Marathon

A self-serve interface to your cluster

Connor Doyle @ Mesosphere, Inc.

We are Mesosphere

Overview

What is Mesos?

Intro to Marathon

Upcoming Features

Mesos is...

a top-level Apache project

a cluster resource broker

scalable, fault-tolerant, battle-tested

an SDK for distributed apps

Why Mesos?

Static partitioning doesn't scale

Use dynamic allocation instead

Cluster scheduling is a hard problem

NP-hard in fact!

Monolithic schedulers need context

Idea: the "Resource Offer"

Centralize accounting & allocation

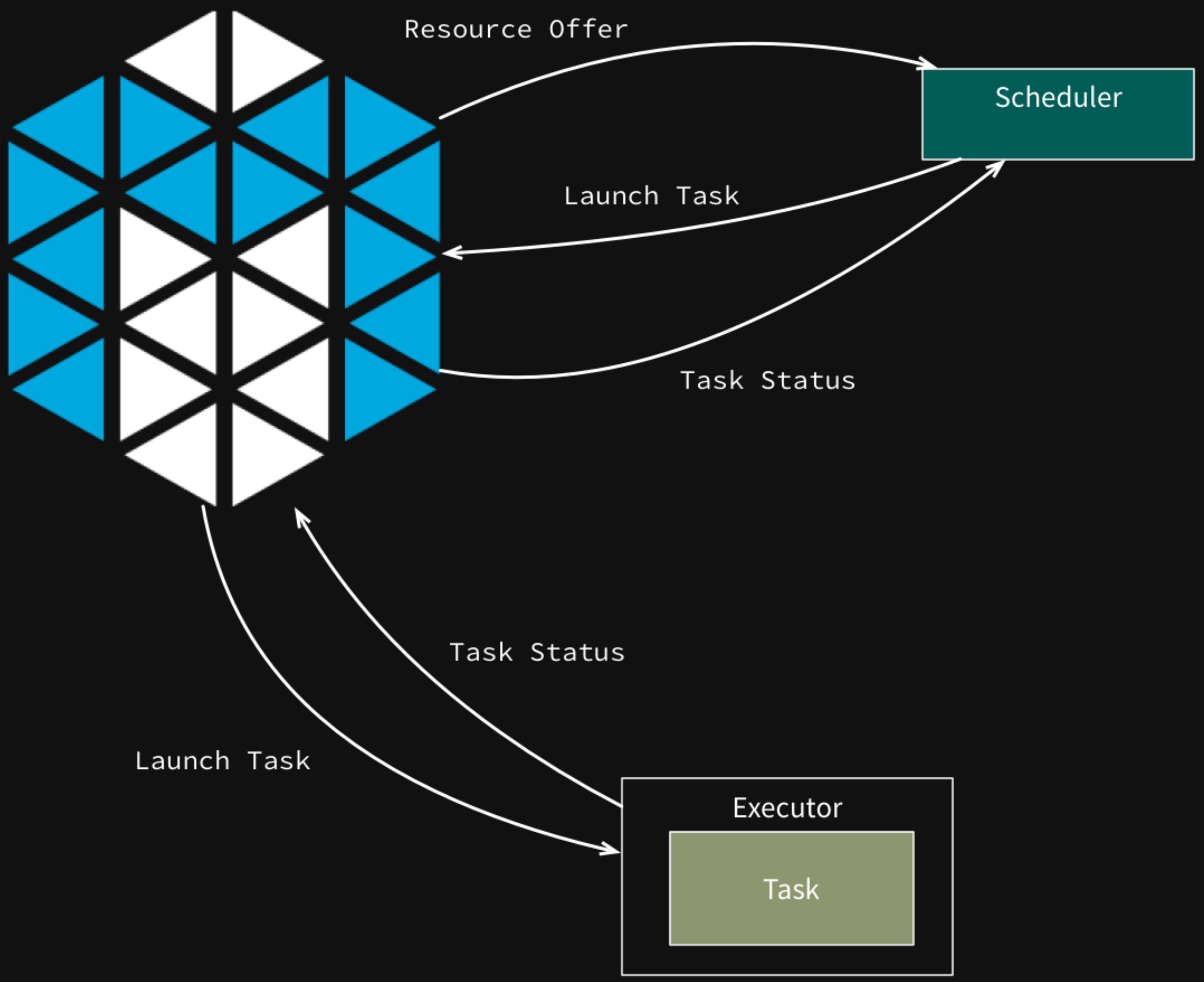
Push some logic down to apps

High Level View of Mesos

Framework = scheduler + executor

Schedulers get resource offers

Executors run tasks


Resource Offer
Scheduler
Launch Task
Task Status
Task Status
Launch Task
Executor
Task

Say hi to Marathon

A self-serve interface to your cluster

Distributed "init" for long-running services

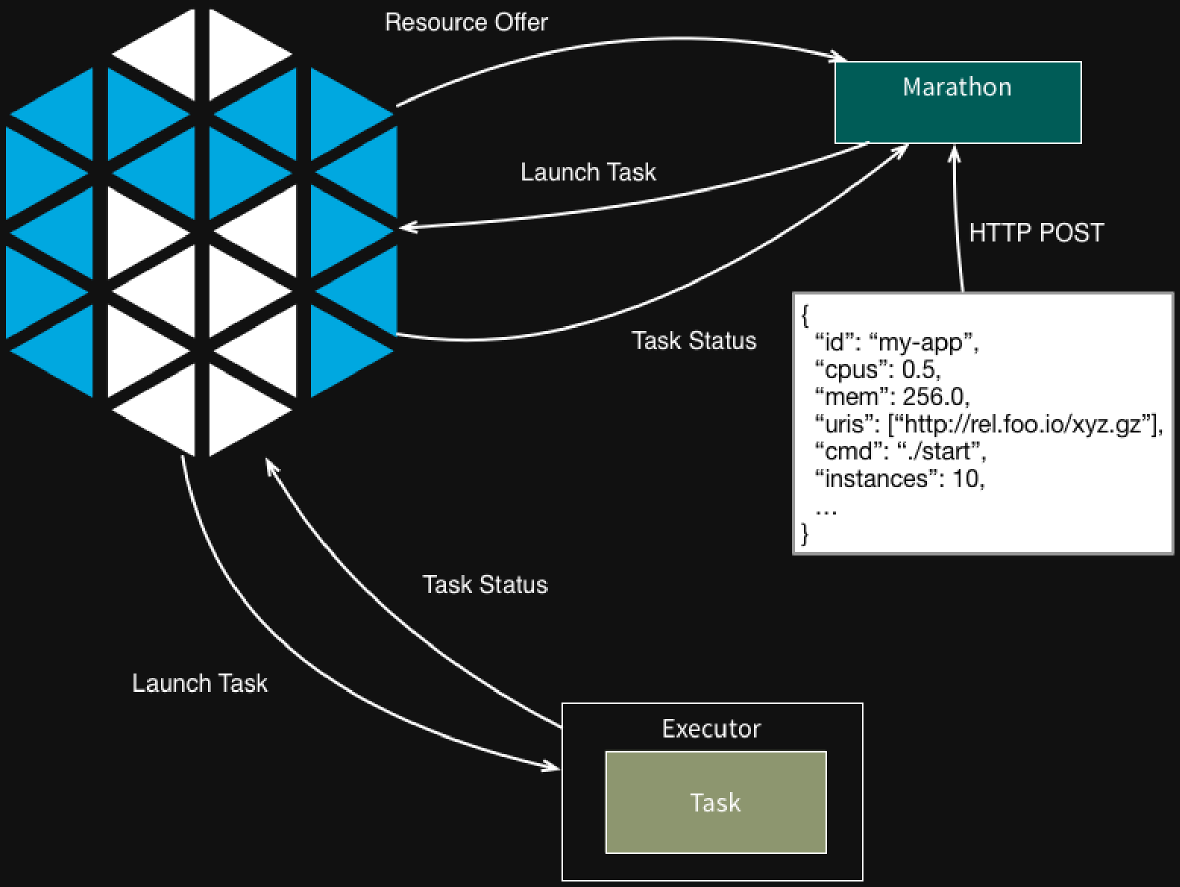
A private fault-tolerant Paas

Marathon Concepts

An app describes a task

A task is an instance of an app

Marathon creates tasks for apps



Resource Offer

Marathon

Launch Task

HTTP POST

Task Status

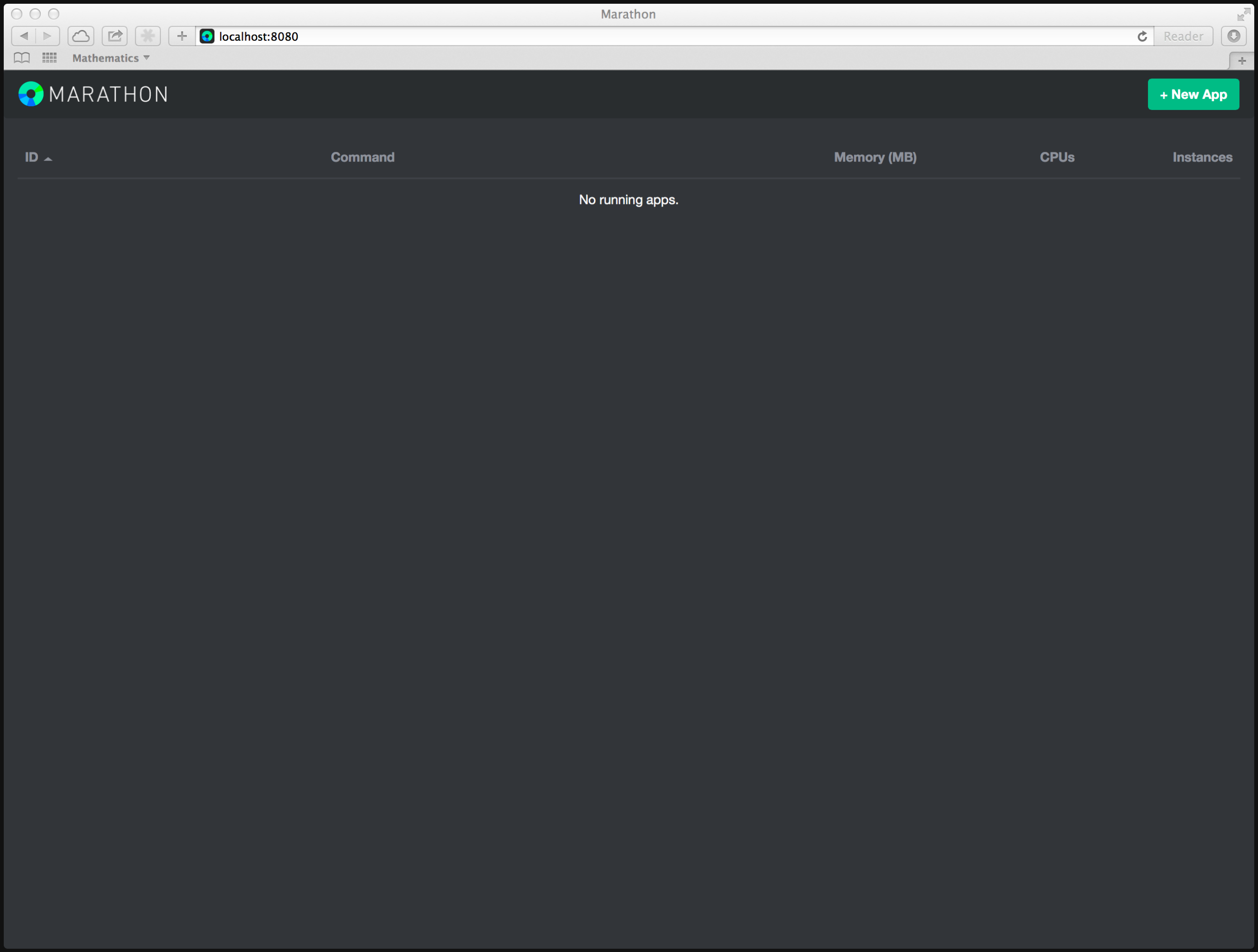
```
{
  “id”: “my-app”,
  “cpus”: 0.5,
  “mem”: 256.0,
  “uris”: [“http://rel.foo.io/xyz.gz”],
  “cmd”: “./start”,
  “instances”: 10,
  …
}
```

Task Status

Launch Task

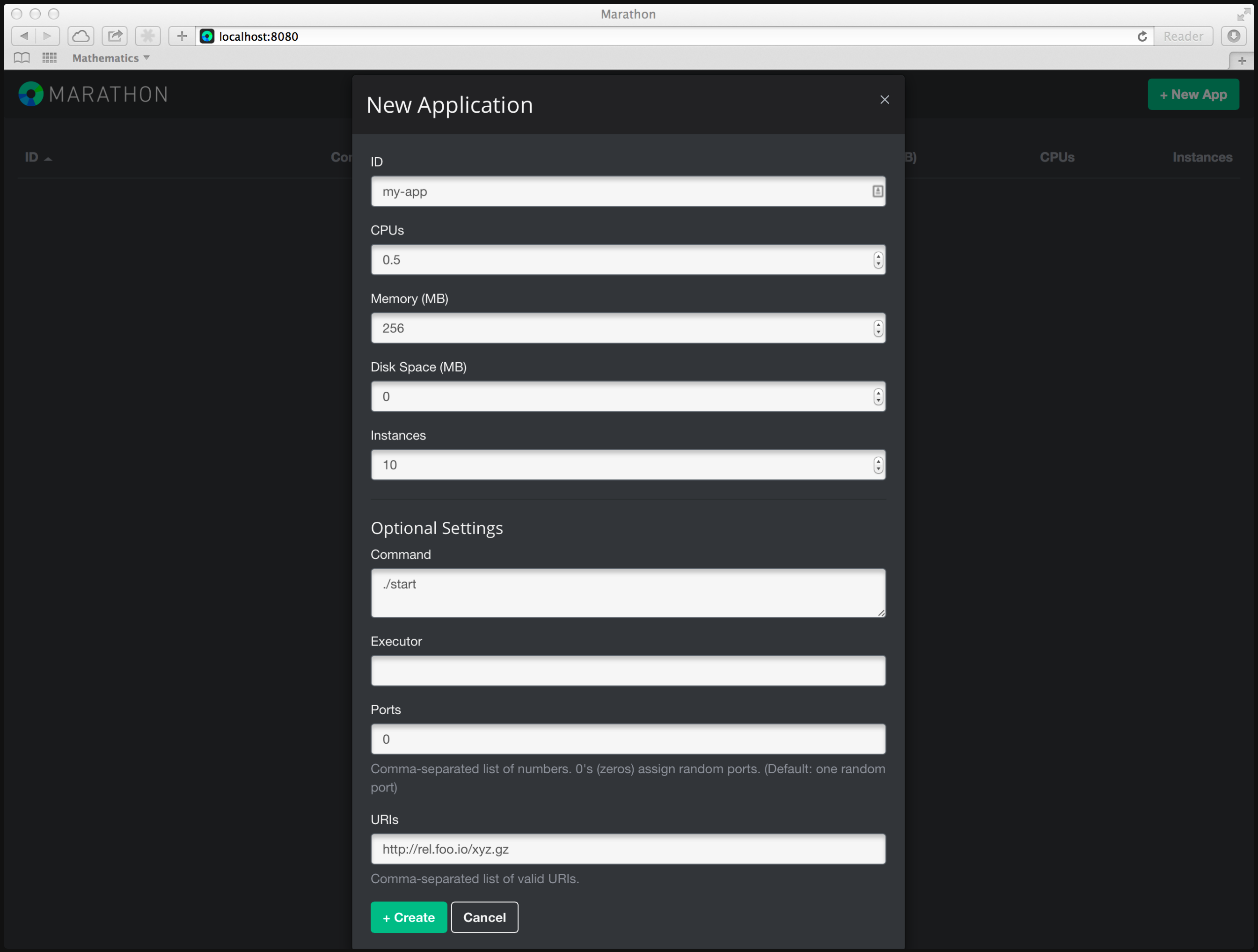
Executor

Task



ID	Command	Memory (MB)	CPUs	Instances

No running apps.



New Application

ID
my-app

CPUs
0.5

Memory (MB)
256

Disk Space (MB)
0

Instances
10

Optional Settings

Command
./start

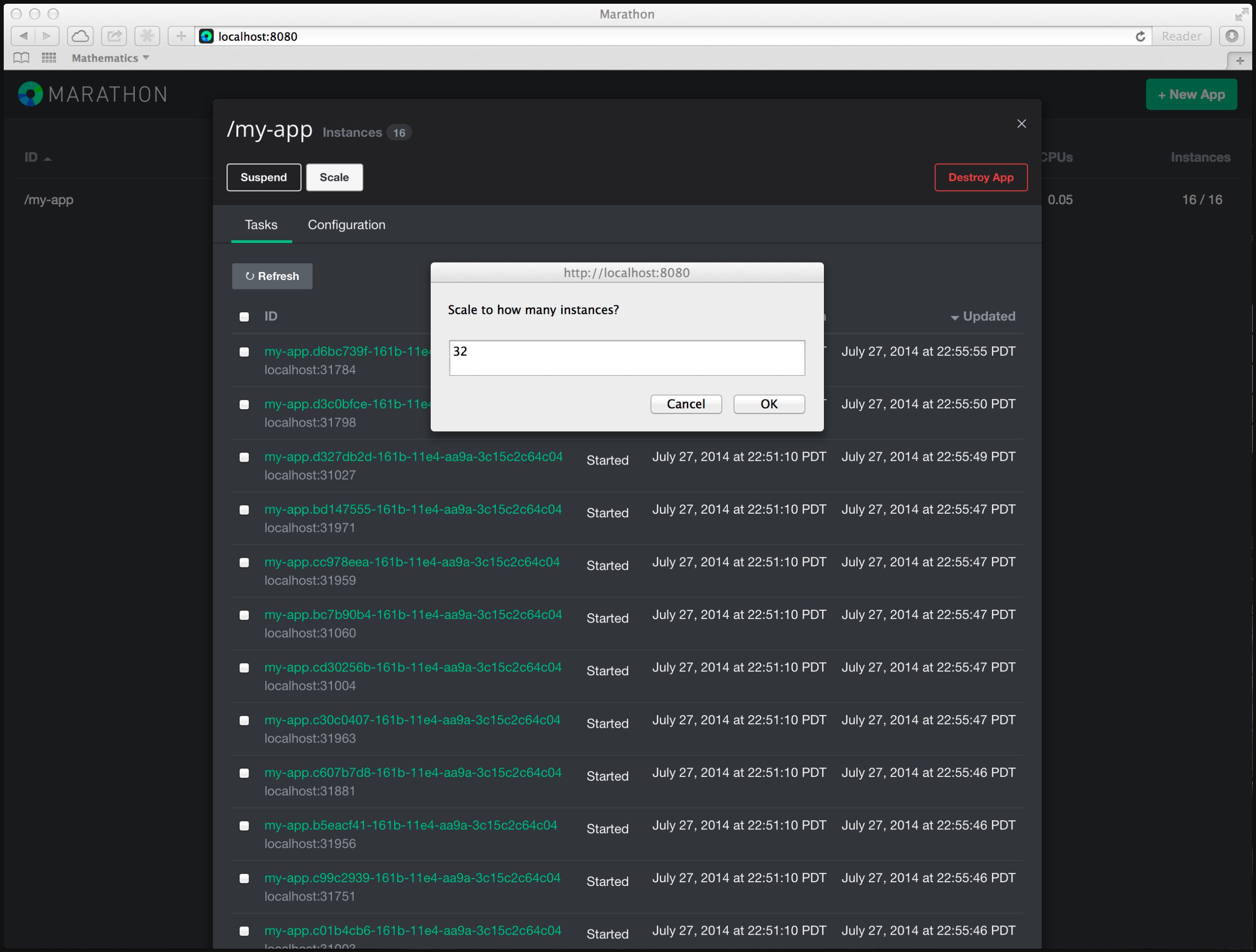
Executor

Ports
0

Comma-separated list of numbers. 0's (zeros) assign random ports. (Default: one random port)

URIs
http://rel.foo.io/xyz.gz

Comma-separated list of valid URIs.



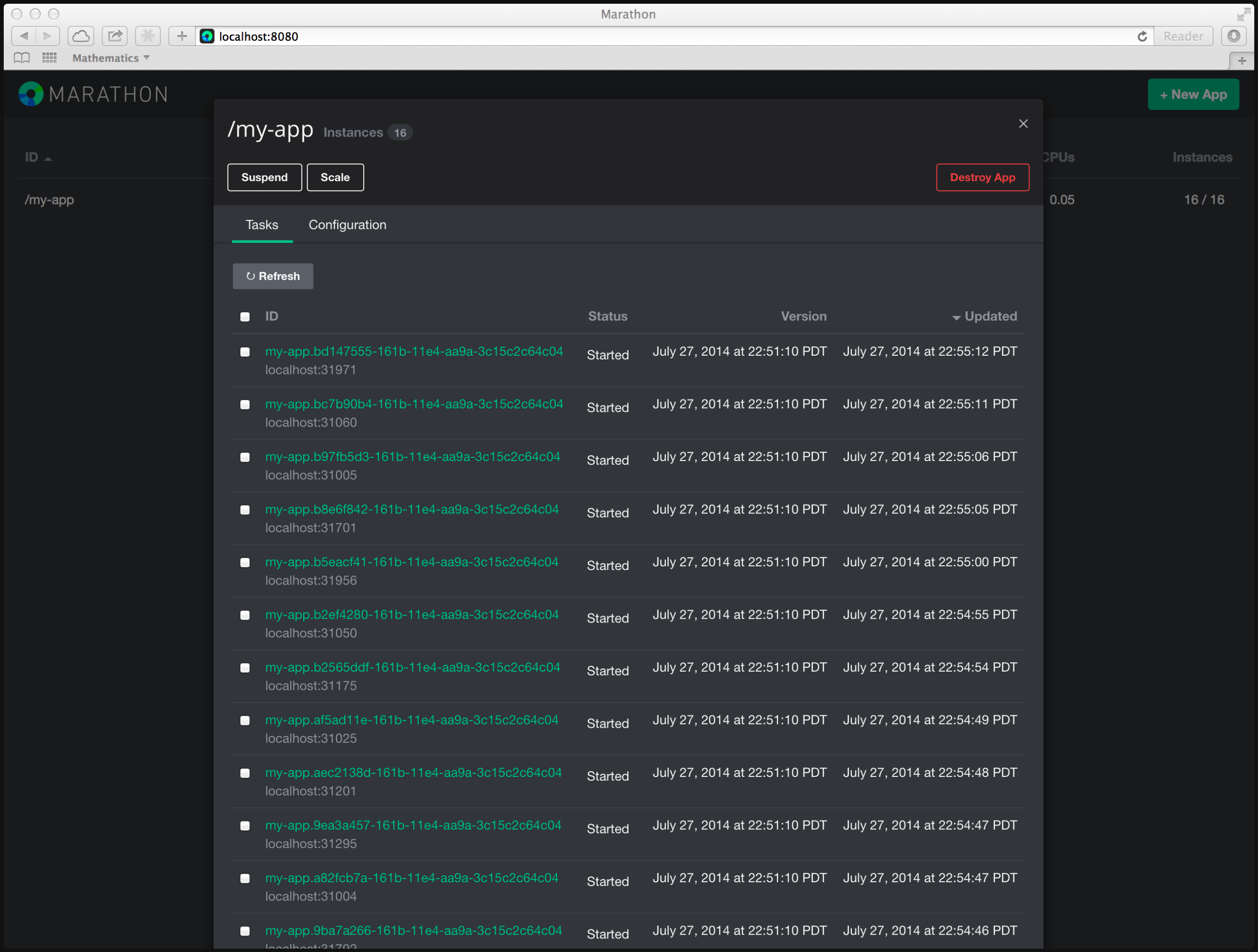
http://localhost:8080

Scale to how many instances?

32

Cancel OK

ID			Updated
my-app.d6bc739f-161b-11e localhost:31784			July 27, 2014 at 22:55:55 PDT
my-app.d3c0bfce-161b-11e localhost:31798			July 27, 2014 at 22:55:50 PDT
my-app.d327db2d-161b-11e4-aa9a-3c15c2c64c04 localhost:31027	Started	July 27, 2014 at 22:51:10 PDT	July 27, 2014 at 22:55:49 PDT
my-app.bd147555-161b-11e4-aa9a-3c15c2c64c04 localhost:31971	Started	July 27, 2014 at 22:51:10 PDT	July 27, 2014 at 22:55:47 PDT
my-app.cc978eea-161b-11e4-aa9a-3c15c2c64c04 localhost:31959	Started	July 27, 2014 at 22:51:10 PDT	July 27, 2014 at 22:55:47 PDT
my-app.bc7b90b4-161b-11e4-aa9a-3c15c2c64c04 localhost:31060	Started	July 27, 2014 at 22:51:10 PDT	July 27, 2014 at 22:55:47 PDT
my-app.cd30256b-161b-11e4-aa9a-3c15c2c64c04 localhost:31004	Started	July 27, 2014 at 22:51:10 PDT	July 27, 2014 at 22:55:47 PDT
my-app.c30c0407-161b-11e4-aa9a-3c15c2c64c04 localhost:31963	Started	July 27, 2014 at 22:51:10 PDT	July 27, 2014 at 22:55:47 PDT
my-app.c607b7d8-161b-11e4-aa9a-3c15c2c64c04 localhost:31881	Started	July 27, 2014 at 22:51:10 PDT	July 27, 2014 at 22:55:46 PDT
my-app.b5eacf41-161b-11e4-aa9a-3c15c2c64c04 localhost:31956	Started	July 27, 2014 at 22:51:10 PDT	July 27, 2014 at 22:55:46 PDT
my-app.c99c2939-161b-11e4-aa9a-3c15c2c64c04 localhost:31751	Started	July 27, 2014 at 22:51:10 PDT	July 27, 2014 at 22:55:46 PDT
my-app.c01b4cb6-161b-11e4-aa9a-3c15c2c64c04 localhost:31003	Started	July 27, 2014 at 22:51:10 PDT	July 27, 2014 at 22:55:46 PDT



□	ID	Status	Version	Updated
□	my-app.bd147555-161b-11e4-aa9a-3c15c2c64c04 localhost:31971	Started	July 27, 2014 at 22:51:10 PDT	July 27, 2014 at 22:55:12 PDT
□	my-app.bc7b90b4-161b-11e4-aa9a-3c15c2c64c04 localhost:31060	Started	July 27, 2014 at 22:51:10 PDT	July 27, 2014 at 22:55:11 PDT
□	my-app.b97fb5d3-161b-11e4-aa9a-3c15c2c64c04 localhost:31005	Started	July 27, 2014 at 22:51:10 PDT	July 27, 2014 at 22:55:06 PDT
□	my-app.b8e6f842-161b-11e4-aa9a-3c15c2c64c04 localhost:31701	Started	July 27, 2014 at 22:51:10 PDT	July 27, 2014 at 22:55:05 PDT
□	my-app.b5eacf41-161b-11e4-aa9a-3c15c2c64c04 localhost:31956	Started	July 27, 2014 at 22:51:10 PDT	July 27, 2014 at 22:55:00 PDT
□	my-app.b2ef4280-161b-11e4-aa9a-3c15c2c64c04 localhost:31050	Started	July 27, 2014 at 22:51:10 PDT	July 27, 2014 at 22:54:55 PDT
□	my-app.b2565ddf-161b-11e4-aa9a-3c15c2c64c04 localhost:31175	Started	July 27, 2014 at 22:51:10 PDT	July 27, 2014 at 22:54:54 PDT
□	my-app.af5ad11e-161b-11e4-aa9a-3c15c2c64c04 localhost:31025	Started	July 27, 2014 at 22:51:10 PDT	July 27, 2014 at 22:54:49 PDT
□	my-app.aec2138d-161b-11e4-aa9a-3c15c2c64c04 localhost:31201	Started	July 27, 2014 at 22:51:10 PDT	July 27, 2014 at 22:54:48 PDT
□	my-app.9ea3a457-161b-11e4-aa9a-3c15c2c64c04 localhost:31295	Started	July 27, 2014 at 22:51:10 PDT	July 27, 2014 at 22:54:47 PDT
□	my-app.a82fcb7a-161b-11e4-aa9a-3c15c2c64c04 localhost:31004	Started	July 27, 2014 at 22:51:10 PDT	July 27, 2014 at 22:54:47 PDT
□	my-app.9ba7a266-161b-11e4-aa9a-3c15c2c64c04 localhost:31792	Started	July 27, 2014 at 22:51:10 PDT	July 27, 2014 at 22:54:46 PDT

Marathon does it!

Start, stop, scale, update apps

View running tasks

Kill an individual task

Nice UI, Nice API

Marathon does it!

Event bus

App versioning

Highly available, no SPoF

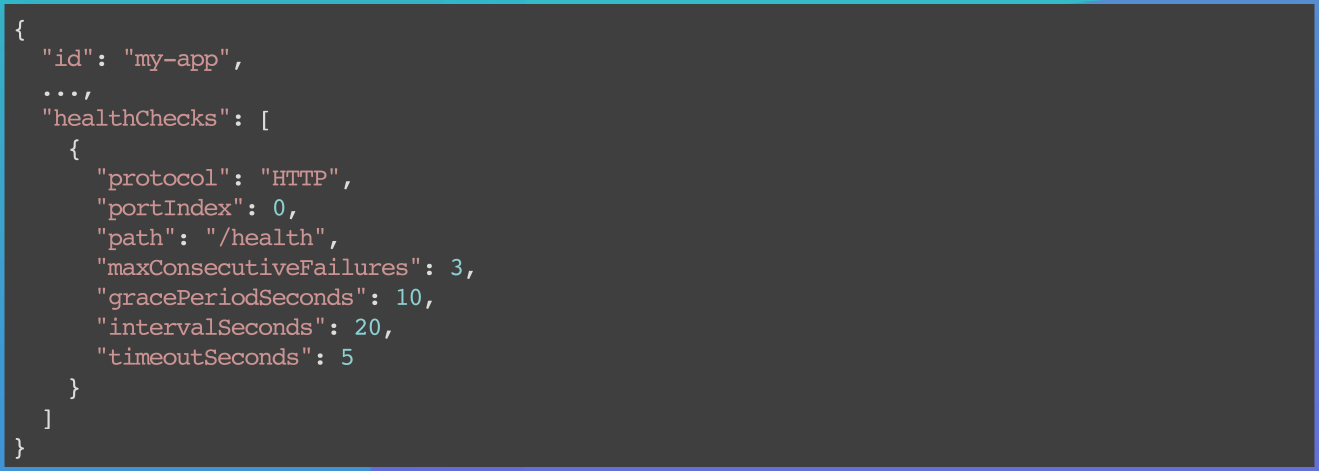
Marathon does it!

Placement constraints

```
// usage

{ "constraints": [["attribute", "OPERATOR", "value"]] }

// examples

{ "constraints": [["hostname", "UNIQUE"]] }

{ "constraints": [["rack_id", "CLUSTER", "rack-1"]] }

{ "constraints": [["rack_id", "GROUP_BY"]] }
```

Marathon does it!

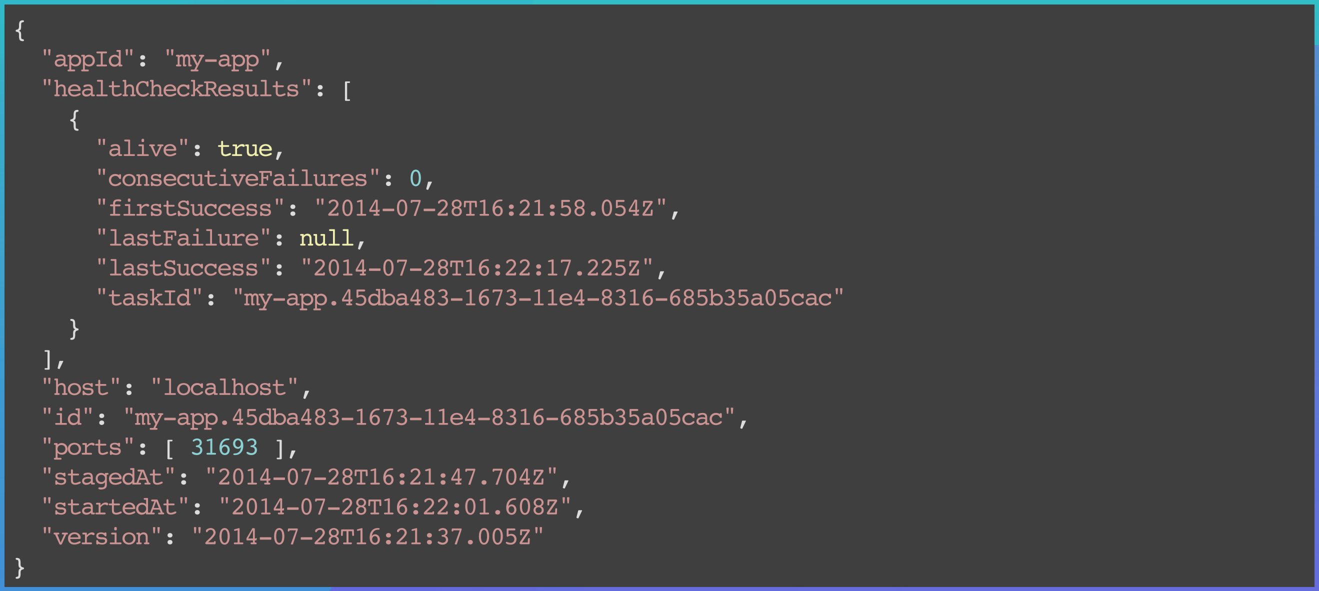
Task health checks (TCP, HTTP)



```
{
  "id": "my-app",
  ...,
  "healthChecks": [
    {
      "protocol": "HTTP",
      "portIndex": 0,
      "path": "/health",
      "maxConsecutiveFailures": 3,
      "gracePeriodSeconds": 10,
      "intervalSeconds": 20,
      "timeoutSeconds": 5
    }
  ]
}
```

Marathon does it!

Task health checks (TCP, HTTP)



```
{
  "appId": "my-app",
  "healthCheckResults": [
    {
      "alive": true,
      "consecutiveFailures": 0,
      "firstSuccess": "2014-07-28T16:21:58.054Z",
      "lastFailure": null,
      "lastSuccess": "2014-07-28T16:22:17.225Z",
      "taskId": "my-app.45dba483-1673-11e4-8316-685b35a05cac"
    }
  ],
  "host": "localhost",
  "id": "my-app.45dba483-1673-11e4-8316-685b35a05cac",
  "ports": [ 31693 ],
  "stagedAt": "2014-07-28T16:21:47.704Z",
  "startedAt": "2014-07-28T16:22:01.608Z",
  "version": "2014-07-28T16:21:37.005Z"
}
```

Get excited -- new features!

Rolling deploy / restart

Namespaced apps

Dependencies

Executor health checks

Artifact staging

Configurable exponential backoff

Marathon ❤ REST

POST /v2/apps
GET /v2/apps
PUT /v2/apps/{appId}
GET /v2/apps/{appId}/tasks
DELETE /v2/apps/{appId}/tasks/{taskId}
...

Service Discovery

Set environment variables

Read config from device (rsync'ed to fs)

Read from K-V store

Use DNS

HAProxy works pretty well

Tasks Resource

```
GET /v2/tasks HTTP/1.1
Accept: text/plain

HTTP/1.1 200 OK
Content-Type: text/plain
Server: Jetty(8.y.z-SNAPSHOT)
Transfer-Encoding: chunked

my-app   19385   localhost:31336  localhost:31364
my-app   11186   localhost:31337  localhost:31365
```

What about Docker?

Decouples dev from deployment

`docker build`

Now what? Deploy… somehow

Mesos to the rescue!

POST /v2/apps

```
{
  “id”: “cassandra",
  “container”: {
    “image”: “docker:///mesosphere/cassandra:2.0.6”,
    “options”: [“-v”, “/mnt:/mnt:rw”, “-e”, “CLUSTER_NAME=prod”]
  }
}
```

Existing support

External containerizer called Deimos

Calls out to Docker on task launch

For imminent release

Docker as a 1st class citizen

ContainerInfo vs DockerInfo?

One of these two will land (0.19.1 or 0.20.0 at the latest)

Thanks!

Come and talk to us