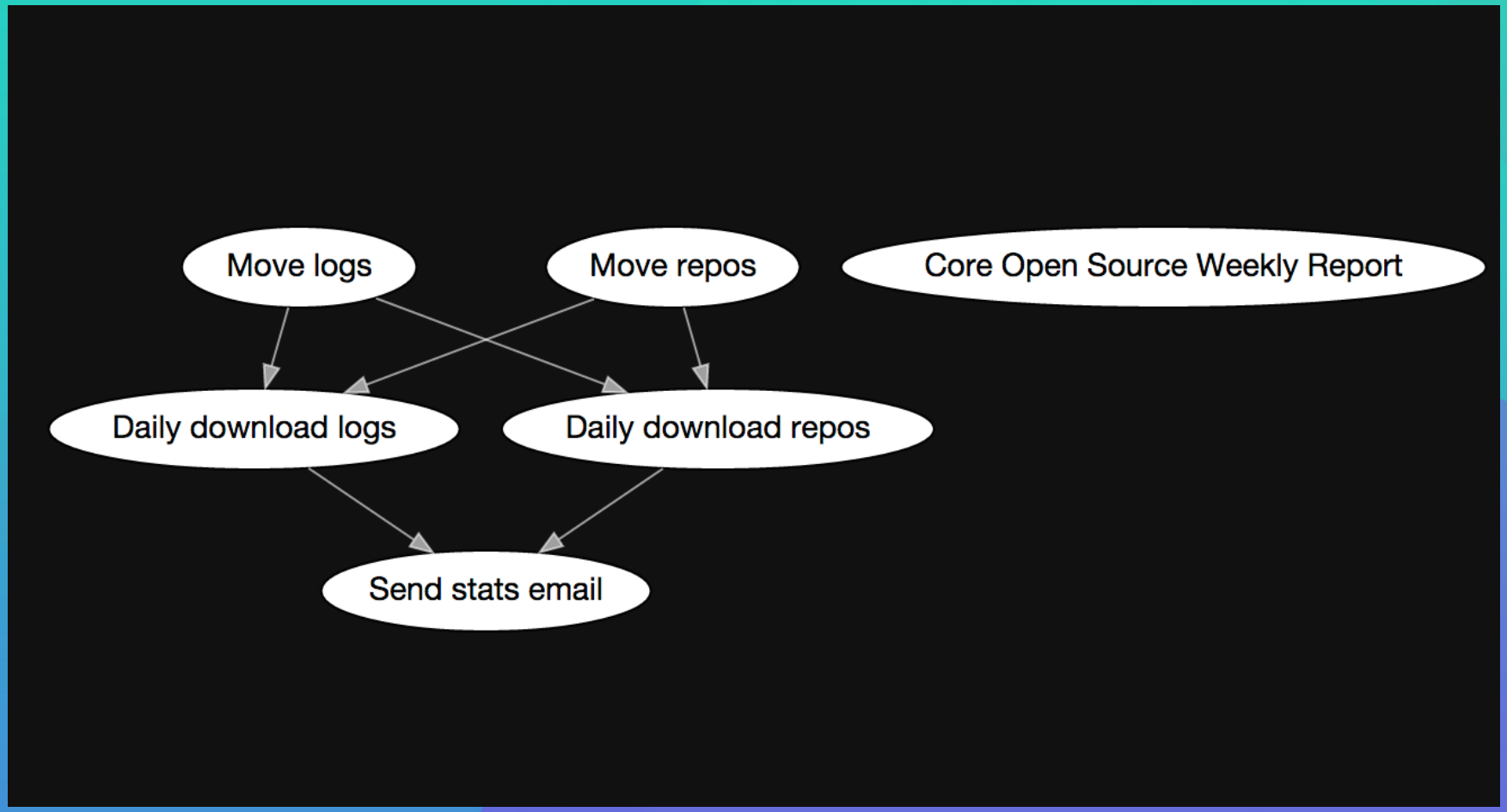
Demo!

Introducing Chronos



a scheduler for batch and one-off jobs

Distribute a graph of jobs



Dependency graph for execution

Features

- Distributed job scheduler
- Web interface, API
- Highly available, no SPoF
- Native Docker support
- Easy scheduling with repeating intervals

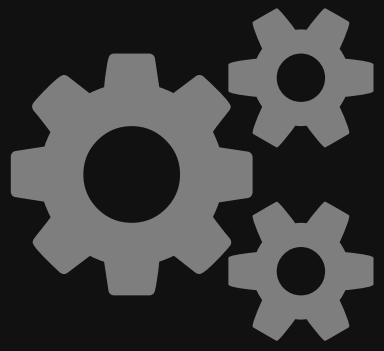
Chronos ♥ REST

```
PUT chronos-node:8080/scheduler/job/job1
```

```
GET chronos-node:8080/scheduler/jobs
```

```
DELETE chronos-node:8080/scheduler/task/kill/job2
```

Mesoswhat?

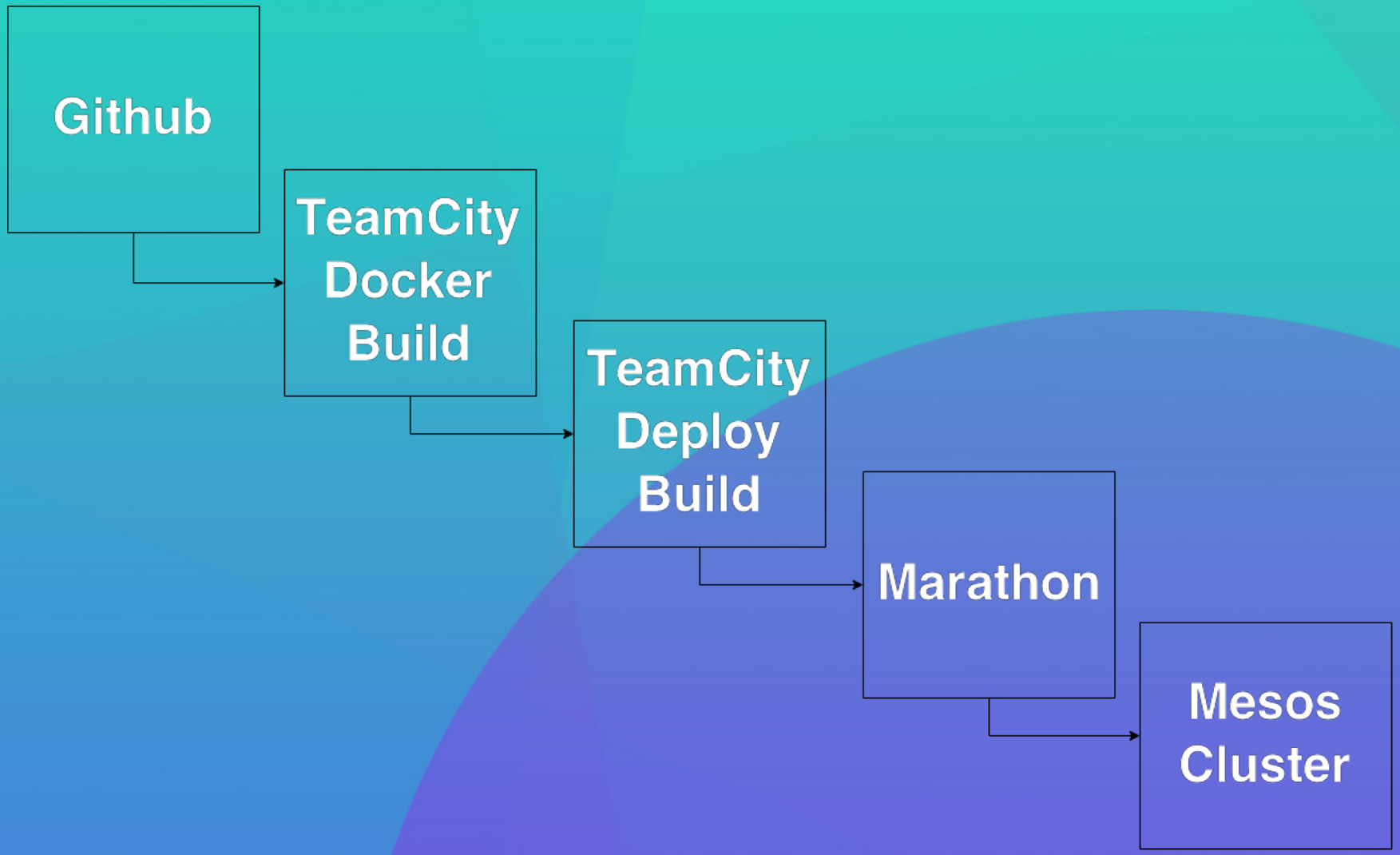


Marathon

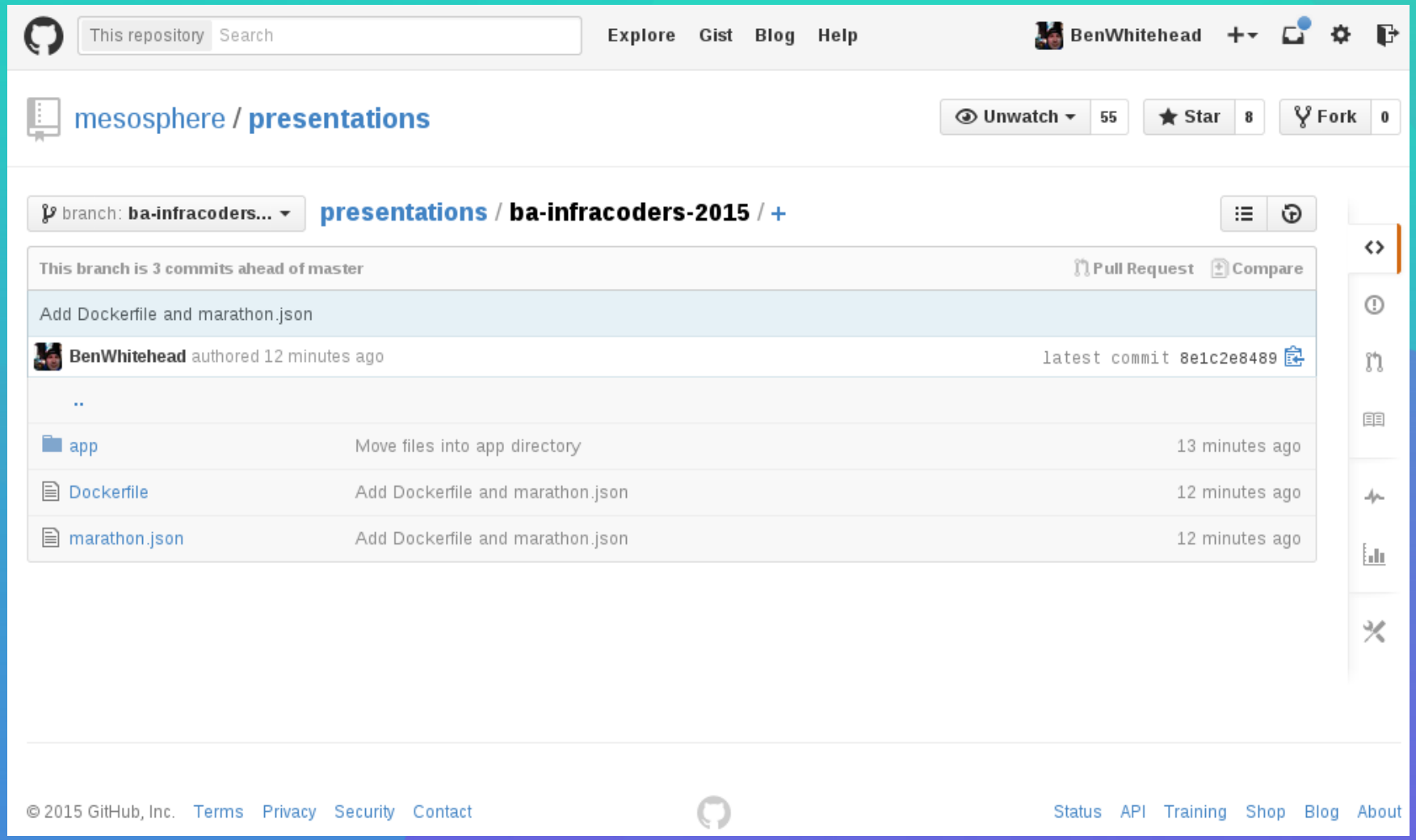
Chronos

Demo!

Continuous Delivery Pipeline



Code Base


This repository Search
Explore Gist Blog Help
BenWhitehead
mesosphere / presentations
Unwatch 55
Star 8
Fork 0
branch: ba-infracoders...
presentations / ba-infracoders-2015 / +
This branch is 3 commits ahead of master
Pull Request
Compare
Add Dockerfile and marathon.json
BenWhitehead authored 12 minutes ago
latest commit 8e1c2e8489
..
app
Move files into app directory
13 minutes ago
Dockerfile
Add Dockerfile and marathon.json
12 minutes ago
marathon.json
Add Dockerfile and marathon.json
12 minutes ago
© 2015 GitHub, Inc.
Terms
Privacy
Security
Contact
Status
API
Training
Shop
Blog
About

Docker Build

Steps

1. docker login
2. docker build
3. docker push
4. report tag

Artifacts

1. docker-tag
2. target/marathon.json

Dockerfile

```
FROM nginx
MAINTAINER Mesosphere support@mesosphere.io

EXPOSE 80

ADD app/ /usr/share/nginx/html
```

marathon.json

```
{
  "id": "/mesosphere/cd-demo-app",
  "instances": 1,
  "cpus": 1,
  "mem": 512,
  "container": {
    "type": "DOCKER",
    "docker": {
      "image": "mesosphere/cd-demo-app:$tag",
      "network": "BRIDGE",
      "portMappings": [
        { "servicePort": 28080, "containerPort": 80, "hostPort": 0, "protocol": "tcp" },
        { "servicePort": 28443, "containerPort": 443, "hostPort": 0, "protocol": "tcp" }
      ]
    }
  },
  "healthChecks": [
    {
      "gracePeriodSeconds": 120,
      "intervalSeconds": 30,
      "maxConsecutiveFailures": 3,
      "path": "/"
```

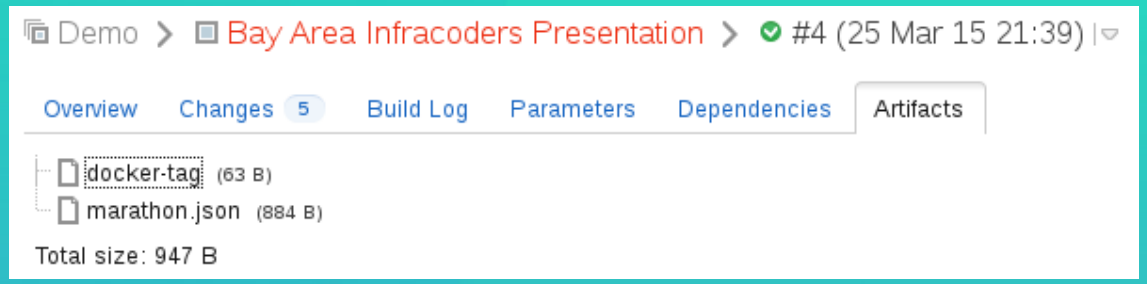
Report Tag

```
mkdir -p target

echo %TAG% > target/docker-tag

cat marathon.json | \
  jq '.container.docker.image |= "%DOCKER_IMAGE%:%TAG%"' > target/marathon.json
```

Artifacts


Demo
Bay Area Infracoders Presentation
#4 (25 Mar 15 21:39)
Overview
Changes 5
Build Log
Parameters
Dependencies
Artifacts
docker-tag (63 B)
marathon.json (884 B)
Total size: 947 B

Deploy

1. http PUT http://marathon/v2/apps < marathon.json
2. Send slack message

Slack Message

```
echo '{
  "username"   :"%SLACK_USERNAME%",
  "channel"    :"%SLACK_CHANNEL%",
  "text"       :"%SLACK_MESSAGE%",
  "icon_emoji" :"%SLACK_EMOJI%",
  "mrkdwn"     :%SLACK_MARKDOWN%
}' | http --print=HhBb --json POST %SLACK_WEBHOOK_URL%
```

Build Parameters

Parameters (11) edit »

Configuration parameters:

Name	Value
MARATHON_HOST	[redacted]
MARATHON_JSON_FILE	marathon.json
MARATHON_JSON_TYPE	apps
MARATHON_PORT	8080
SCHEME	http
SLACK_CHANNEL	#demo
SLACK_EMOJI	:teamcity:
SLACK_MARKDOWN	true
SLACK_MESSAGE	Heads Up! %teamcity.build.triggeredBy% just deployed <https://teamcity.mesosphere.io/viewLog.html?buildId=%teamcity.build.id%&tab=buildResultsDiv&buildTypeId=%system.teamcity.buildType.id%|%system.teamcity.projectName% :: %system.teamcity.buildConfName% #%build.counter%>
SLACK_USERNAME	TeamCity
SLACK_WEBHOOK_URL	[redacted]

Roll Out

Marathon will now begin rolling out[1] the updated version of the application and your team has just been notified of the deployment.

[1] https://mesosphere.github.io/marathon/docs/deployments.html#rolling-restarts

Run a Docker container

```
http -v POST http://10.53.20.188:8080/v2/apps @app-ruby.json
```

```
{
  "container": {
    "type": "DOCKER",
    "docker": {
      "image": "superguenter/demo-app"
    }
  },
  "cmd": "rails server -p $PORT",
  "id": "rails-demo",
  "instances": 1,
  "cpus": 0.01,
  "mem": 256,
  "ports": [3000]
}
```

Scale!

```
http -v PUT http://10.53.20.188:8080/v2/apps/demo @scale-app.json
```

```
{
  "instances": 3
}
```

Deploy a new Python version

```
http -v PUT http://10.53.20.188:8080/v2/apps/demo @app-python.json
```

```
{
  "container": {
    "type": "DOCKER",
    "docker": {
      "image": "superguenter/demo-app"
    }
  },
  "cmd":  "python -m SimpleHTTPServer $PORT",
  "id": "demo",
  "instances": 3,
  "cpus": 0.01,
  "mem": 256,
  "ports": [3000]
}
```

Use HTTP healthchecks

```
http -v POST http://10.53.20.188:8080/v2/apps @app-with-healthcheck.json
```

```
{
  "container": {
    "type": "DOCKER",
    "docker": {
      "image": "superguenter/demo-app"
    }
  },
  "cmd": "rails server -p $PORT",
  "id": "healthcheck-demo",
  "instances": 1,
  "cpus": 0.01,
  "mem": 256,
  "ports": [3000],
  "healthChecks": [
    {
      "path": "/",
      "portIndex": 0,
      "protocol": "HTTP",
      "gracePeriodSeconds": 30,
      "intervalSeconds": 30,
      "timeoutSeconds": 30,
```

Add a UNIQUE hostname constraint

```
http -v POST http://10.53.20.188:8080/v2/apps @app-with-constraint.json
```

```
{
  "container": {
    "type": "DOCKER",
    "docker": {
      "image": "superguenter/demo-app"
    }
  },
  "cmd": "python -m SimpleHTTPServer $PORT",
  "id": "constraint-demo",
  "instances": 6,
  "cpus": 0.01,
  "mem": 256,
  "ports": [3000],
  "healthChecks": [
    {
      "path": "/",
      "portIndex": 0,
      "protocol": "HTTP"
    }
  ],
  "constraints": [
```

Thanks!

Come and talk to us

P.S., we're hiring!

Get Mesosphere packages: mesosphere.io/downloads

Read about Marathon: github.com/mesosphere/marathon

Read about Chronos: github.com/mesos/chronos

Try out Mesosphere on GCE: google.mesosphere.io

Come work with us: mesosphere.io/jobs