Sunil Shah

CONTAINERISED CONTINUOUS DELIVERY AT LOW COST AND WEB SCALE!

© 2016 Mesosphere, Inc. All Rights Reserved.

Introduction

OBJECTIVES

At the end of this talk, you should be able to:

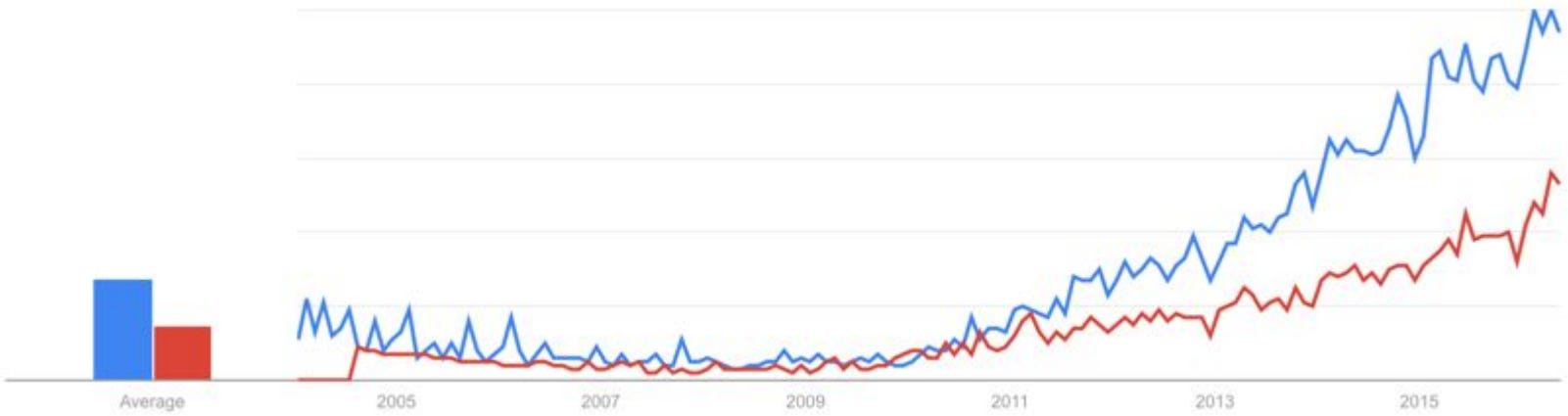
1. Understand how Apache Mesos & DC/OS, works
2. Deploy and configure a continuous deployment pipeline onto DC/OS

© 2016 Mesosphere, Inc. All Rights Reserved.

Introduction

WHY BOTHER?

Why is continuous delivery interesting now?



Google Trends for continuous delivery (blue) and continuous deployment (red)

© 2016 Mesosphere, Inc. All Rights Reserved.

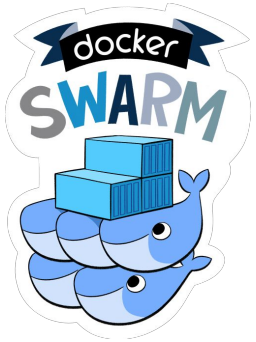
WHY BOTHER?

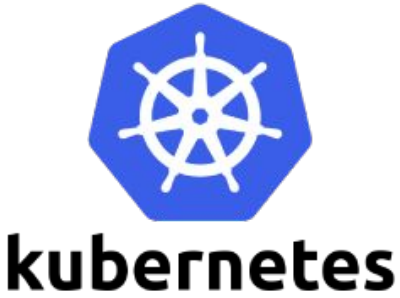
1. **It's much easier to get compute resources nowadays!**
   - Doesn't cost much (sometimes it's even free - just ask a graduate student)
   - **EVERY** platform and their dog has an API

© 2016 Mesosphere, Inc. All Rights Reserved.

WHY BOTHER?

1. It's much easier to get compute resources nowadays!
2. **It turns out getting sleep is good for you!**
   - The National Sleep Foundation recommends 7-9 hours of sleep per night
   - Container orchestrators take the manual pain out of waking up and rebooting an application (to varying degrees of success)
   - Let your devs dev and ops sleep!

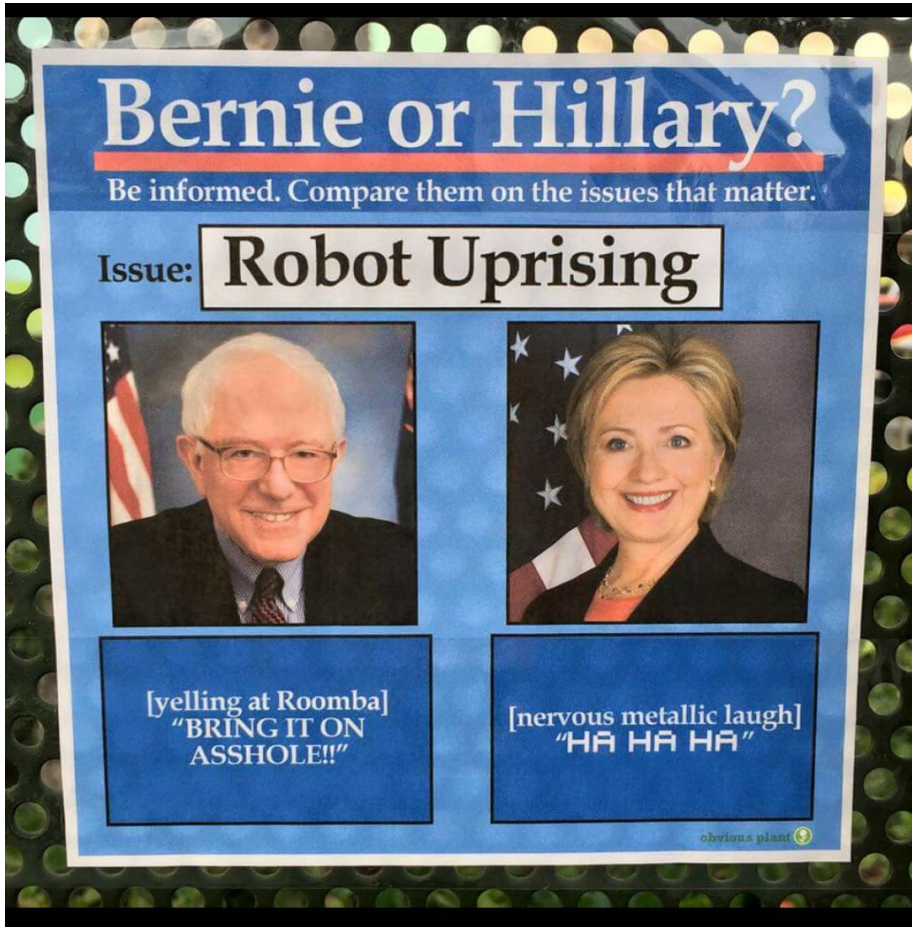




© 2016 Mesosphere, Inc. All Rights Reserved.

WHY BOTHER?

1. It's much easier to get compute resources nowadays!
2. It turns out getting sleep is good for you!
3. **Containers mean you can!**
   - No need for humans to ssh in and `apt-get package install python-mylibrary123`.
   - Is this the beginning of the robot uprising?



© 2016 Mesosphere, Inc. All Rights Reserved.

Introduction

MESOSPHERE DATACENTER OPERATING SYSTEM (DC/OS)

DC/OS Universe

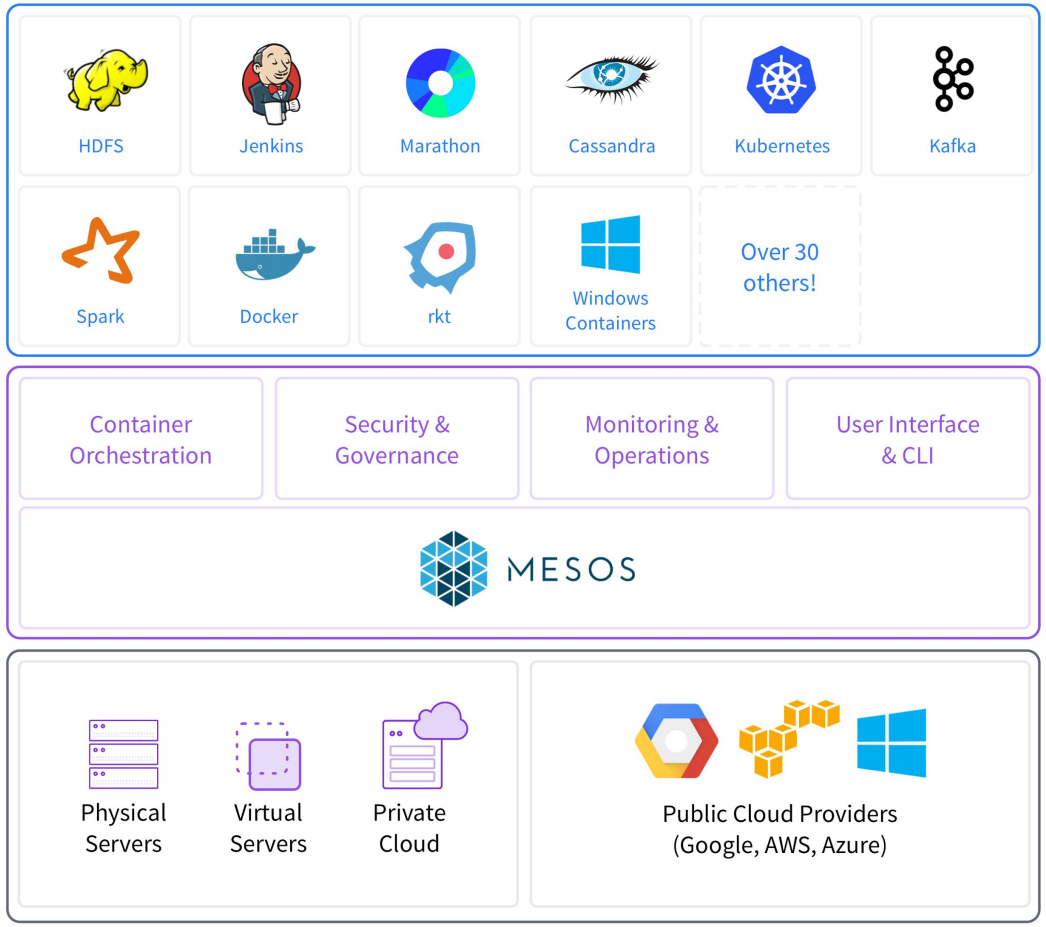
- Datacenter-wide services to power your apps
- Turnkey installation and lifecycle management
- Today's most popular services (and those yet to come)

Mesosphere DC/OS

- Container operations & big data operations
- Fault tolerance & high availability
- Open source & production proven at scale

Any Infrastructure

- Requires only a modern Linux distro (Windows coming soon)



© 2016 Mesosphere, Inc. All Rights Reserved.

DC/OS - THE DEFINITIVE PLATFORM FOR MODERN APPS

2000 2010 2013 2015 2016

Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center

Benjamin Hindman, Andy Konwinski, Matei Zaharia,
Ali Ghodsi, Anthony D. Joseph, Randy Katz, Scott Shenker, Ion Stoica
University of California, Berkeley

Thursday 30th September, 2010, 12:57

Abstract

We present Mesos, a platform for sharing commodity clusters between multiple diverse cluster computing frameworks, such as Hadoop and MPI. Sharing improves cluster utilization and avoids per-framework data repli-

The solutions of choice to share a cluster today are either to statically partition the cluster and run one framework per partition, or allocate a set of VMs to each framework. Unfortunately, these solutions achieve neither high utilization nor efficient data sharing. The main

Hewlett Packard Enterprise

Microsoft

DC/OS

DCOS Launched | DC/OS OSS Project

Mesosphere

Proprietary | Apache Mesos Project

Google
Borg & Omega

facebook
Tupperware & Bistro

airbnb

PayPal

NETFLIX

Microsoft Azure

AUTODESK

TWO SIGMA

GROUPON

verizon

yelp

Time Warner Cable

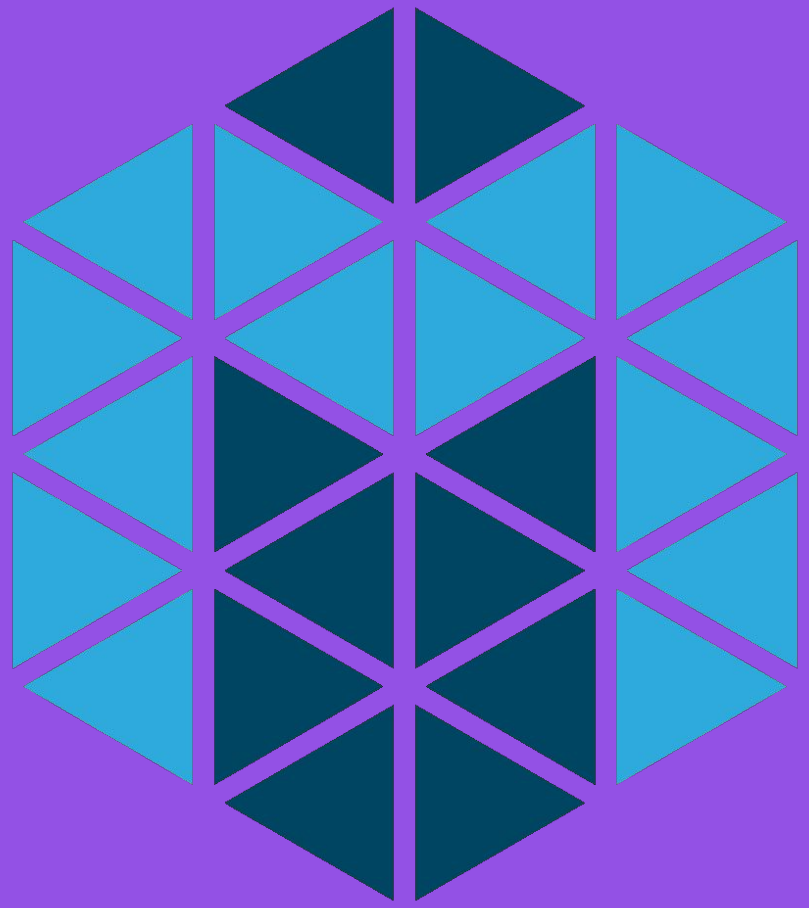
Bloomberg

JPMorgan Chase & Co.

esri

UBER

© 2015 Mesosphere, Inc. All Rights Reserved.



APACHE MESOS: THE STORY OF

© 2016 Mesosphere, Inc. All Rights Reserved.

Apache Mesos: The Story Of

THE BIRTH OF MESOS

Spring 2009

CS262B

Ben Hindman, Andy Konwinski and Matei Zaharia create “Nexus” as their CS262B class project.

March 2010

TWITTER TECH TALK

The grad students working on Mesos give a tech talk at Twitter.

September 2010

MESOS PUBLISHED

Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center is published as a technical report.

December 2010

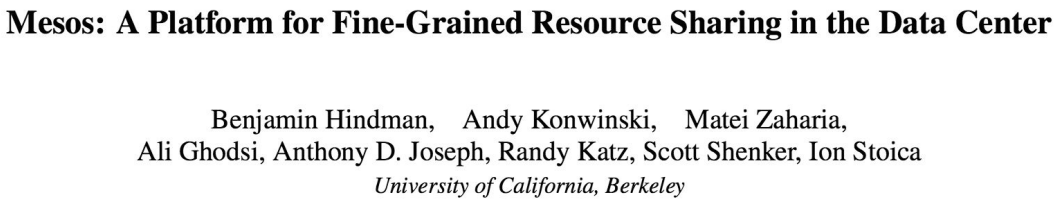
APACHE INCUBATION

Mesos enters the Apache Incubator.

© 2016 Mesosphere, Inc. All Rights Reserved.

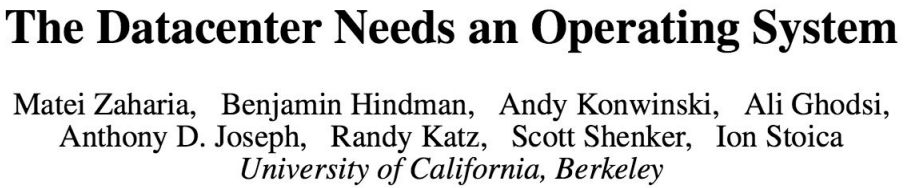
Apache Mesos: The Story Of

GRAD STUDENTS LEARNED HOW TO SHARE



Mesos: A Platform for Fine-Grained Resource Sharing in the Data Center

Benjamin Hindman, Andy Konwinski, Matei Zaharia,
Ali Ghodsi, Anthony D. Joseph, Randy Katz, Scott Shenker, Ion Stoica
University of California, Berkeley



The Datacenter Needs an Operating System

Matei Zaharia, Benjamin Hindman, Andy Konwinski, Ali Ghodsi,
Anthony D. Joseph, Randy Katz, Scott Shenker, Ion Stoica
University of California, Berkeley

Sharing resources between batch processing frameworks:

- Hadoop
- MPI
- Spark

What does an operating system provide?

- Resource management
- Programming abstractions
- Security
- Monitoring, debugging, logging

© 2016 Mesosphere, Inc. All Rights Reserved.

Apache Mesos: The Story Of

CLUSTERING YOUR RESOURCES FOR YOU

Apache Mesos is a **cluster resource manager**.

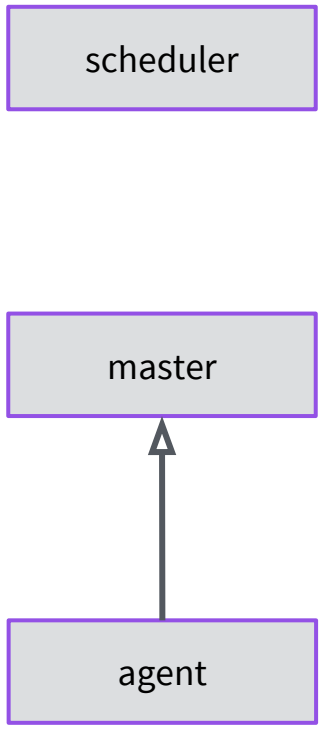
It handles:

- **Aggregating resources** and **offering them to schedulers**
- **Launching tasks** (i.e. processes) on those resources
- **Communicating the state of those tasks** back to schedulers
- Tasks can be:
  - Long running services
  - Ephemeral / batch jobs

© 2016 Mesosphere, Inc. All Rights Reserved.

Apache Mesos: The Story Of

A SIMPLE MESOS CLUSTER ON MONDAY MORNING

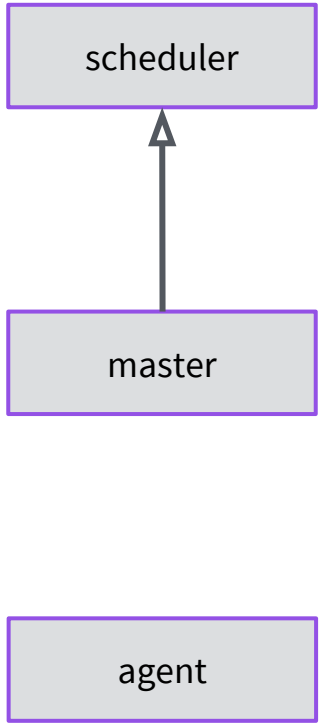


"Sir, I have some spare resources:
4 CPUs, 8 GB of memory and 1 TB of disk."

© 2016 Mesosphere, Inc. All Rights Reserved.

Apache Mesos: The Story Of

A SIMPLE MESOS CLUSTER ON MONDAY MORNING

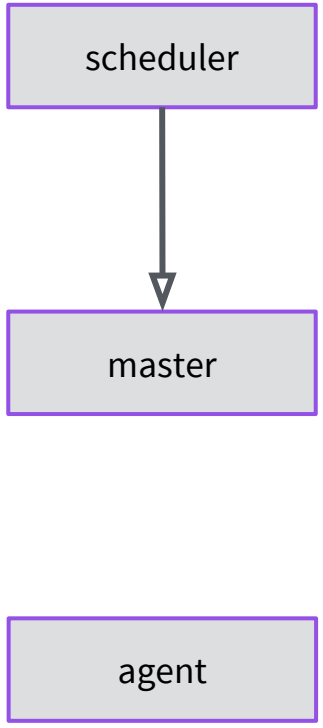


“Hey, scheduler, would you like some compute resources?”

© 2016 Mesosphere, Inc. All Rights Reserved.

Apache Mesos: The Story Of

A SIMPLE MESOS CLUSTER ON MONDAY MORNING



“Not right now, but thanks!”

© 2016 Mesosphere, Inc. All Rights Reserved.

Apache Mesos: The Story Of

A SIMPLE MESOS CLUSTER ON MONDAY MORNING

“Happy Monday! Here’s a bunch of work.”

scheduler

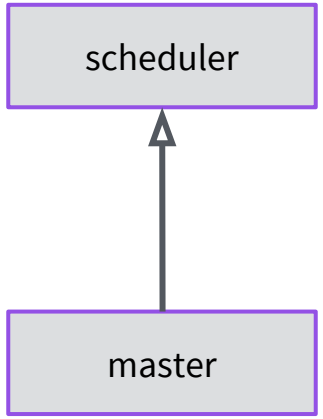
master

agent

© 2016 Mesosphere, Inc. All Rights Reserved.

Apache Mesos: The Story Of

A SIMPLE MESOS CLUSTER ON MONDAY MORNING



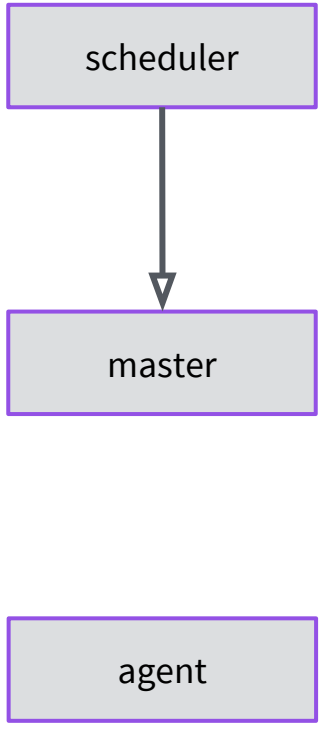
“Changed your mind?”

agent

© 2016 Mesosphere, Inc. All Rights Reserved.

Apache Mesos: The Story Of

A SIMPLE MESOS CLUSTER ON MONDAY MORNING

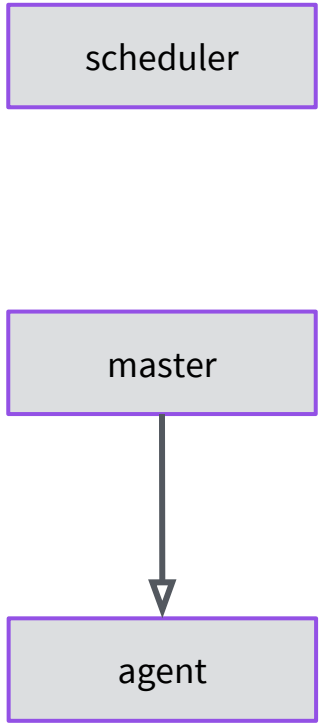


“How’d you know? Mind running this for me please?”

© 2016 Mesosphere, Inc. All Rights Reserved.

Apache Mesos: The Story Of

A SIMPLE MESOS CLUSTER ON MONDAY MORNING

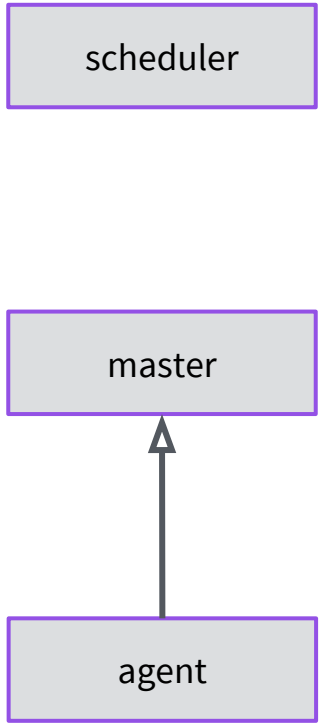


“Still got those spare resources? This task wants to run on them. Let me know how it goes.”

© 2016 Mesosphere, Inc. All Rights Reserved.

Apache Mesos: The Story Of

A SIMPLE MESOS CLUSTER ON MONDAY MORNING

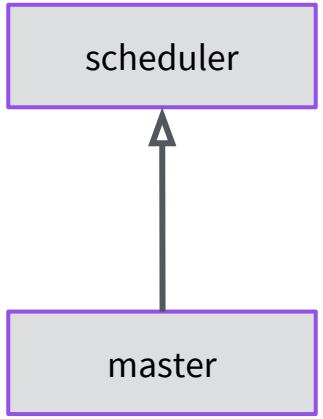


“Sure thing, it’s running.”

© 2016 Mesosphere, Inc. All Rights Reserved.

Apache Mesos: The Story Of

A SIMPLE MESOS CLUSTER ON MONDAY MORNING



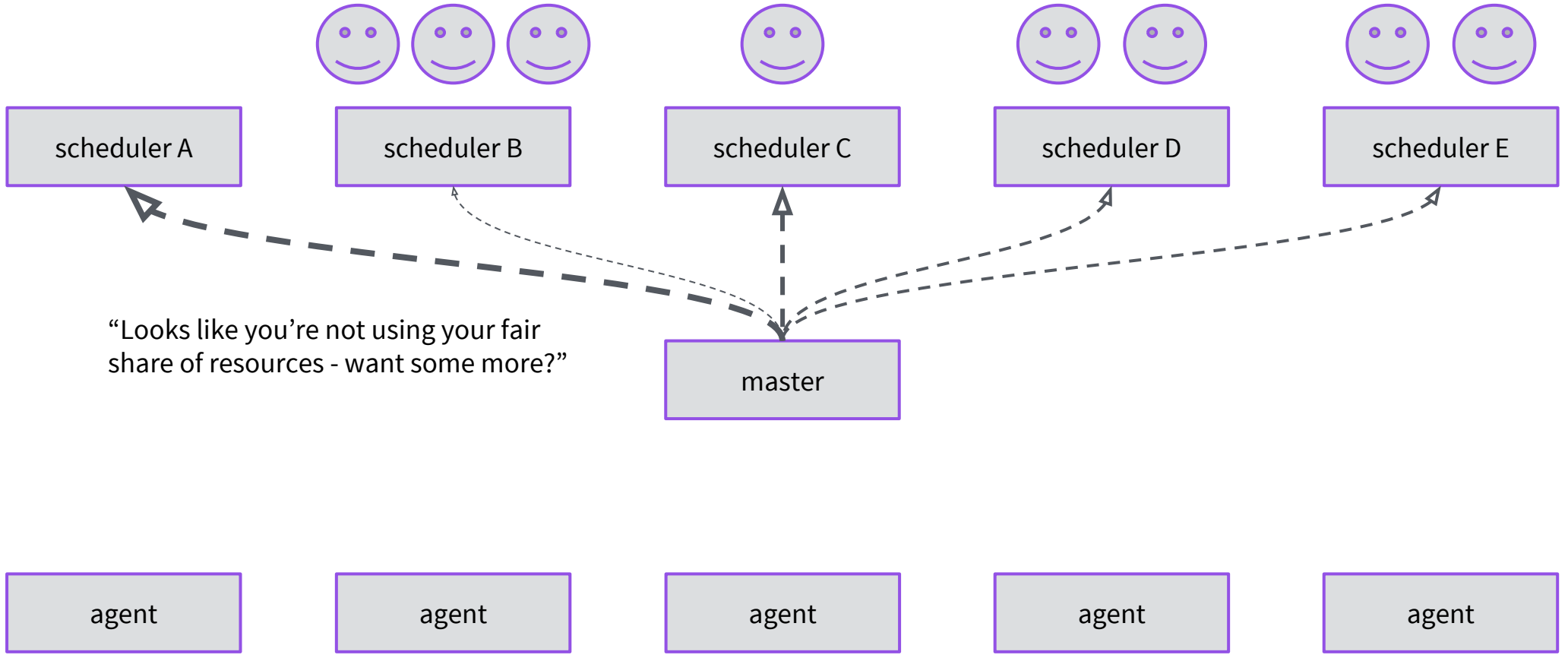
agent

"Your task is running! I'll let you know if that changes."

© 2016 Mesosphere, Inc. All Rights Reserved.

Apache Mesos: The Story Of

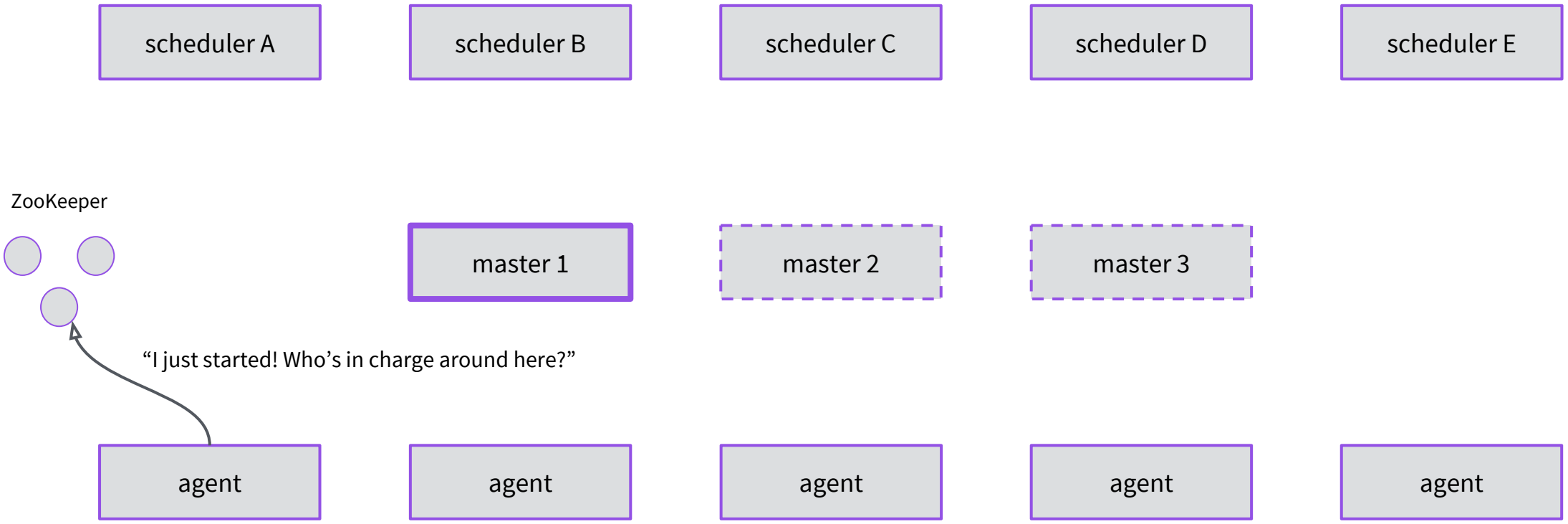
FAIRNESS FOR ALL SCHEDULERS



© 2016 Mesosphere, Inc. All Rights Reserved.

Apache Mesos: The Story Of

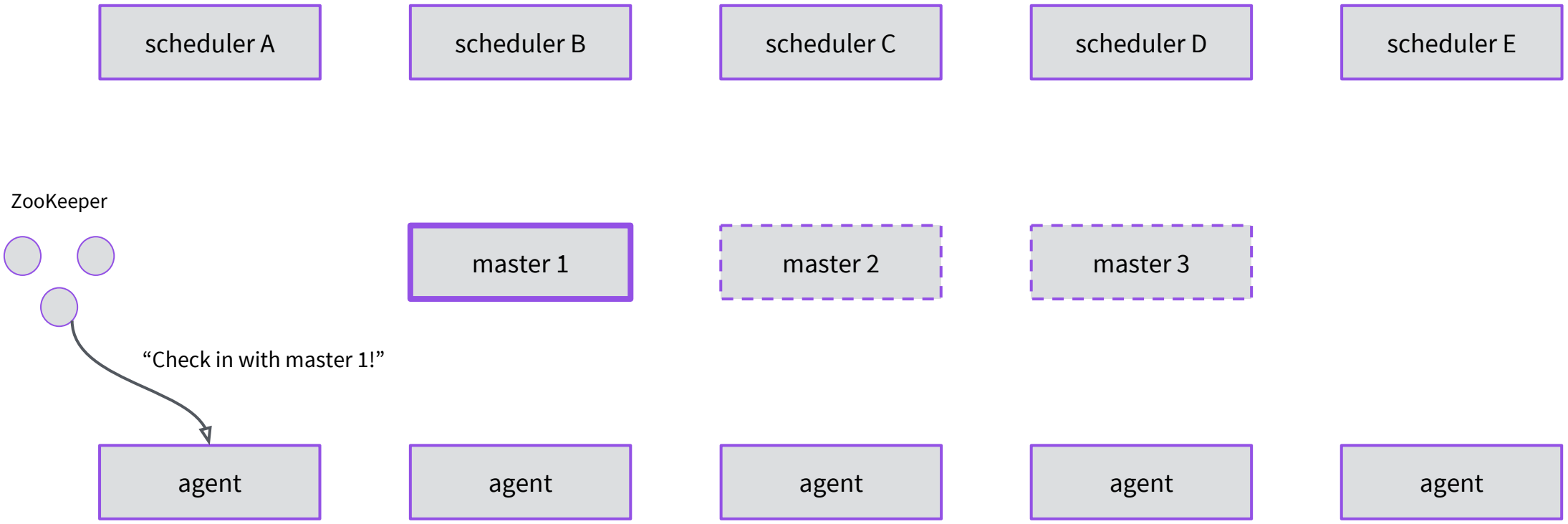
HELPING YOUR OPERATOR SLEEP WELL



© 2016 Mesosphere, Inc. All Rights Reserved.

Apache Mesos: The Story Of

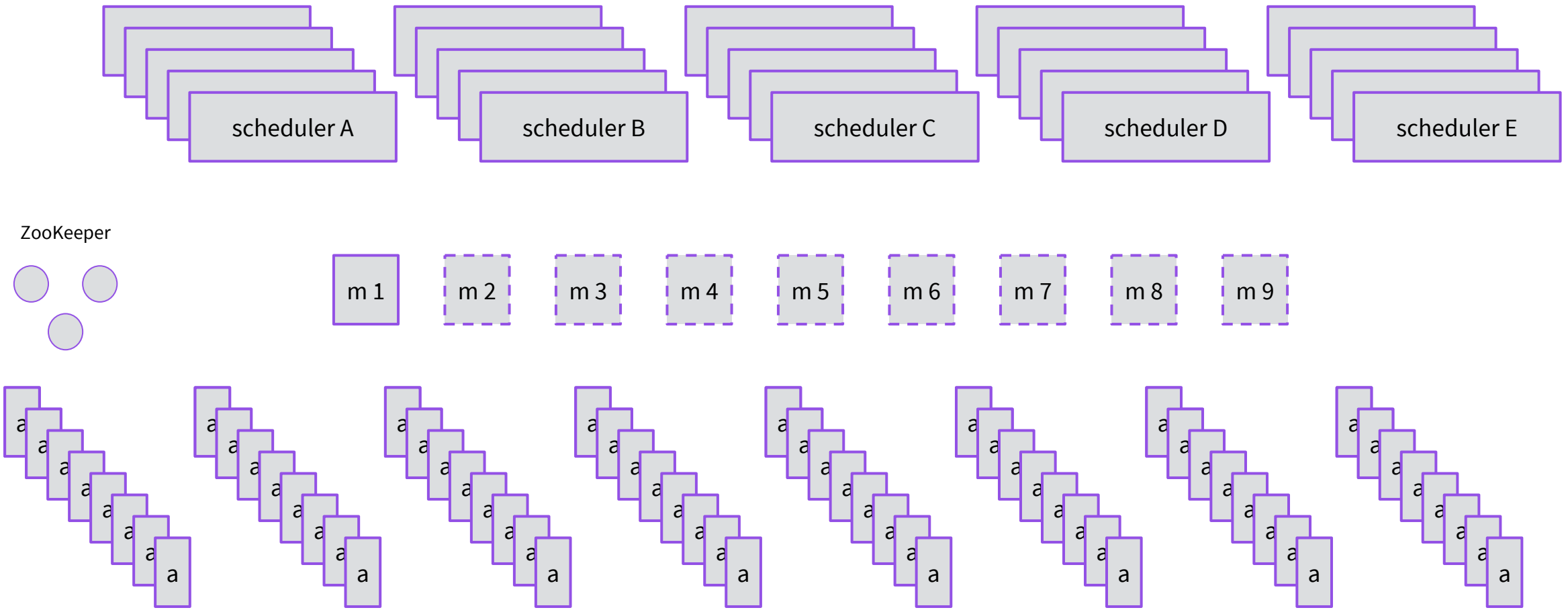
HELPING YOUR OPERATOR SLEEP WELL



© 2016 Mesosphere, Inc. All Rights Reserved.

Apache Mesos: The Story Of

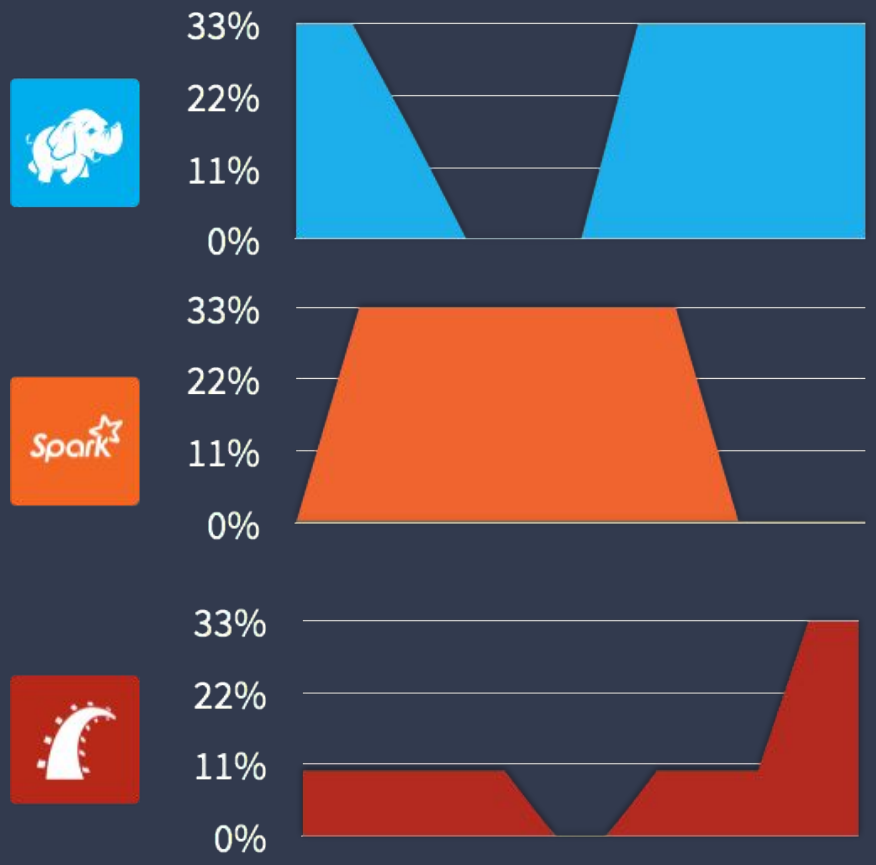
MESOS CLUSTERS CAN BE REALLY, REALLY LARGE

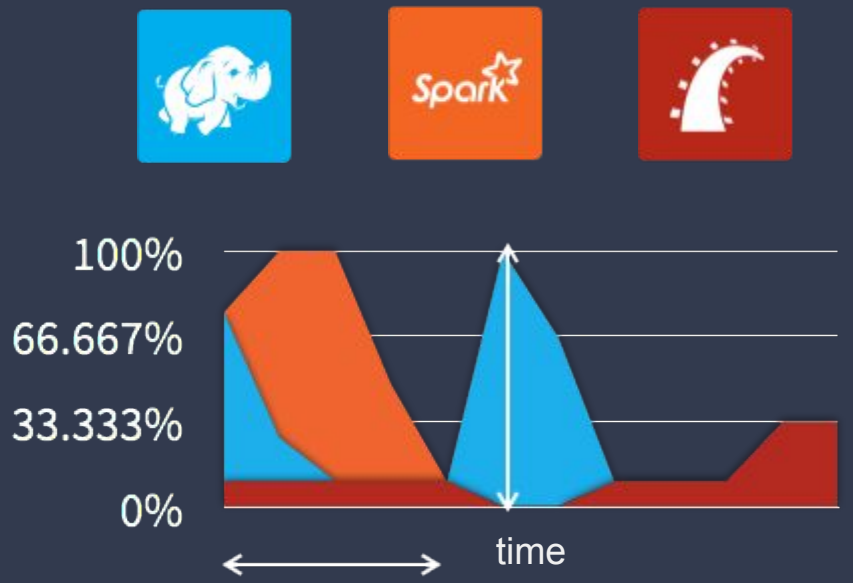


© 2016 Mesosphere, Inc. All Rights Reserved.

Apache Mesos: The Story Of

NOT USING MESOS IS EXPENSIVE!





© 2016 Mesosphere, Inc. All Rights Reserved.

MARATHON
(OR, HOW TO RUN MICROSERVICES ON MESOS)

© 2016 Mesosphere, Inc. All Rights Reserved.

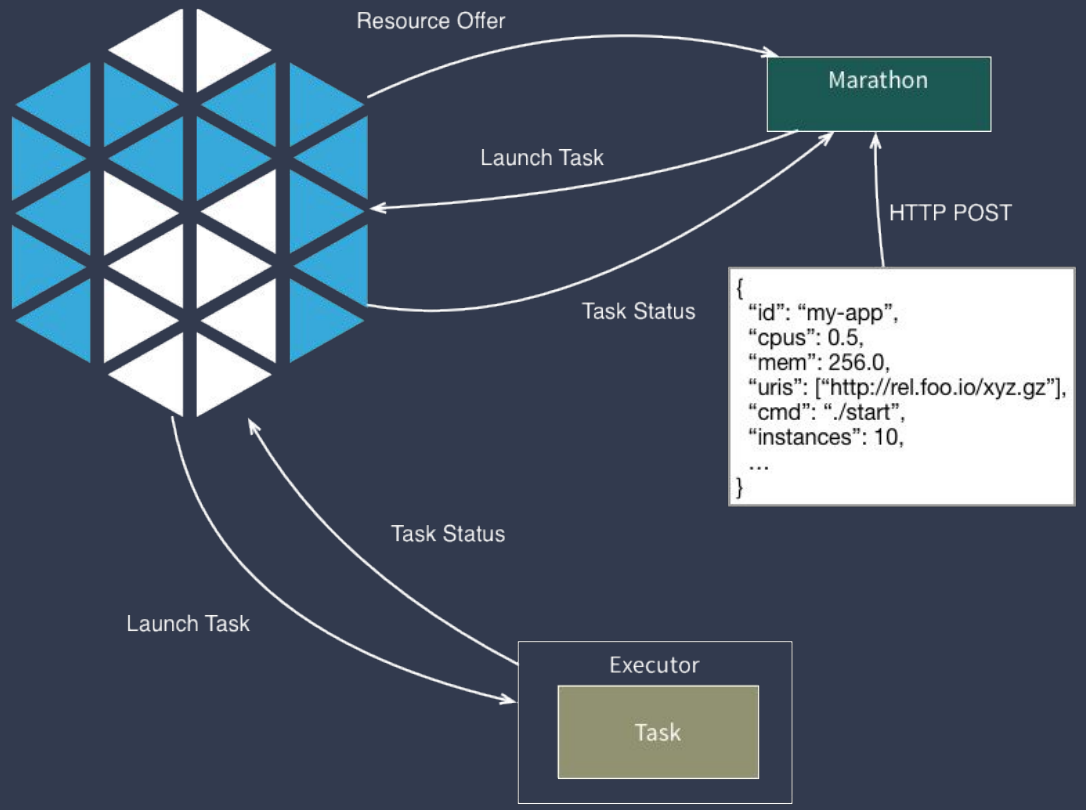
Marathon

MARATHON TALKS TO MESOS

Mesos can’t run applications on its own (!)

That’s where schedulers like Apache Aurora and Marathon come in.

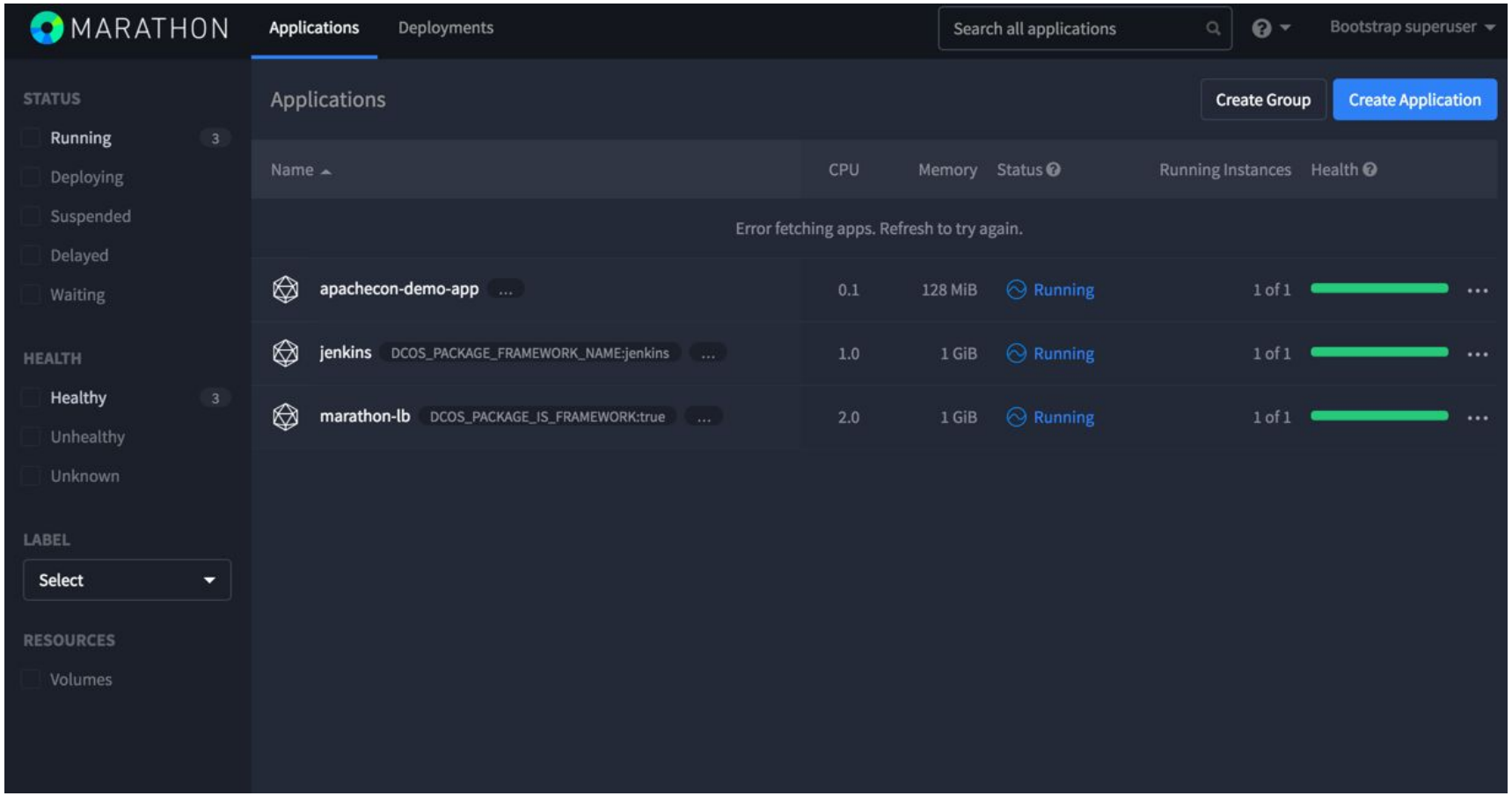
Marathon keeps your application running! A bit like a distributed “init.d”.



© 2016 Mesosphere, Inc. All Rights Reserved.

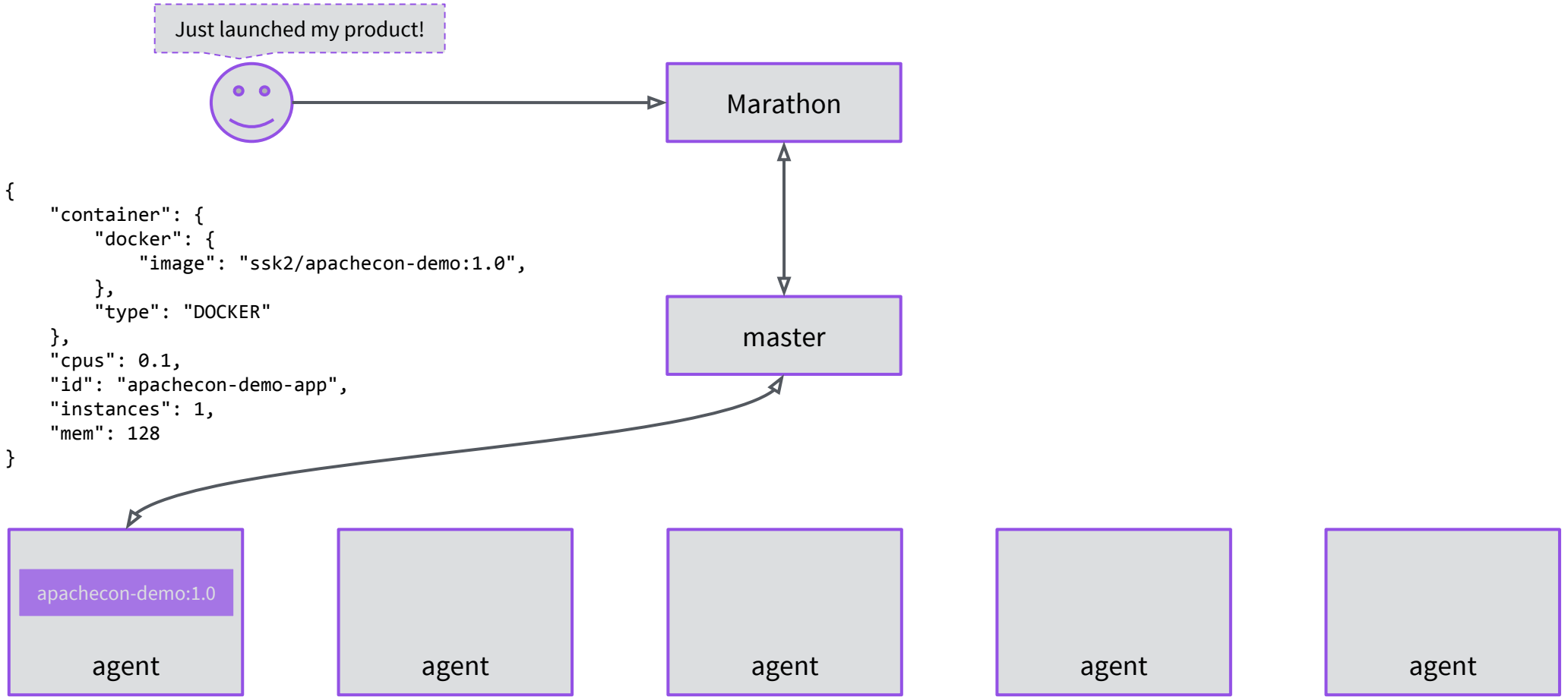
Marathon

A SELF SERVE INTERFACE TO YOUR CLUSTER



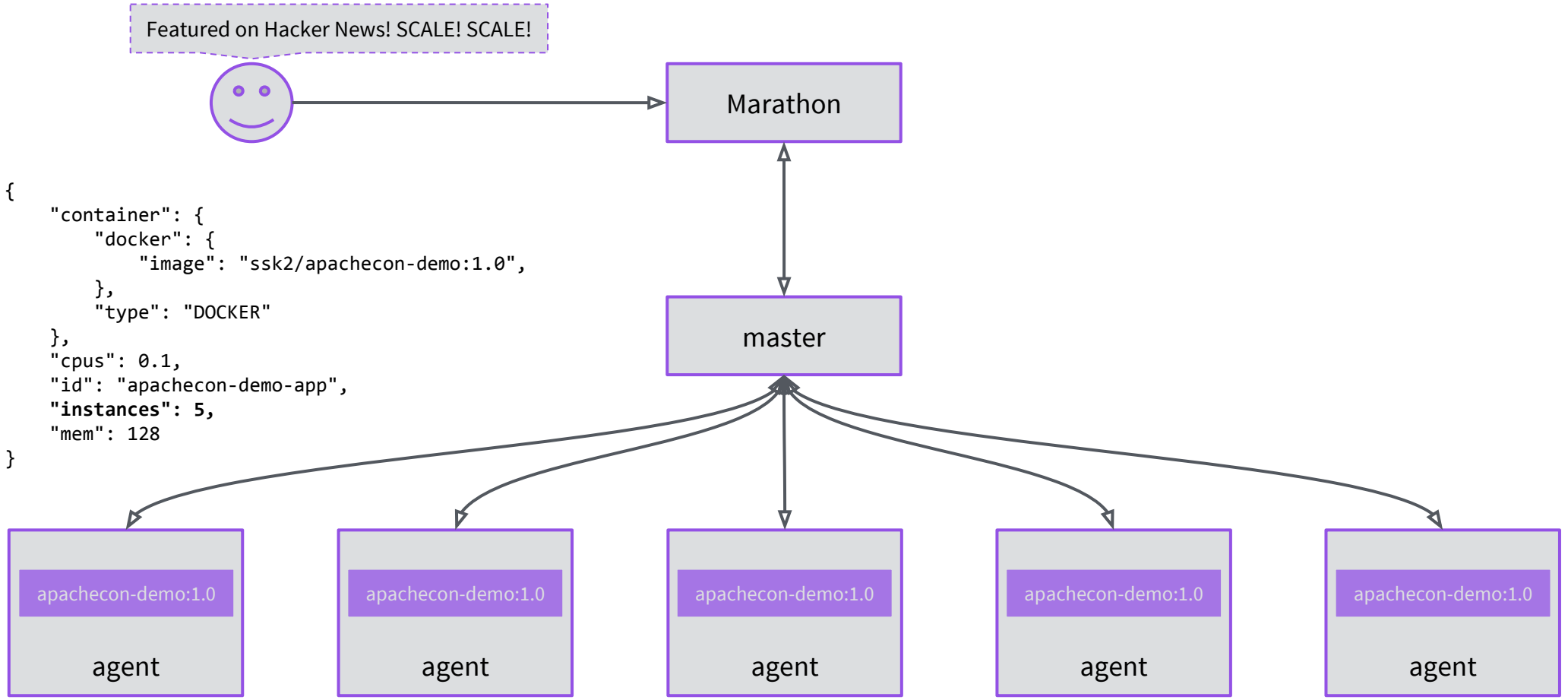
© 2016 Mesosphere, Inc. All Rights Reserved.

MARATHON MANAGES THE APPLICATION LIFECYCLE



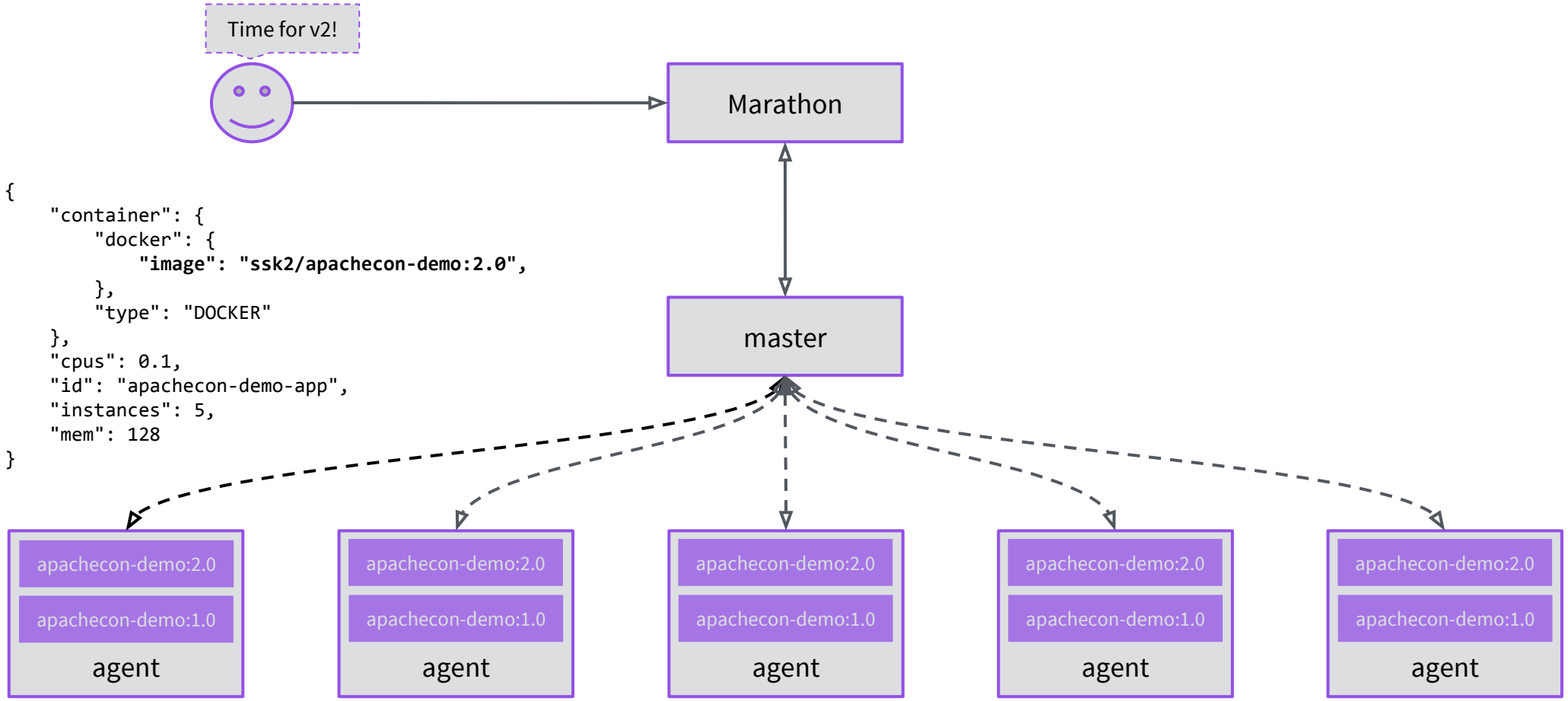
© 2016 Mesosphere, Inc. All Rights Reserved.

MARATHON MANAGES THE APPLICATION LIFECYCLE



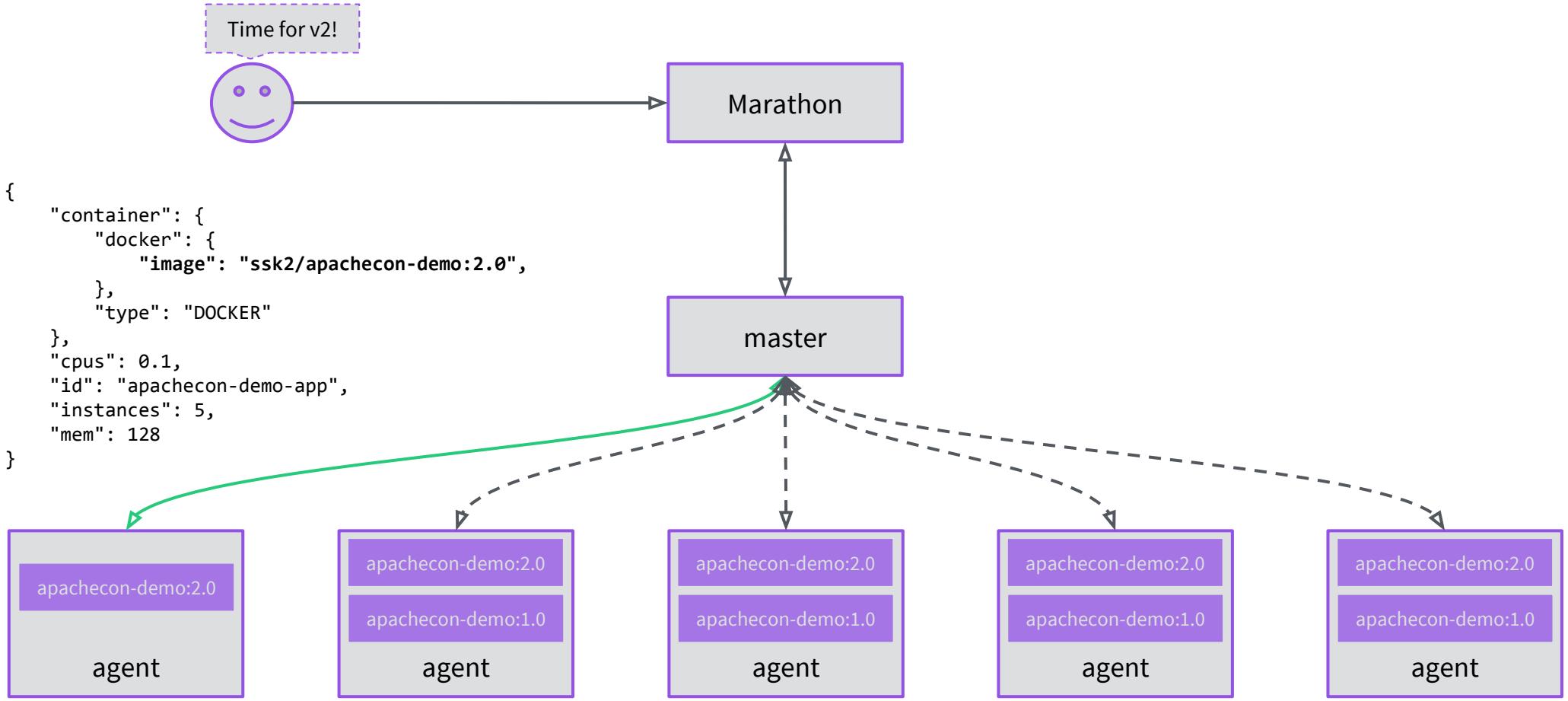
© 2016 Mesosphere, Inc. All Rights Reserved.

MARATHON MANAGES THE APPLICATION LIFECYCLE



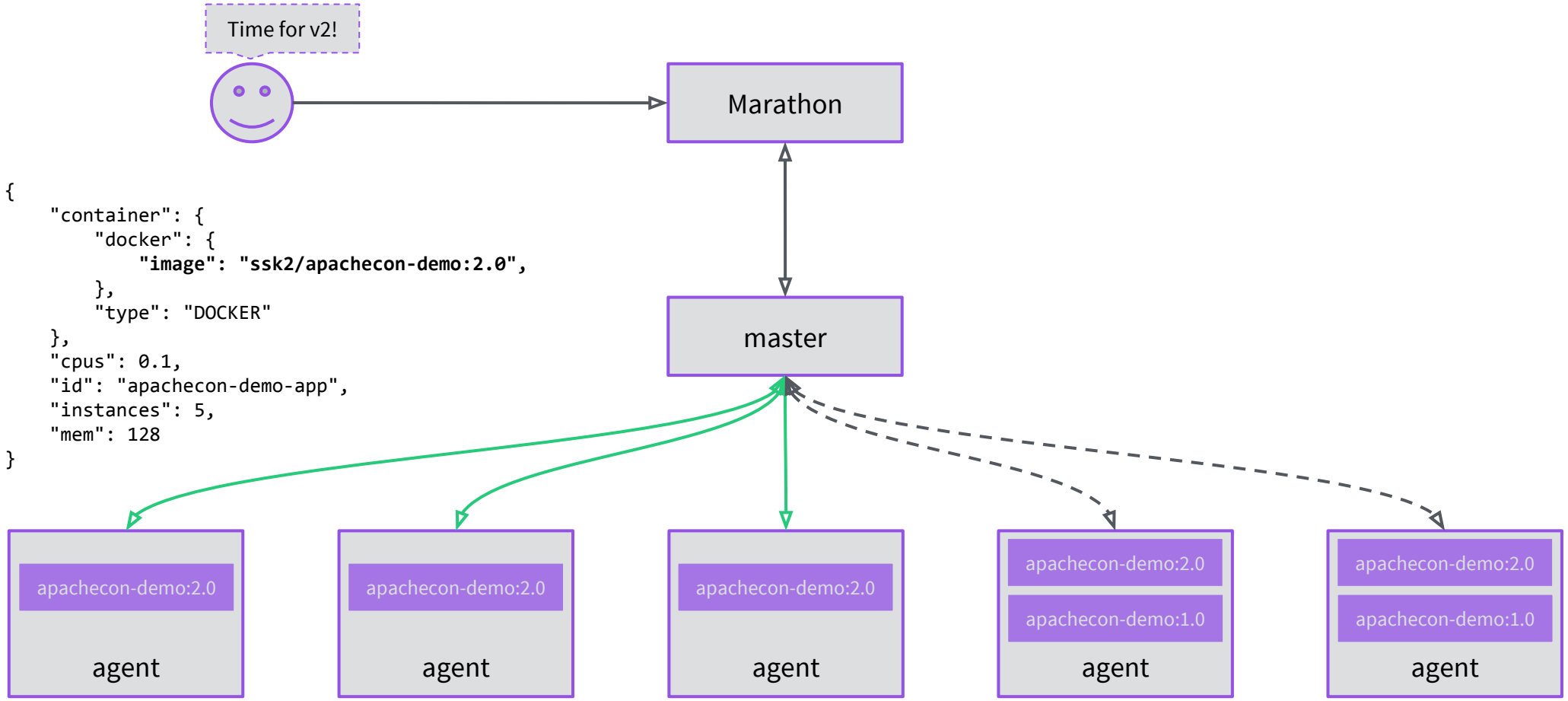
© 2016 Mesosphere, Inc. All Rights Reserved.

MARATHON MANAGES THE APPLICATION LIFECYCLE



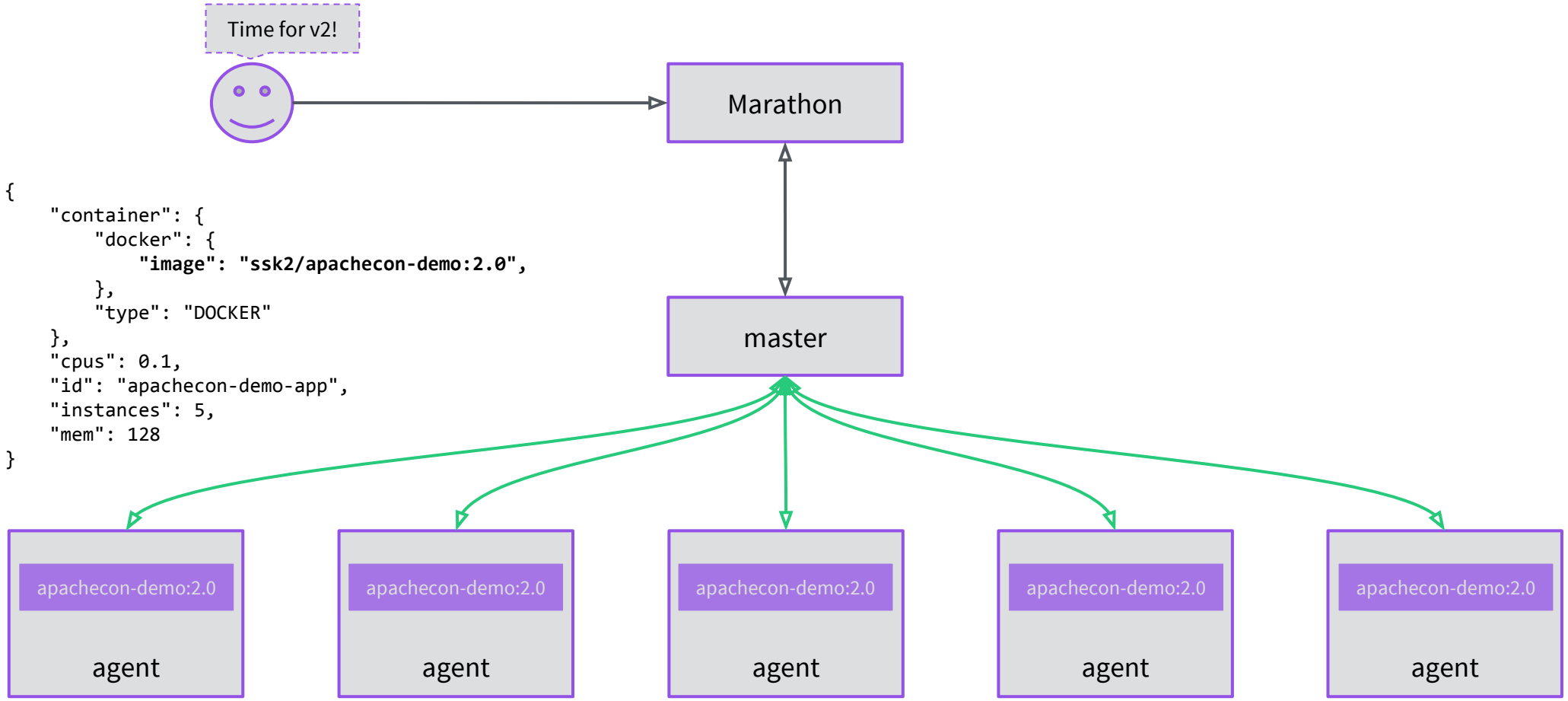
© 2016 Mesosphere, Inc. All Rights Reserved.

MARATHON MANAGES THE APPLICATION LIFECYCLE



© 2016 Mesosphere, Inc. All Rights Reserved.

MARATHON MANAGES THE APPLICATION LIFECYCLE



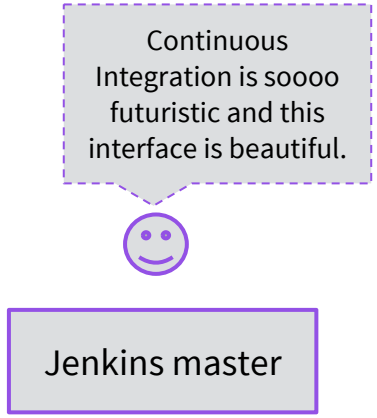
© 2016 Mesosphere, Inc. All Rights Reserved.

JENKINS ON DC/OS
(AND WHY YOU SHOULD PROBABLY BE RUNNING IT LIKE THIS)

© 2016 Mesosphere, Inc. All Rights Reserved.

Jenkins on DC/OS

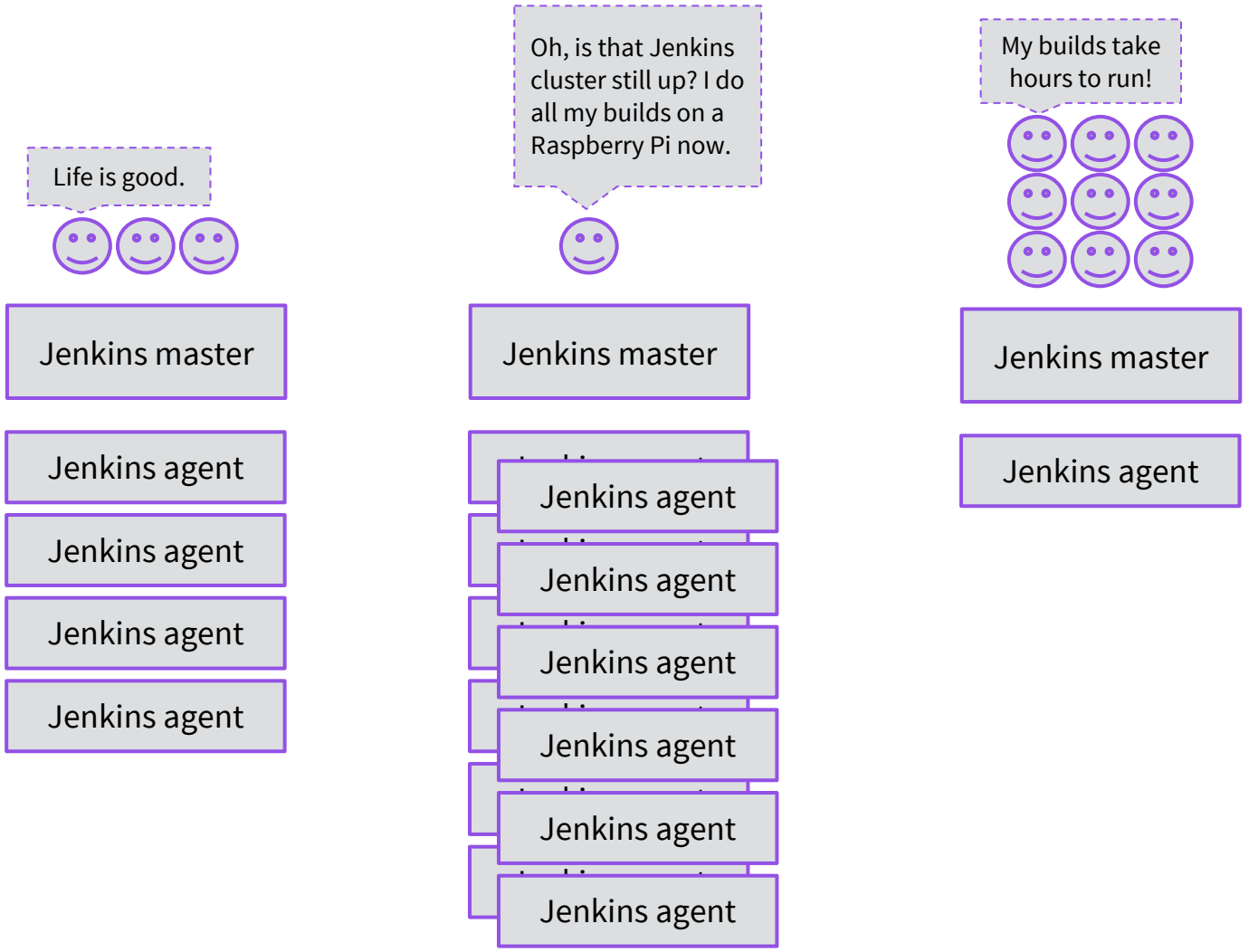
WHEN IT BEGAN



© 2016 Mesosphere, Inc. All Rights Reserved.

Jenkins on DC/OS

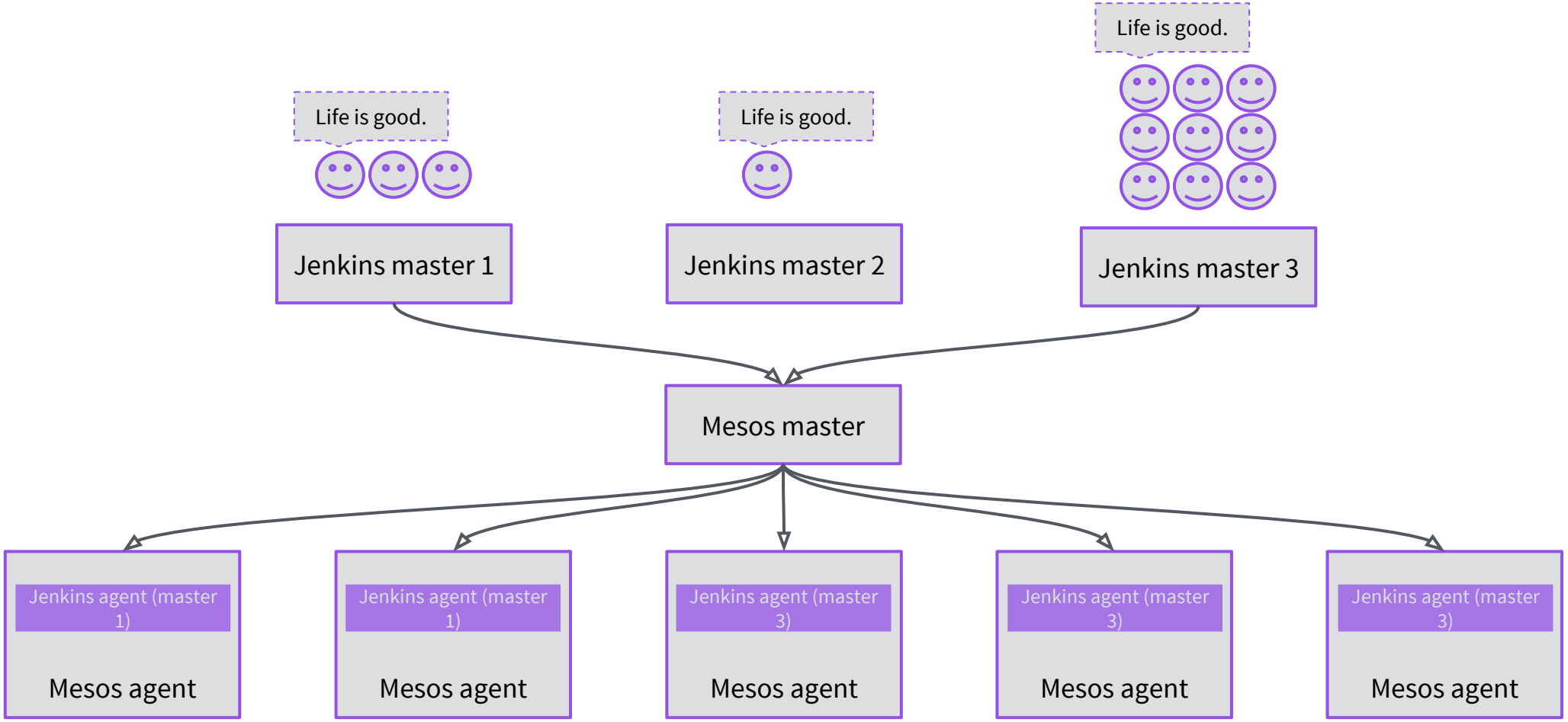
THE OLD WORLD



© 2016 Mesosphere, Inc. All Rights Reserved.

Jenkins on DC/OS

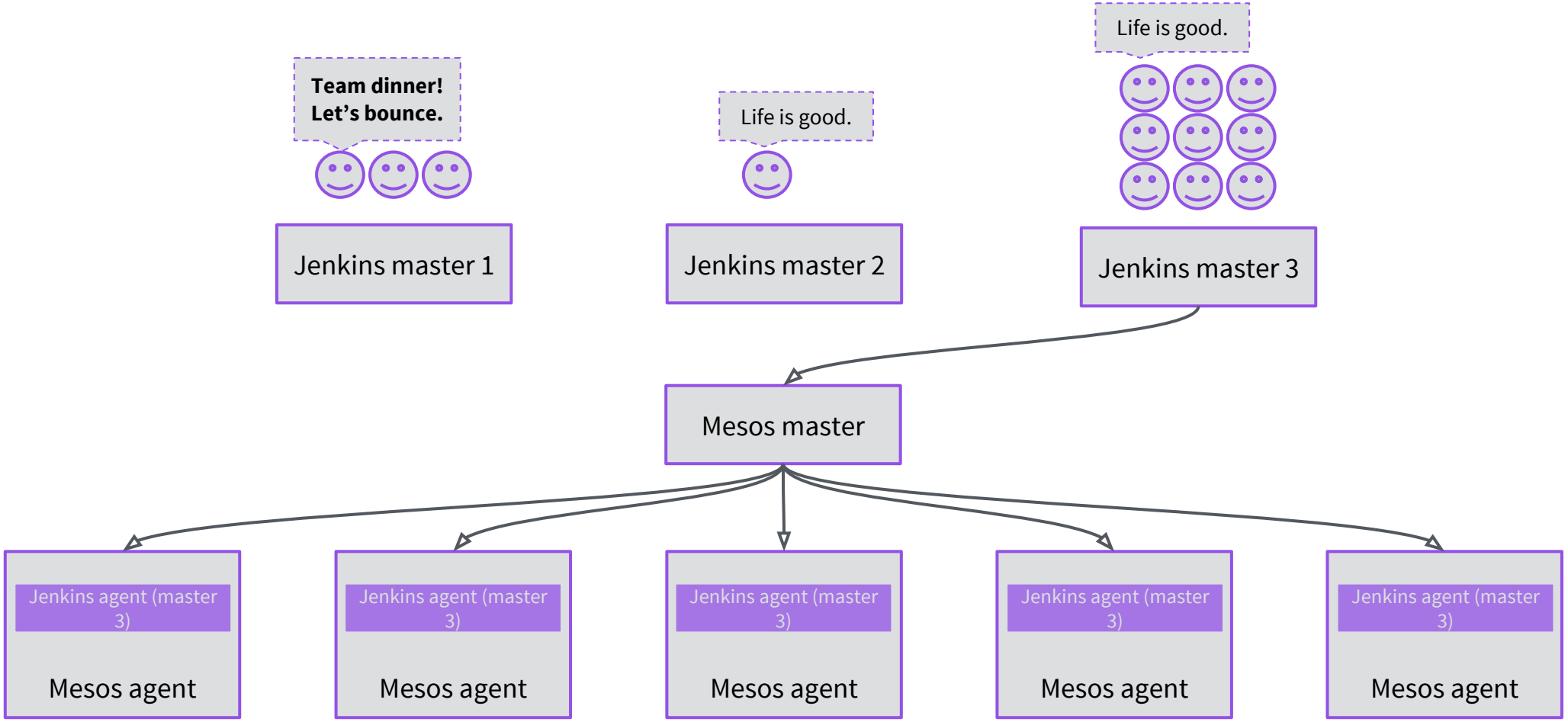
JUST USE WHAT YOU NEED, WHEN YOU NEED IT AND SHARE THE ~~LOVE~~ RESOURCES



© 2016 Mesosphere, Inc. All Rights Reserved.

Jenkins on DC/OS

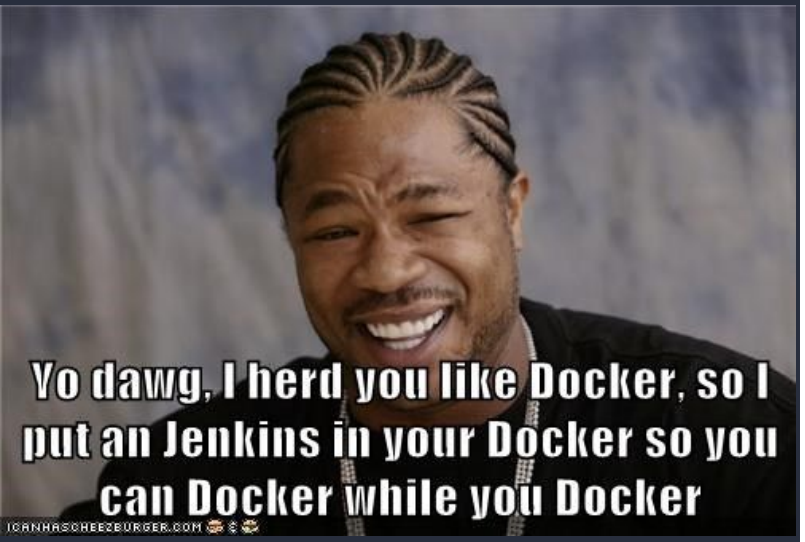
JUST USE WHAT YOU NEED, WHEN YOU NEED IT AND SHARE THE ~~LOVE~~ RESOURCES



© 2016 Mesosphere, Inc. All Rights Reserved.

Jenkins on DC/OS

BUILDING DOCKER IN DOCKER: ONE WEIRD TRICK



This brave new world of containers running in containers has a bit of a nesting problem.

We run everything inside a container to make it easy to bundle dependencies and to isolate it from other processes.

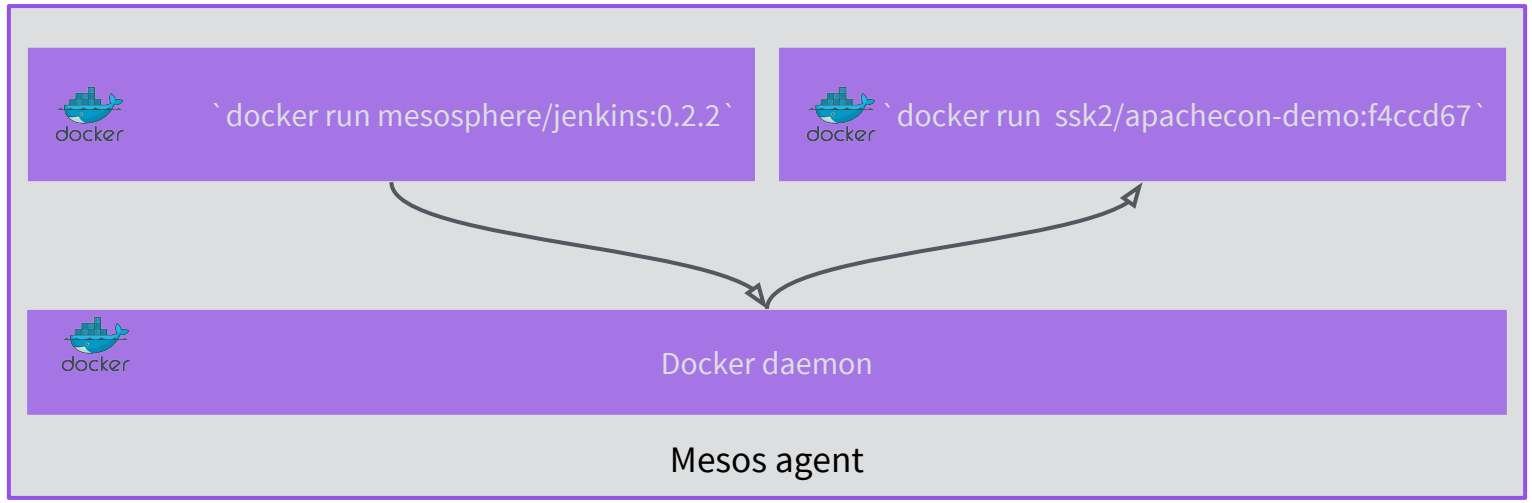
But when the thing that's running wants to build a container, what do you do?

© 2016 Mesosphere, Inc. All Rights Reserved.

Jenkins on DC/OS

BUILDING DOCKER IN DOCKER: ONE WEIRD TRICK

One recommended approach is to *bind mount* in the host system's Docker daemon.



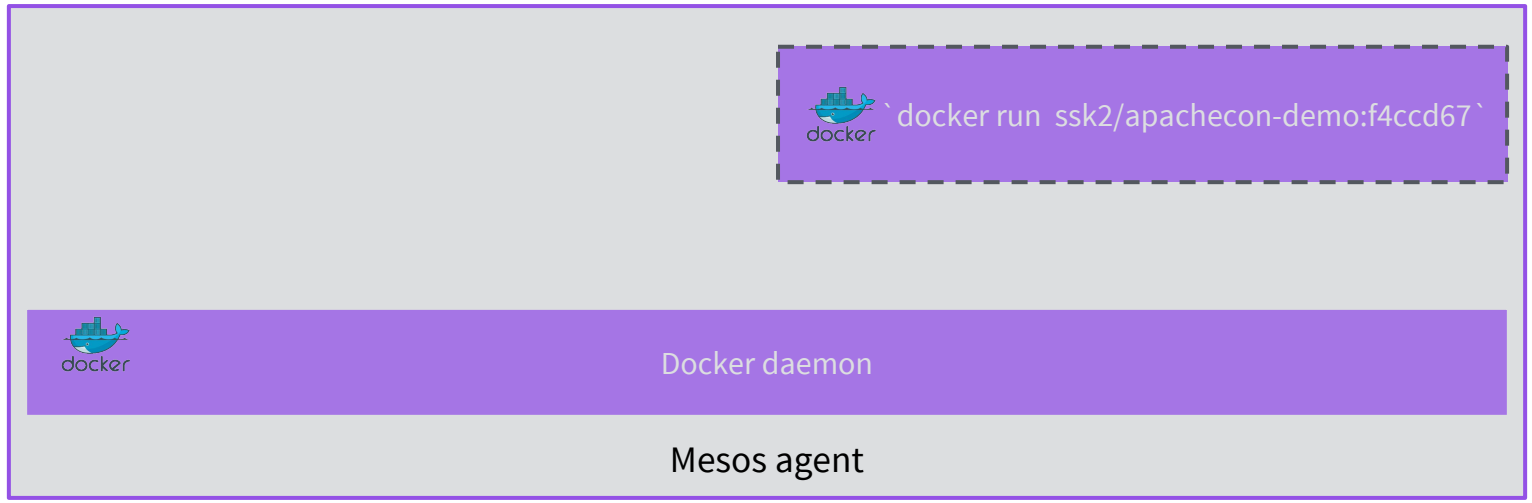
© 2016 Mesosphere, Inc. All Rights Reserved.

Jenkins on DC/OS

BUILDING DOCKER IN DOCKER: ONE WEIRD TRICK

This doesn't work for Mesos though! It doesn't track containers that it doesn't launch.

The sibling container becomes orphaned and runs forever.



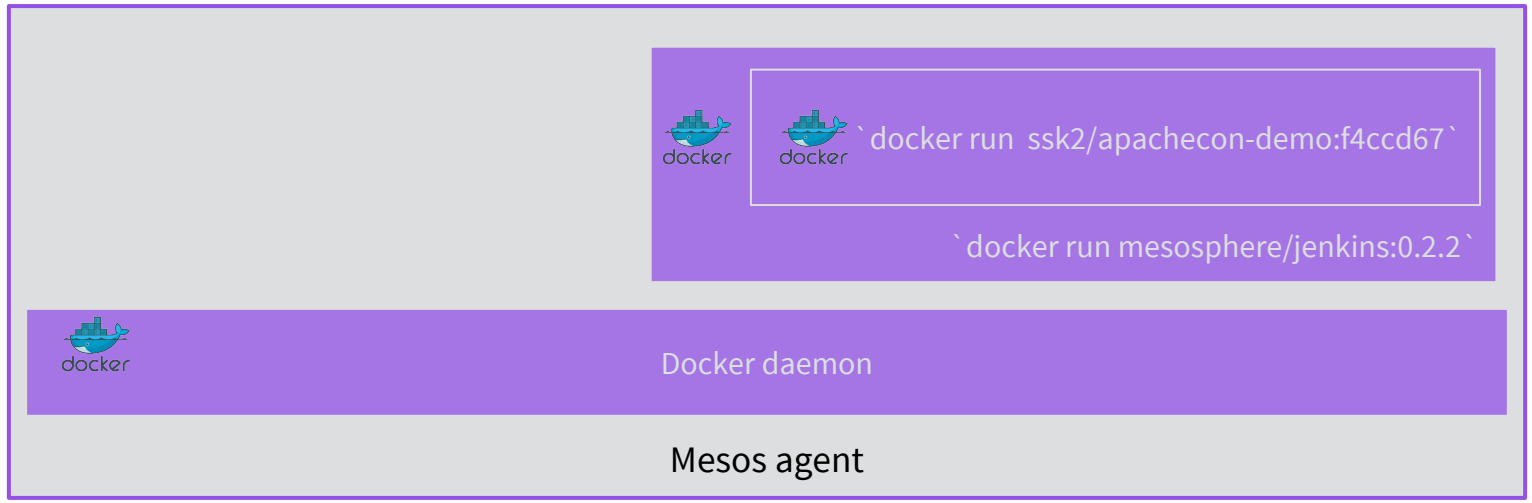
© 2016 Mesosphere, Inc. All Rights Reserved.

Jenkins on DC/OS

BUILDING DOCKER IN DOCKER: ONE WEIRD TRICK

Our solution is to use a customised Docker-in-Docker container.

This is a little slower but Mesos takes care of the resources!



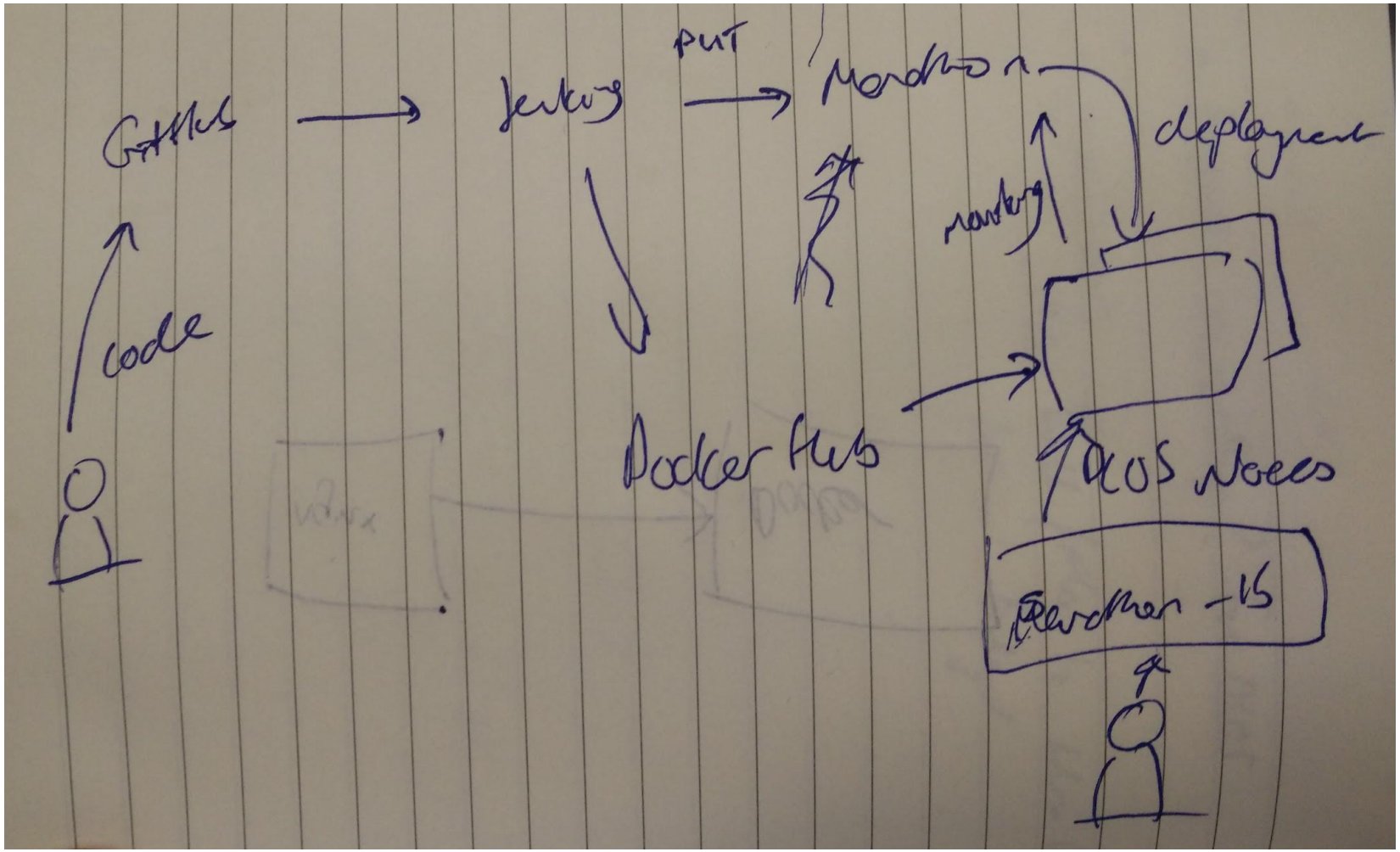
© 2016 Mesosphere, Inc. All Rights Reserved.

CONTINUOUS DEPLOYMENT

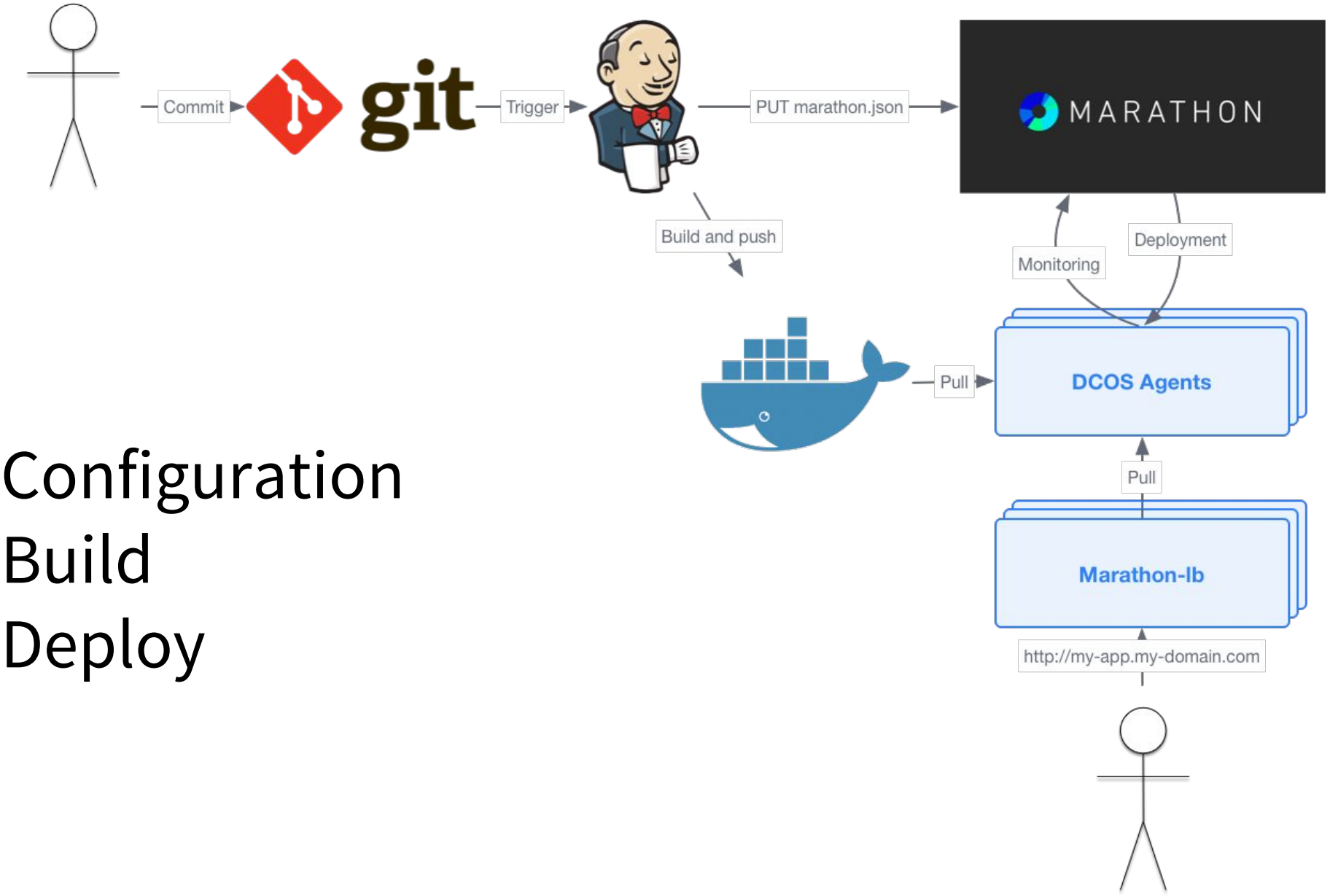
© 2016 Mesosphere, Inc. All Rights Reserved.

Continuous Deployment

PIPELINE: A FIRST PASS



© 2015 Mesosphere, Inc. All Rights Reserved.

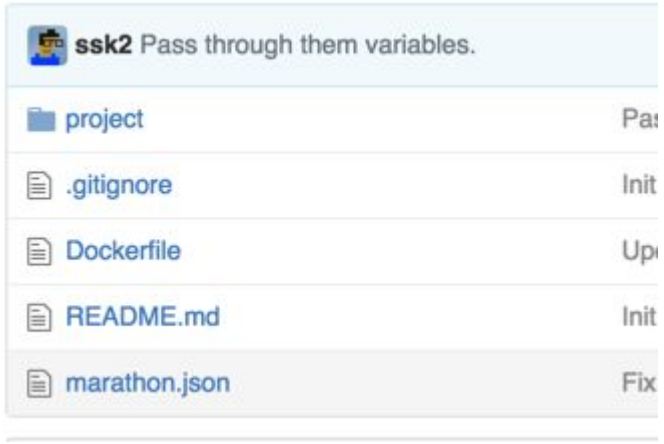


1. Configuration
2. Build
3. Deploy

© 2015 Mesosphere, Inc. All Rights Reserved.

Continuous Deployment

1. CONFIGURATION



Building a CD pipeline requires configuration in a couple of places:

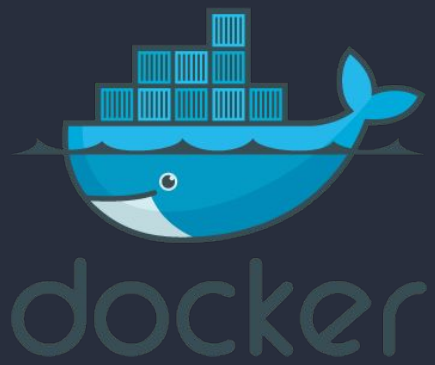
1. Docker and Marathon files in your repo
2. Build configuration in Jenkins*

*in the future, you'll be able to check in your build configuration alongside your repository too!

© 2015 Mesosphere, Inc. All Rights Reserved.

1. CONFIGURATION

DEPENDENCY MANAGEMENT



Docker is becoming the de-facto container format for packaging applications:

- Encapsulates dependencies
- Runs on your laptop
- Runs on your cluster

DC/OS has native support for Docker.

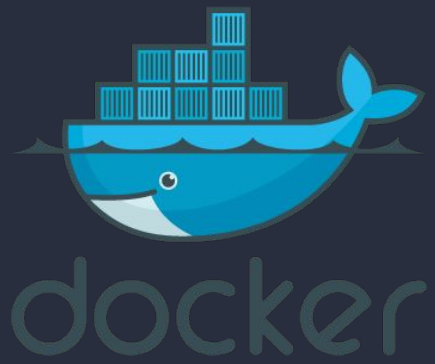
Just stick a Dockerfile (or two) in the root of your repository!

© 2015 Mesosphere, Inc. All Rights Reserved.

1. CONFIGURATION

DEPENDENCY MANAGEMENT

```
FROM jekyll/jekyll
ADD site /srv/jekyll
```



© 2015 Mesosphere, Inc. All Rights Reserved.

1. CONFIGURATION

APPLICATION CONFIGURATION

Marathon application definitions are JSON files that describe:

- resources required
- how many instances to run
- what command to run
- how to check your application is healthy

marathon.json should live in the root of your project repository.

MARATHON

© 2015 Mesosphere, Inc. All Rights Reserved.

1. CONFIGURATION

APPLICATION CONFIGURATION

```
{
  "id": "apachecon-demo",
  "container": {
    "type": "DOCKER",
    "docker": {
      "image": "ssk2/cd-demo:latest",
      "network": "BRIDGE",
      "portMappings": [{
          "containerPort": 80,
          "protocol": "tcp"
        }]
    }
  },
  "labels": {
    "HAPROXY_0_VHOST":
"sunil-889-publicsl-781ifozhqg3z-1399492298.us-west-2.elb.amazonaws.com",
    "HAPROXY_GROUP": "external"
  },
  "instances": 1,
  "cpus": 0.1,
  "mem": 128
}
```

MARATHON

© 2015 Mesosphere, Inc. All Rights Reserved.

Continuous Deployment

2. BUILDING

It’s trivial to install Jenkins on DCOS:

1. Create a JSON file:

```
{"jenkins": {"framework-name": "my-jenkins" }}
```

2. Install:

```
$ dcos package install --options=my-jenkins-config.json jenkins:
```

3. ???
4. Profit!

© 2015 Mesosphere, Inc. All Rights Reserved.

Continuous Deployment

2. BUILDING

Now, set up Jenkins:

1. Save your Docker Hub credentials

2. Set up triggered build to build and push Docker image

```
docker build . -t ssk2/whereisbot:$(GIT_BRANCH)

docker push ssk2/whereisbot:$(GIT_BRANCH)
```

3. Set up triggered build to update marathon.json using jq and PUT to Marathon

```
http PUT https://dcos/service/my-marathon/v2/app/ssk2/whereisbot < marathon.json
```

© 2015 Mesosphere, Inc. All Rights Reserved.

Continuous Deployment

2. BUILDING

Next, let's create a build:

1. Set up a build that polls GitHub periodically to build and push Docker image

```
docker login -u ${DOCKER_HUB_USERNAME} -p ${DOCKER_HUB_PASSWORD} -e sunil@mesosphere.com

docker build -t ssk2/apachecon-demo:$(GIT_COMMIT) .

docker push ssk2/apachecon-demo:$(GIT_COMMIT)
```

2. Add a Marathon post deploy step pointing to the DC/OS Marathon:
   - Set any variables you'd like to override.

© 2015 Mesosphere, Inc. All Rights Reserved.

Continuous Deployment

3. DEPLOYING

When you PUT to Marathon's API, you trigger a deployment.

```
http PUT https://dcos/service/my-marathon/v2/app/ssk2/whereisbot < marathon.json
```

Marathon attempts to scale application to desired state by:

- Launching new instances
  - By default try to launch 100% of instances requested at once

- Killing old instances when new instances are healthy

© 2015 Mesosphere, Inc. All Rights Reserved.

THANK YOU!

Come and talk to us!

- Email me at sunil@mesosphere.io
- Slides will be up at http://mesosphere.github.io/presentations
- Check out https://dcos.io