

BUILDING YOUR FIRST APP WITH APACHE MESOS AND DC/OS

QCon
LONDON

github.com/mesosphere/training/tree/master/qcon-training-03-2017

sys admin/SRE

developer

data engineer

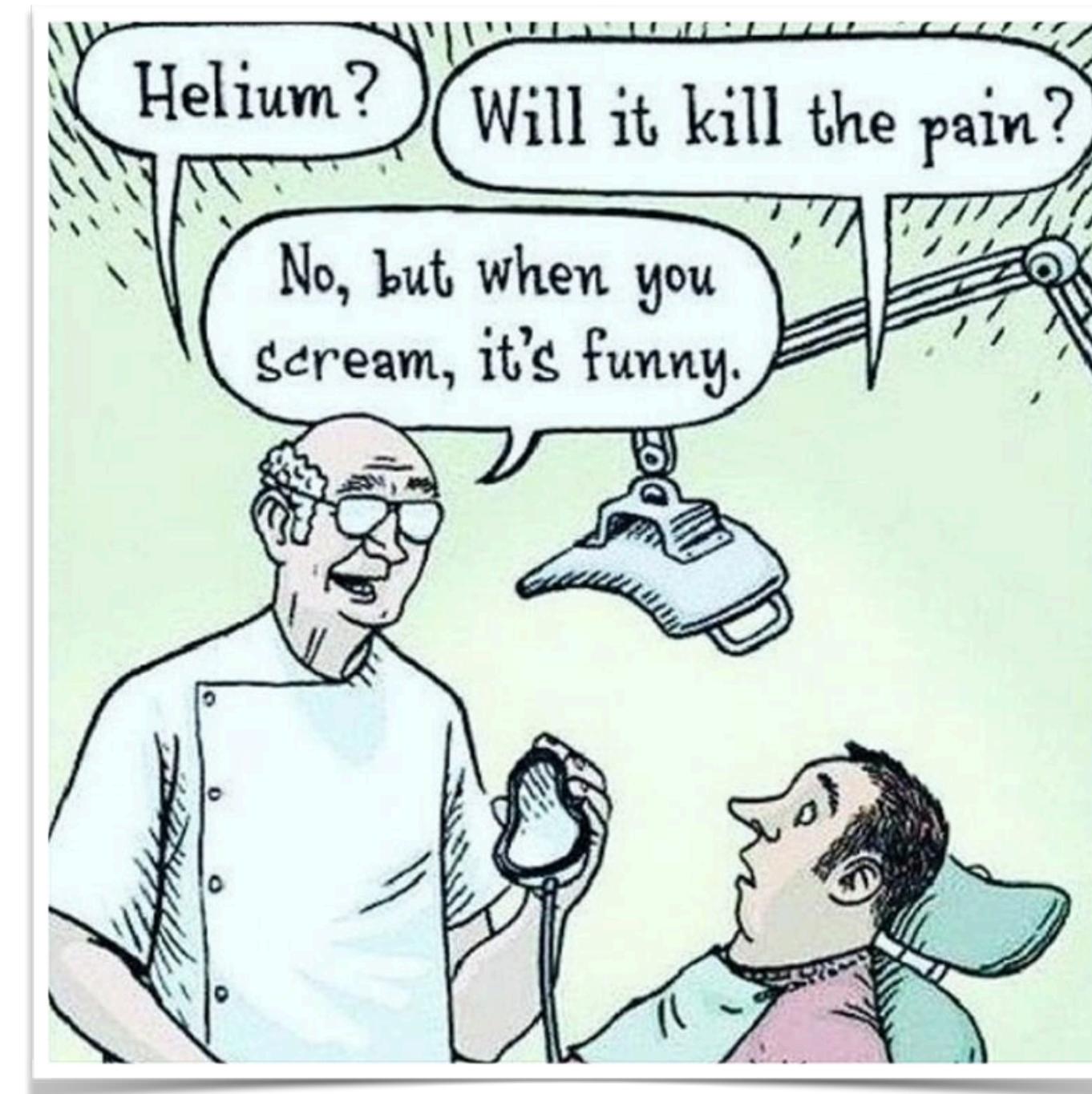
devops

QA/test engineer

architect



MOTIVATION



CHALLENGE

THE WALL STREET JOURNAL.

Home World U.S. Politics Economy Business Tech Markets Opinion Arts Life

ESSAY

Why Software Is Eating The World

By MARC ANDREESSEN

August 20, 2011

This week, Hewlett-Packard (where I am on the board) announced that it is exploring jettisoning its struggling PC business in favor of investing more heavily in software, where it sees better potential for growth. Meanwhile, Google plans to buy up the cellphone handset maker Motorola Mobility. Both moves surprised the tech world. But both moves are also in line with a trend I've observed, one that makes me optimistic about the future growth of the American and world economies, despite the recent turmoil in the stock market.



In an interview with WSJ's Kevin Delaney, Groupon and LinkedIn investor Marc Andreessen insists that the recent popularity of tech companies does not constitute a bubble. He also stressed that both Apple and Google are undervalued and that "the market doesn't like tech."

In short, software is eating the world.

More than 10 years after the peak of the 1990s dot-com bubble, a dozen or so new Internet companies like Facebook and Twitter are sparking controversy in Silicon Valley, due to their rapidly growing private market valuations, and even the occasional successful IPO. With scars from the heyday of Webvan and Pets.com still fresh in the investor psyche, people are asking, "Isn't this just a dangerous new bubble?"

CHALLENGE

Attn: New Data Regulation - Learn More About The GDPR. Speak with Data Privacy Expert from Ghostery! | Ad

Michael Hausenblas
Developer and Cloud Advocate at Mesosphere...
Your recent activity

9 people viewed your profile in the past 3 days

205 views on your update "Strong adoption for open source DC/O..."

Share an update Upload a photo Write an article

20+ new updates

Gayle Grasso RVP, Sales East at Mesosphere, Inc. 1d

Mesosphere, Inc.

In this webinar, we'll show you how, in 60 seconds or less, HTTP-based microservices can plug directly into Marathon for service discovery by installing linkerd, a service mesh for cloud-native applications. Check it out here:... show more

MESOSPHERE

Service discovery and visibility for microservices on DC/OS - Mesosphere mesosphere.com • In this webinar, we'll show you how, in 60 seconds or less, HTTP-based microservices can pl...

Like • Comment • Share

"On Second Thought" Brocade Cartoons shared: Follow

Sponsored

Afraid of admitting you have an Old IP problem? On Second Thought... Brocade <http://bit.ly/1tdfjYE>

SHARE > Networkers Anonymous

12 ways to keep in touch

Roman Shaposhnik has a new job. VP Technology at ODPI

Like Message Skip

11

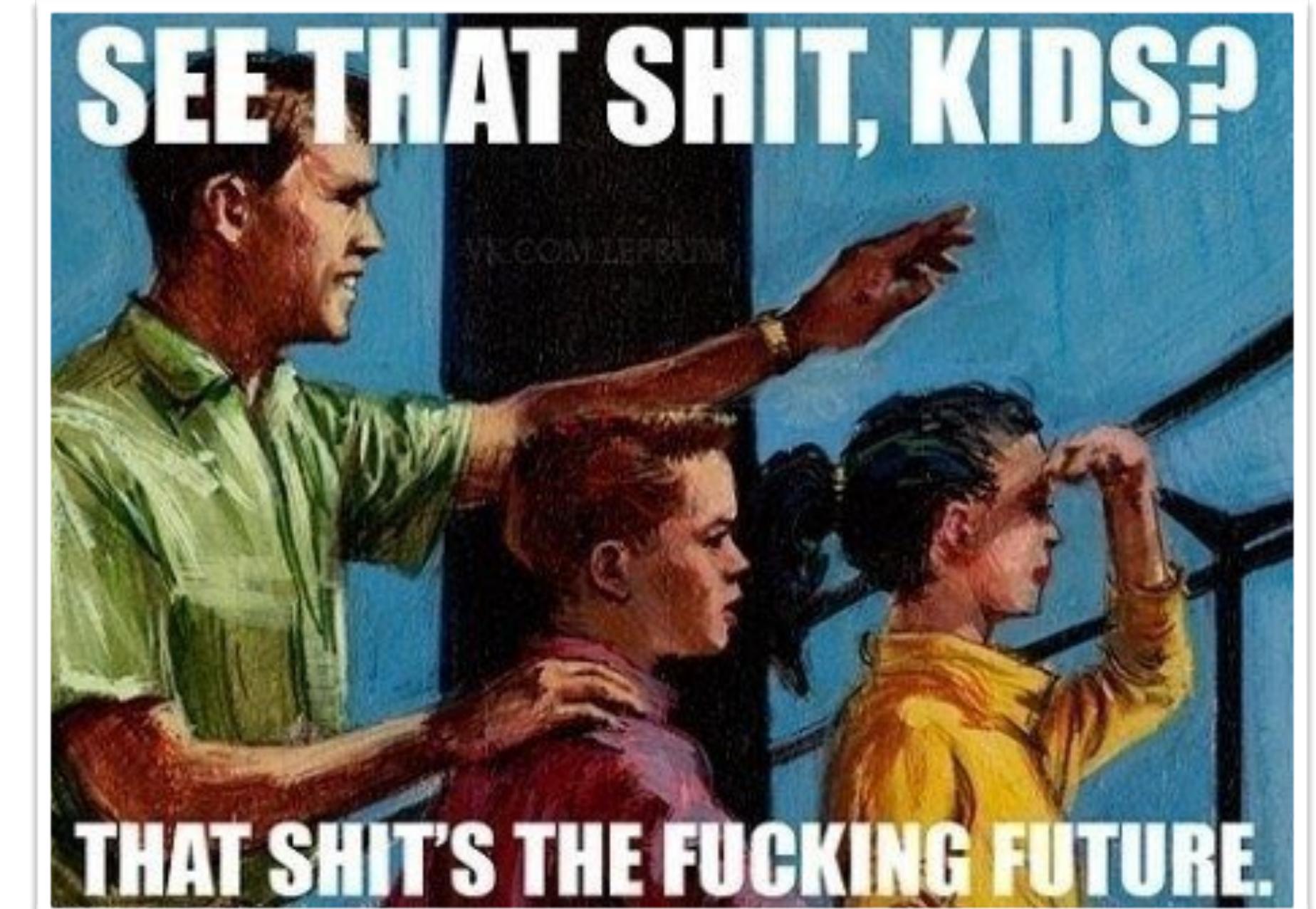
Get the latest jobs and industry news

Michael, explore relevant opportunities with New Ireland Assurance

Follow

About Help Feedback Cookies Privacy & Terms ▾ LinkedIn Corp. © 2016

CONTAINER 101



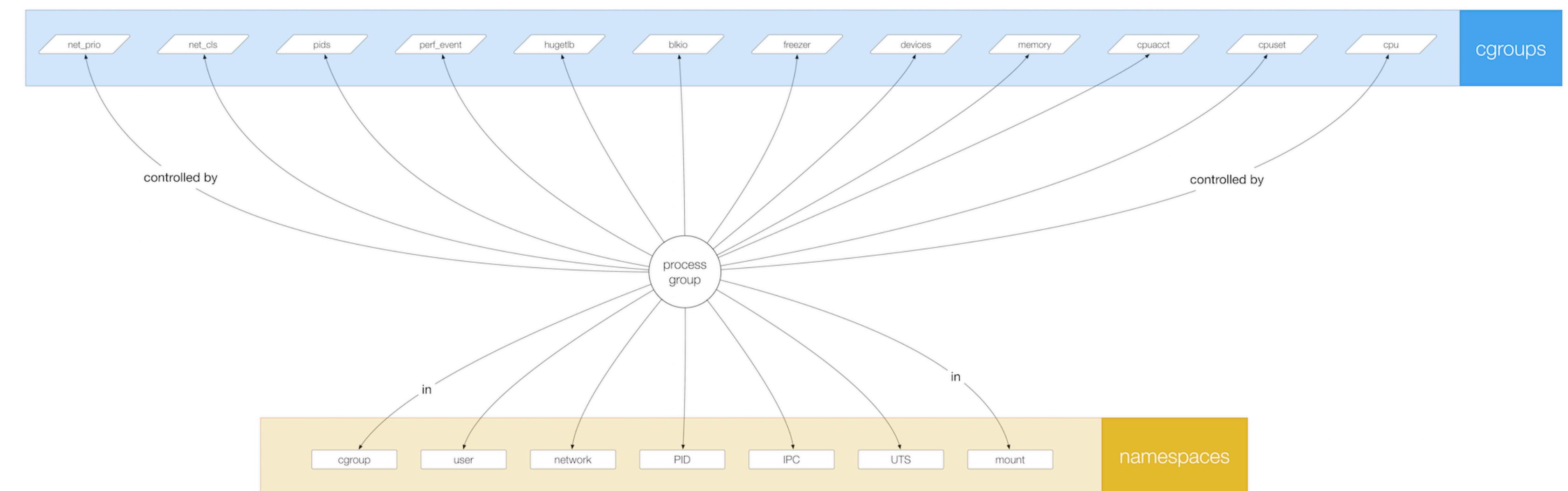
CONTAINER 101

- Containers are not VMs
- app-level dependency management
- lightweight (startup time, footprint, average runtime)
- isolation & resource consumption/accounting

CONTAINER 101

- namespaces (resource isolation)
 - CLONE_NEWNS (Linux 2.4.19) filesystem mount points
 - CLONE_NEWUTS (Linux 2.6.19) hostname & NIS domain name
 - CLONE_NEWIPC (Linux 2.6.19) interprocess communication
 - CLONE_NEWPID (Linux 2.6.24) process ID space
 - CLONE_NEWNET (Linux 2.6.29) network system resources (network devices, IP routing tables, port numbers, etc.)
 - CLONE_NEWUSER (Linux 3.8) user & group ID space
- cgroups (resource consumption throttling & accounting)

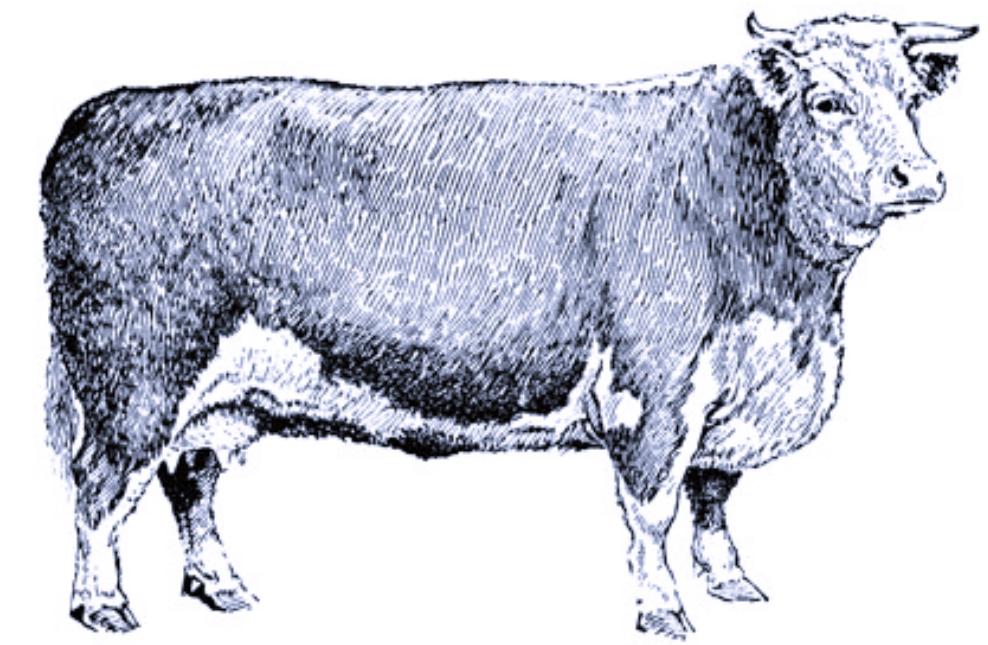
CONTAINER 101



<http://containerz.info>

WRITING APPS

No comments, no documentation but 20 tickets



The Guy Who
Wrote This Is Gone

It's running everywhere

O RLY[?]

FML

application orchestration

CloudFoundry, Heroku, OpenShift

container orchestration

Kubernetes, Marathon, Swarm, Nomad, Firmament

job scheduling

Chronos, Kubernetes, Nomad

containerization

Docker, OCI/runc, rkt/appc, Apache Mesos

resource management

Apache Mesos

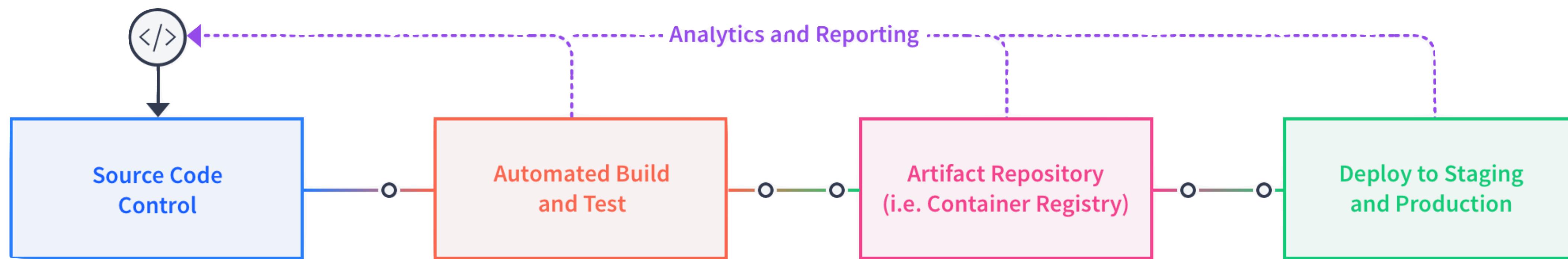
base provisioning

Vagrant, Puppet, Chef, Ansible, Salt

machine management

AWS, Azure, Google Cloud, OpenStack, vSphere

CI/CD PIPELINES



ARTEFACTS LAYERING

examples

layer

Marathon app spec, Kubernetes
RC, Docker compose

runtime

Dockerfile, OCI, rkt

packaging

.scala, .go, .py, .sh, etc.
and config files

source artifacts

VERSION CONTROL ALL THE THINGZ!

- code *and* infrastructure
- ideally: use a DVCS ([GitHub](#), [Gitlab](#))
- immutability (see also <http://p24e.io>)
- be careful with sensitive bits (credentials)

HOW TO BUILD YOUR ARTEFACTS

1st generation

- Jenkins
<https://jenkins.io/>
- Bamboo
<https://www.atlassian.com/software/bamboo>
- TeamCity
<https://www.jetbrains.com/teamcity/>

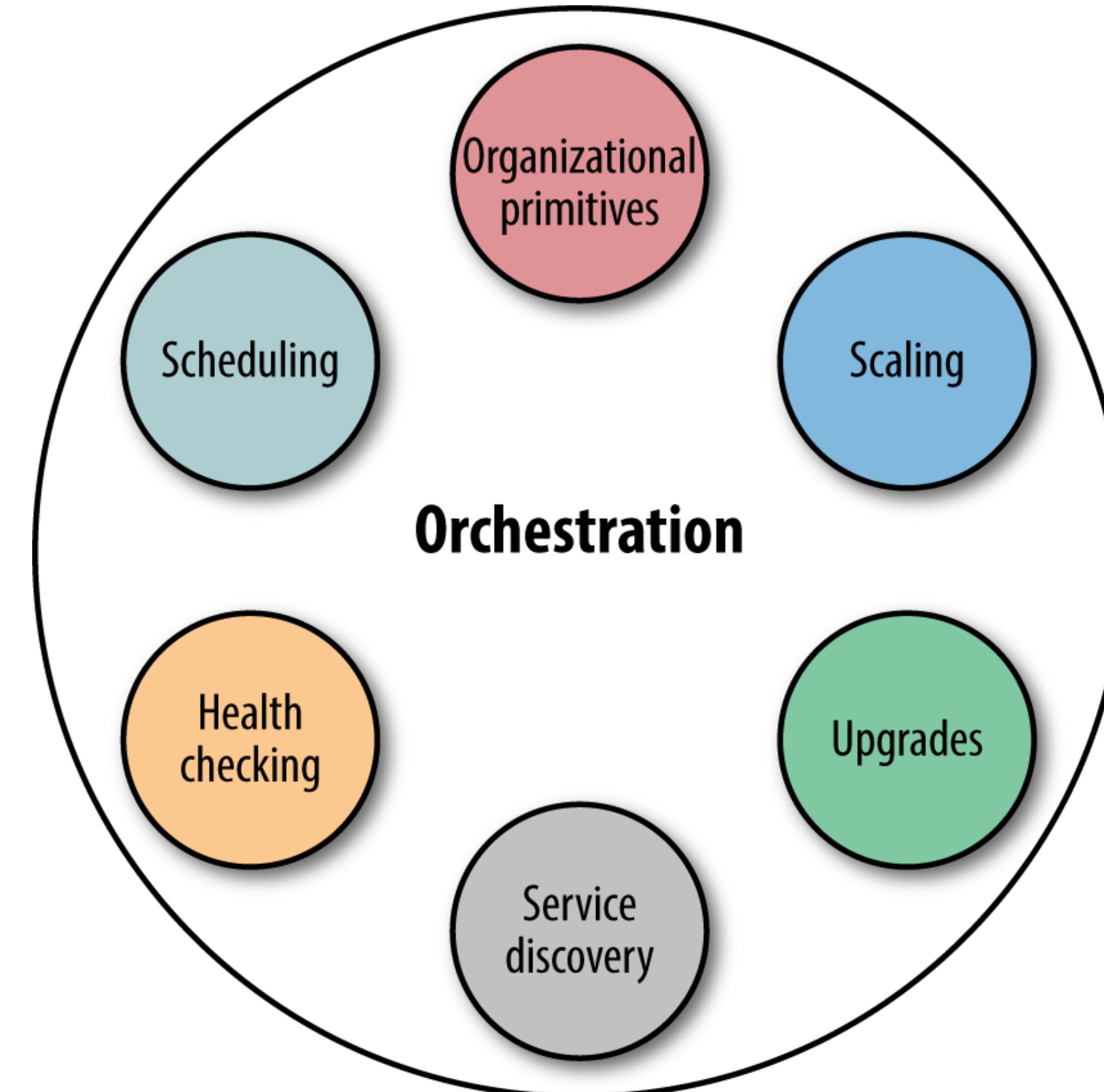
2nd generation

- Travis
<https://travis-ci.org/>
- CircleCI
<https://circleci.com/>
- concourse
<https://concourse.ci/>

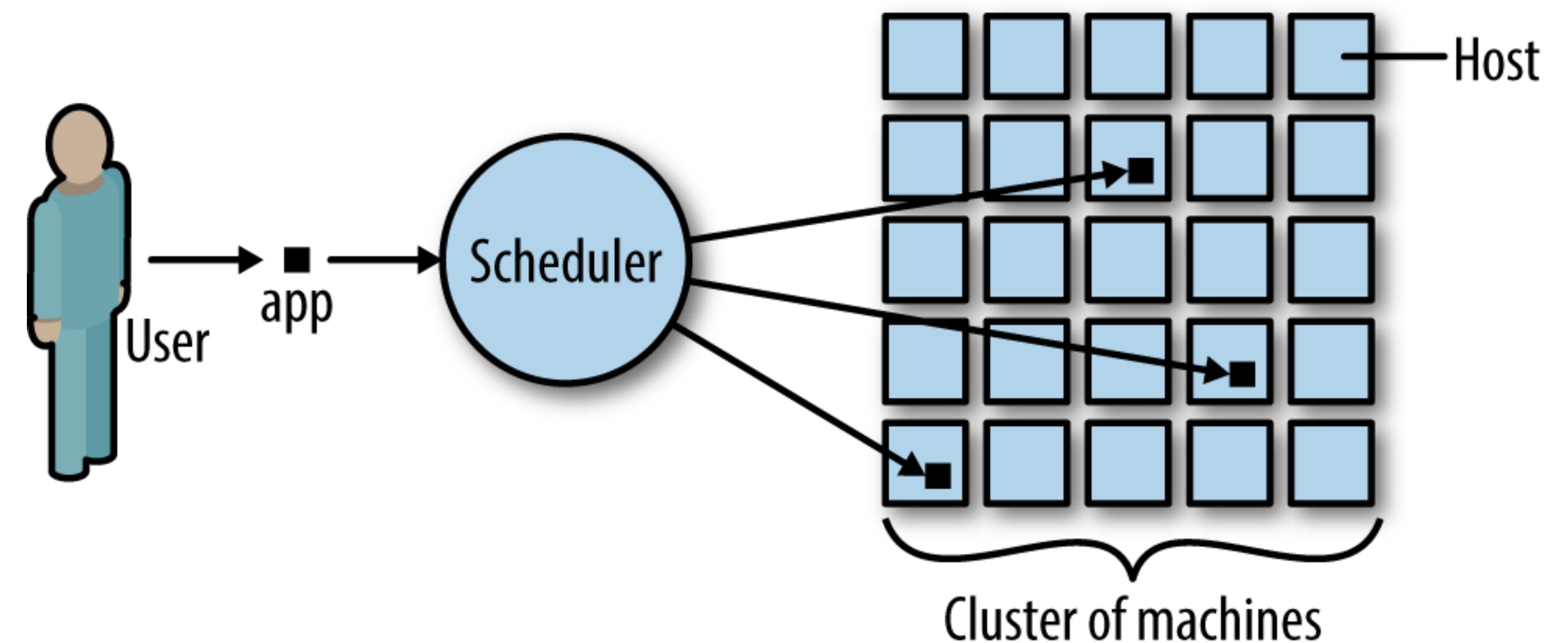
REGISTRIES: WHERE TO PUT YOUR ARTEFACTS

- Docker Hub
<https://hub.docker.com/>
- Google Cloud
<https://cloud.google.com/tools/container-registry/>
- AWS
<https://aws.amazon.com/ecr/>
- CoreOS
<https://quay.io/>
- SUSE Portus
<http://port.us.org/>
- JFrog Artifactory
<https://www.jfrog.com/artifactory/>
- Run your own
<https://docs.docker.com/registry/deploying/>

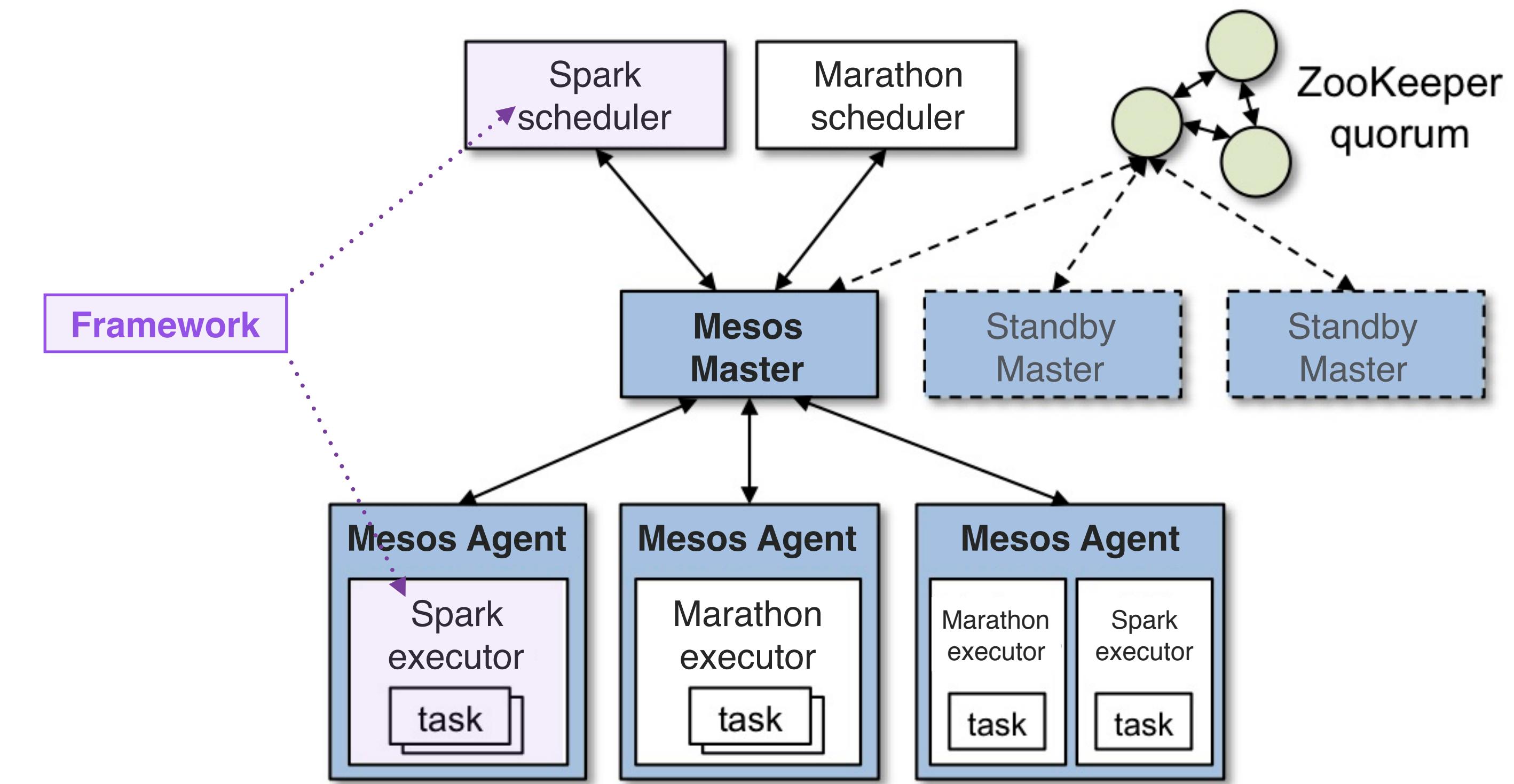
CONTAINER ORCHESTRATION



WHAT DOES A SCHEDULER DO?

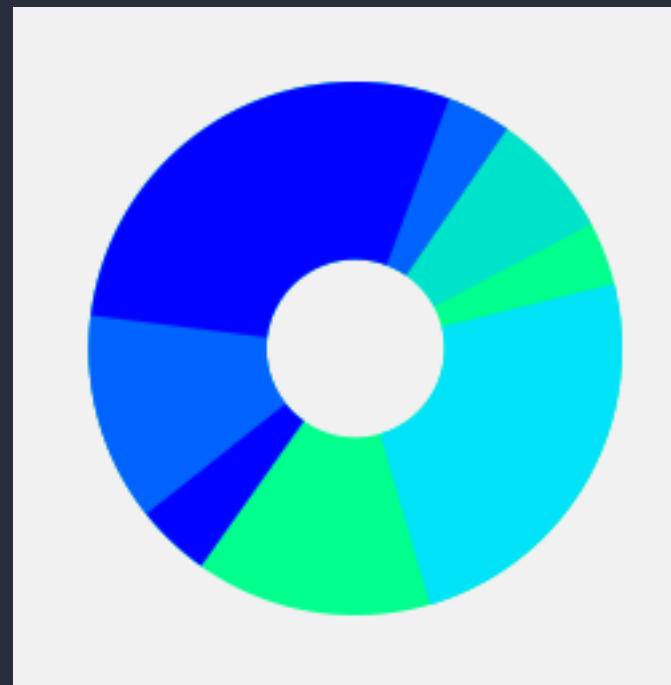


MESOS ARCHITECTURE



mesos.berkeley.edu/mesos_tech_report.pdf

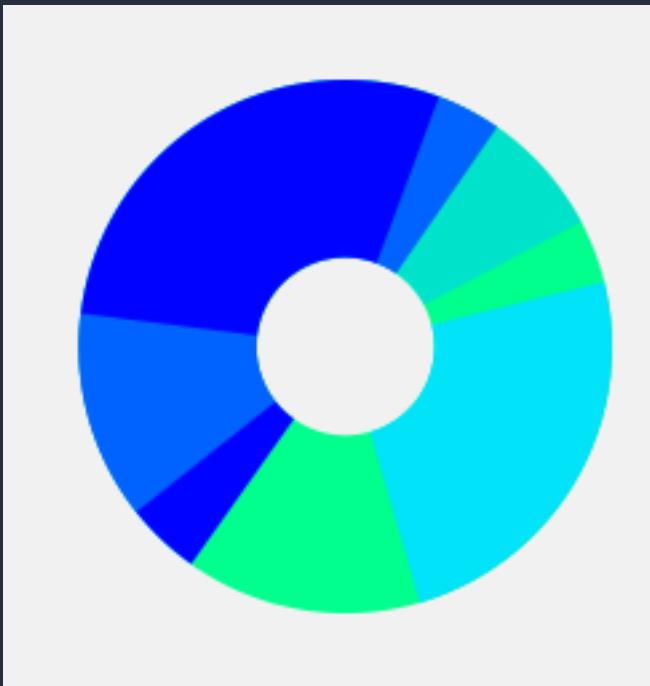
MARATHON



Also acts as the distributed init system for DC/OS

- starts instances of a long-running services
- restarts the instances if they crash
- supports health checks
- supports multitude of upgrade strategies
- HA built in

MARATHON ORGANIZATIONAL PRIMITIVES



group

app

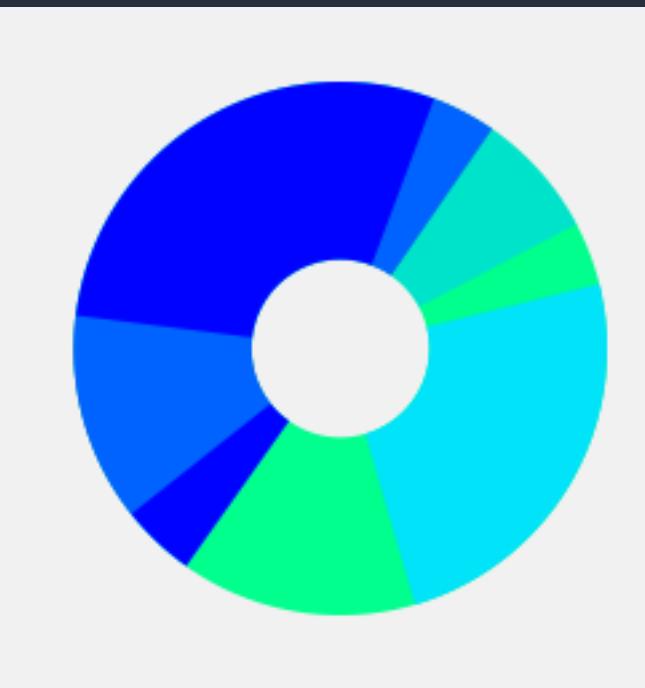
dependency



- groups can contain one or more apps/groups
- good for dependency management/scaling
- labels → good for non-hierarchical organization

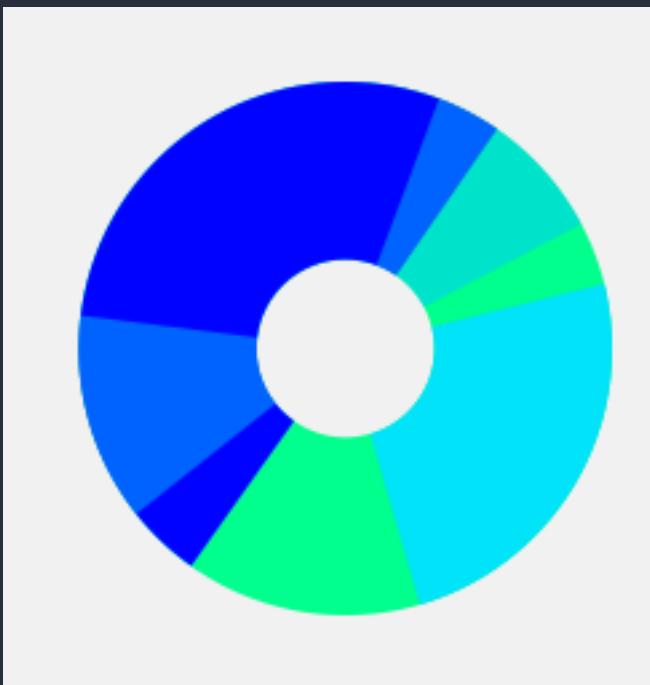
mesosphere.com/blog/2015/06/21/web-application-analytics-using-docker-and-marathon/

MARATHON APP SPEC



```
{  
  "id": "/webserver",  
  "instances": 5,  
  "cpus": 0.1,  
  "mem": 128,  
  "container": {  
    "type": "DOCKER",  
    "docker": {  
      "image": "nginx:1.9.14",  
      "network": "BRIDGE",  
      "portMappings": [  
        {  
          "containerPort": 80,  
          "hostPort": 0  
        }  
      ]  
    },  
    "upgradeStrategy": {  
      "minimumHealthCapacity": 0.9  
    },  
    "acceptedResourceRoles": [  
      "slave_public"  
    ]  
  }  
}
```

MARATHON HTTP API

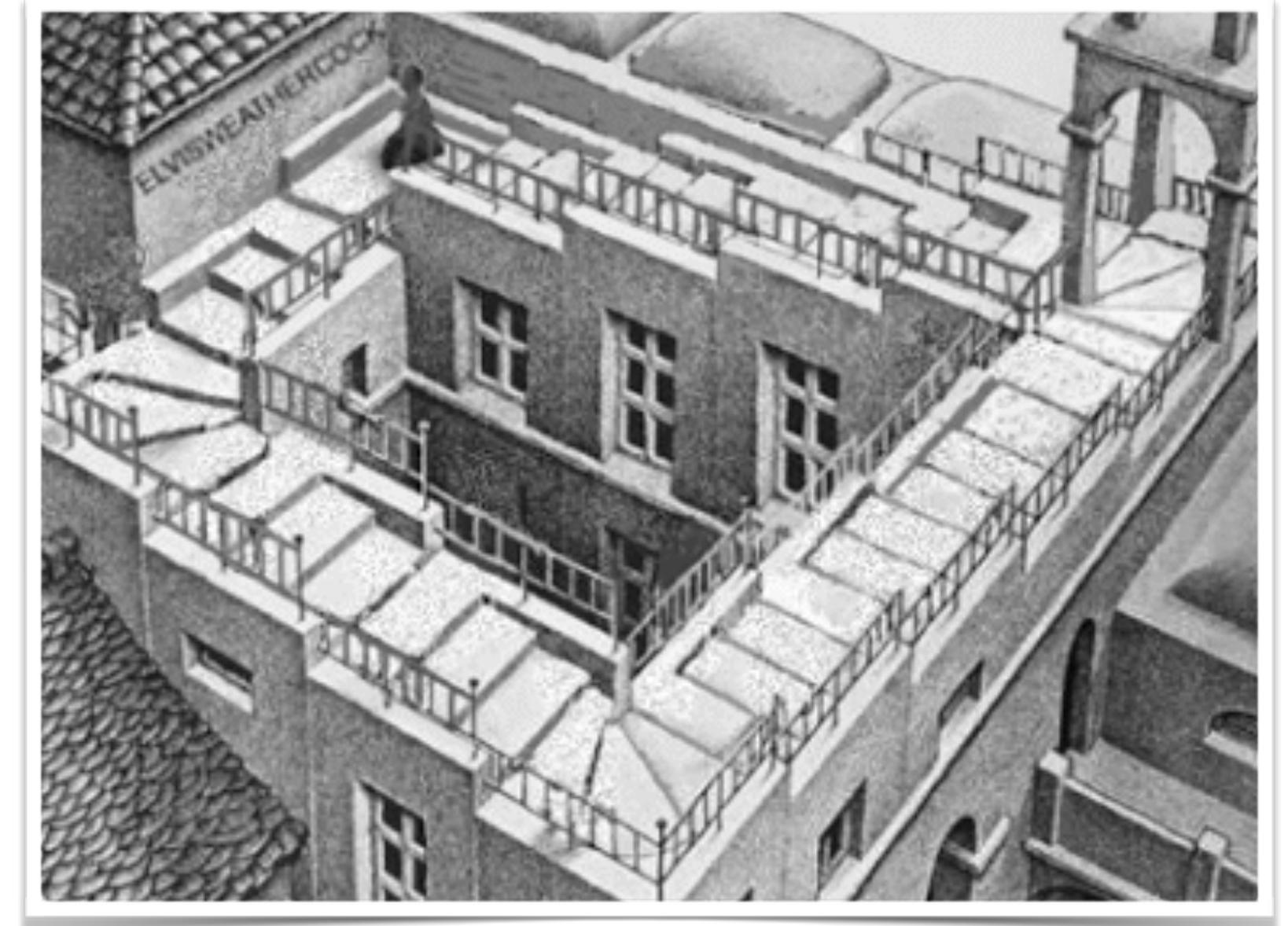


Marathon REST API documentation version 2.0

| /v2/apps | GET | PUT | POST |
|--------------------------------------|------|------|--------|
| /v2/apps | GET | PUT | POST |
| /v2/apps/{app_id} | GET | PUT | DELETE |
| /v2/apps/{app_id}/restart | | | POST |
| /v2/apps/{app_id}/tasks | GET | | DELETE |
| /v2/apps/{app_id}/tasks/{task_id} | | | DELETE |
| /v2/apps/{app_id}/versions | GET | | |
| /v2/apps/{app_id}/versions/{version} | GET | | |
| /v2/deployments | GET | | |
| /v2/deployments | GET | | |
| /v2/deployments/{deployment_id} | | | DELETE |
| /v2/groups | GET | PUT | POST |
| /v2/groups | GET | PUT | POST |
| /v2/groups/versions | GET | | |
| /v2/groups/{group_id} | GET | PUT | POST |
| /v2/groups/{group_id} | GET | PUT | POST |
| /v2/groups/{group_id}/versions | GET | | |
| /v2/tasks | GET | | |
| /v2/tasks | GET | | |
| /v2/tasks/delete | | | POST |
| /v2/artifacts | POST | | |
| /v2/artifacts | POST | | |
| /v2/artifacts/{path} | PUT | POST | DELETE |
| /v2/events | GET | | |
| /v2/events | GET | | |

mesosphere.github.io/marathon/docs/generated/api.html

HANDS-ON



SERVICE DISCOVERY & DEPLOYMENTS



SERVICE DISCOVERY

DNS-based

- 😊 easy to integrate
- 😢 SRV records
- 😢 no health checks
- 😢 TTL

Proxy-based

- 😊 no port conflicts
- 😊 fast failover
- 😢 no UDP
- 😢 management of VIPs or service ports

Application-aware

- 😊 developer fully in control and full-feature
- 😢 implementation effort
- 😢 requires distributed state management (ZK, etcd or Consul)

ZERO-DOWNTIME DEPLOYMENTS

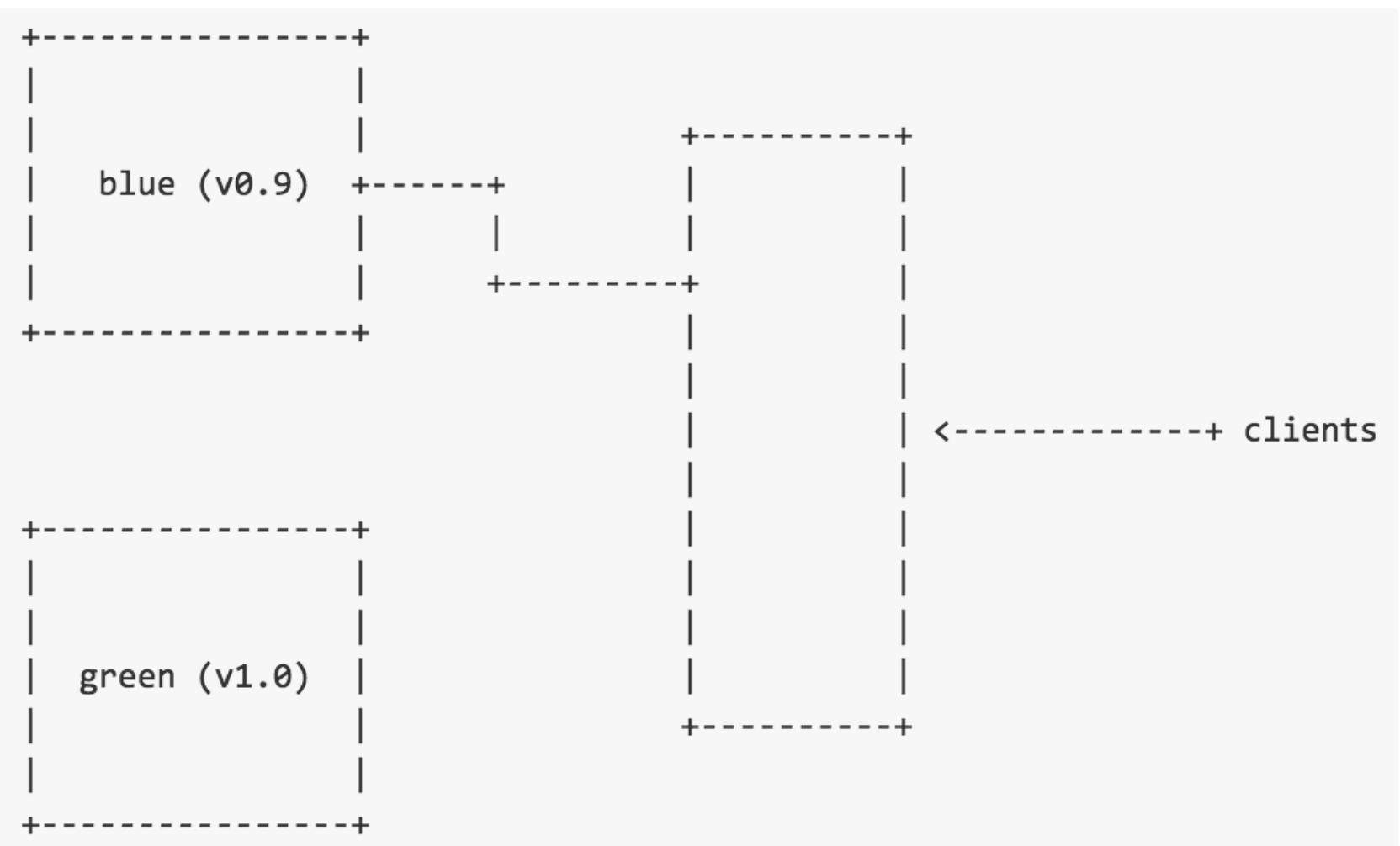
rolling deployment

bring up new version
and terminate old ones
until all old are gone

```
T0: [0.9] [0.9] [0.9] [0.9]  
  
T1: deployment kicks off  
  
T2: [0.9] [0.9] [0.9] [0.9] [1.0]  
     |  
T3: [0.9] [0.9] [0.9] [1.0] [1.0]  
     |  
T4: [0.9] [0.9] [1.0] [1.0] [1.0]  
     |  
T5: [0.9] [1.0] [1.0] [1.0] [1.0]  
     |  
T6: [1.0] [1.0] [1.0] [1.0]  
  
T7: deployment done
```

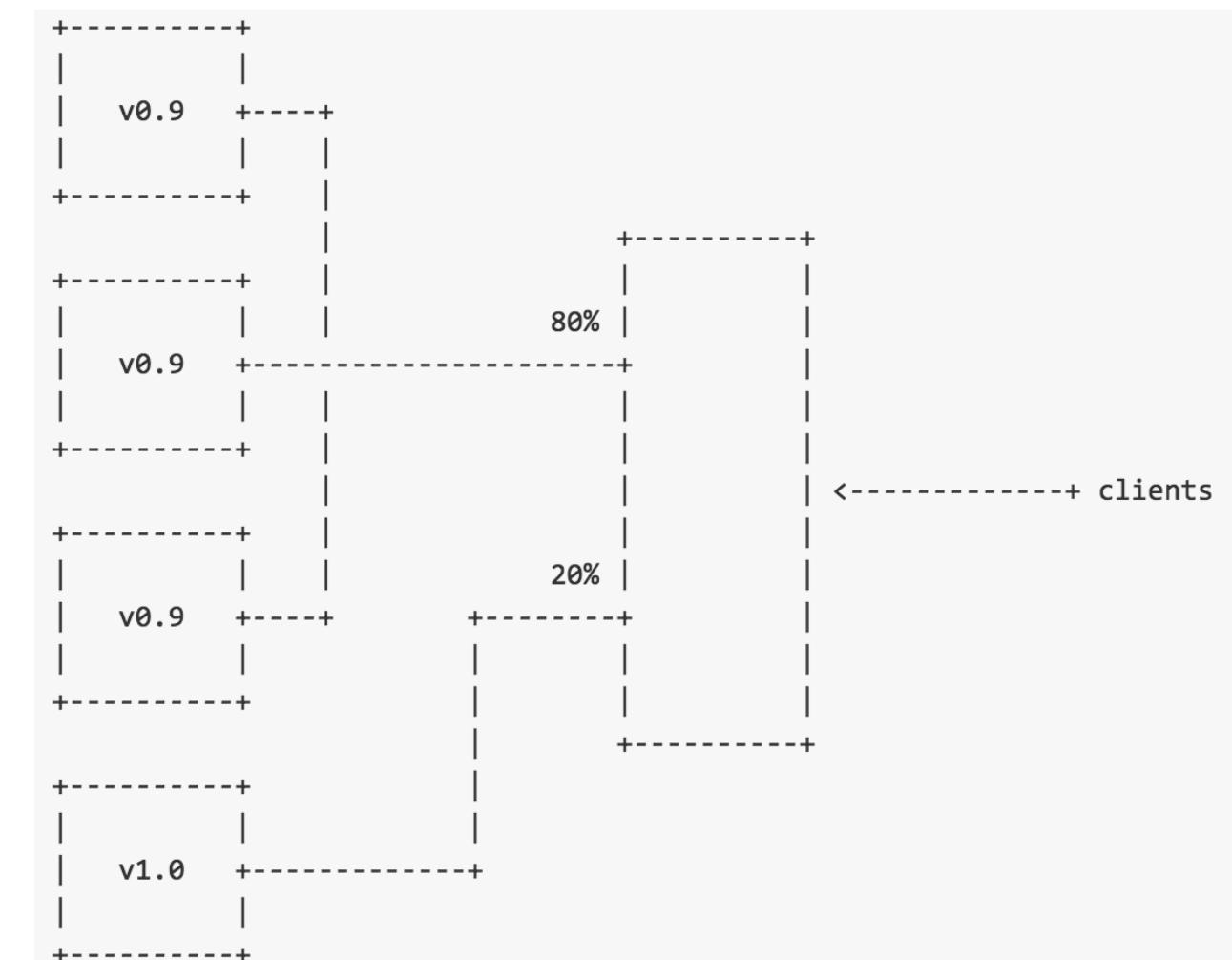
blue-green deployment

launch a new stack and
switch traffic from old to
new when new instances
are healthy

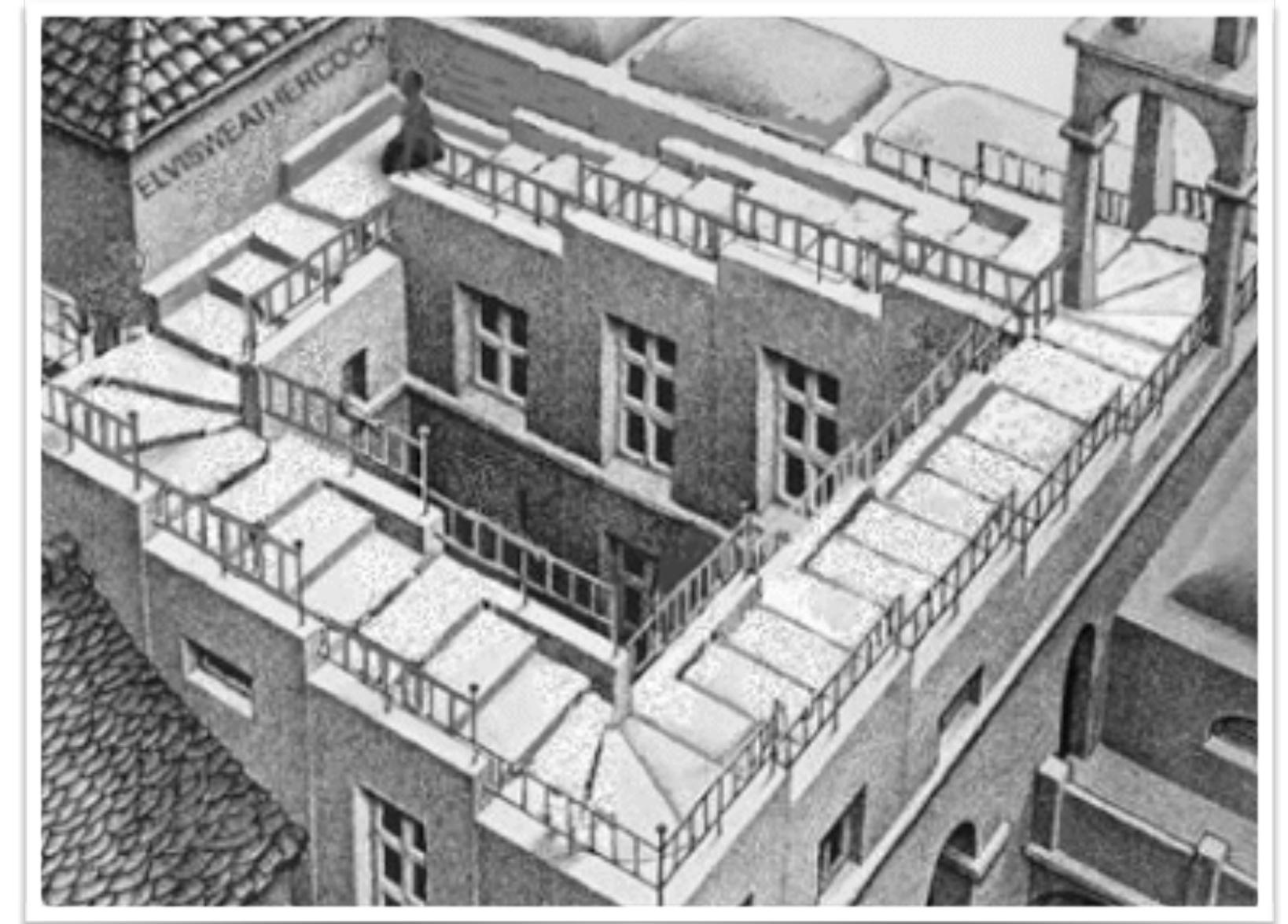


canary deployment

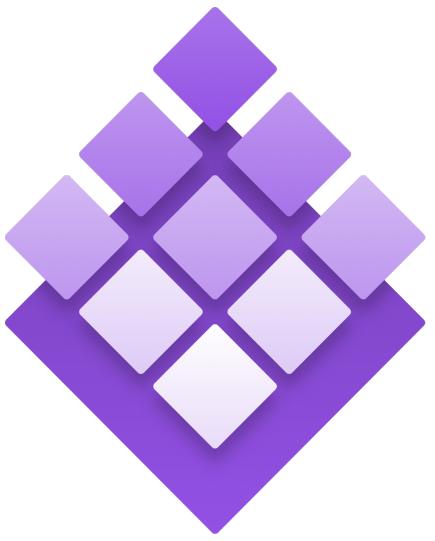
bring up a new version,
start by routing a *small portion* of traffic to the
new app, and slowly
increase



HANDS-ON

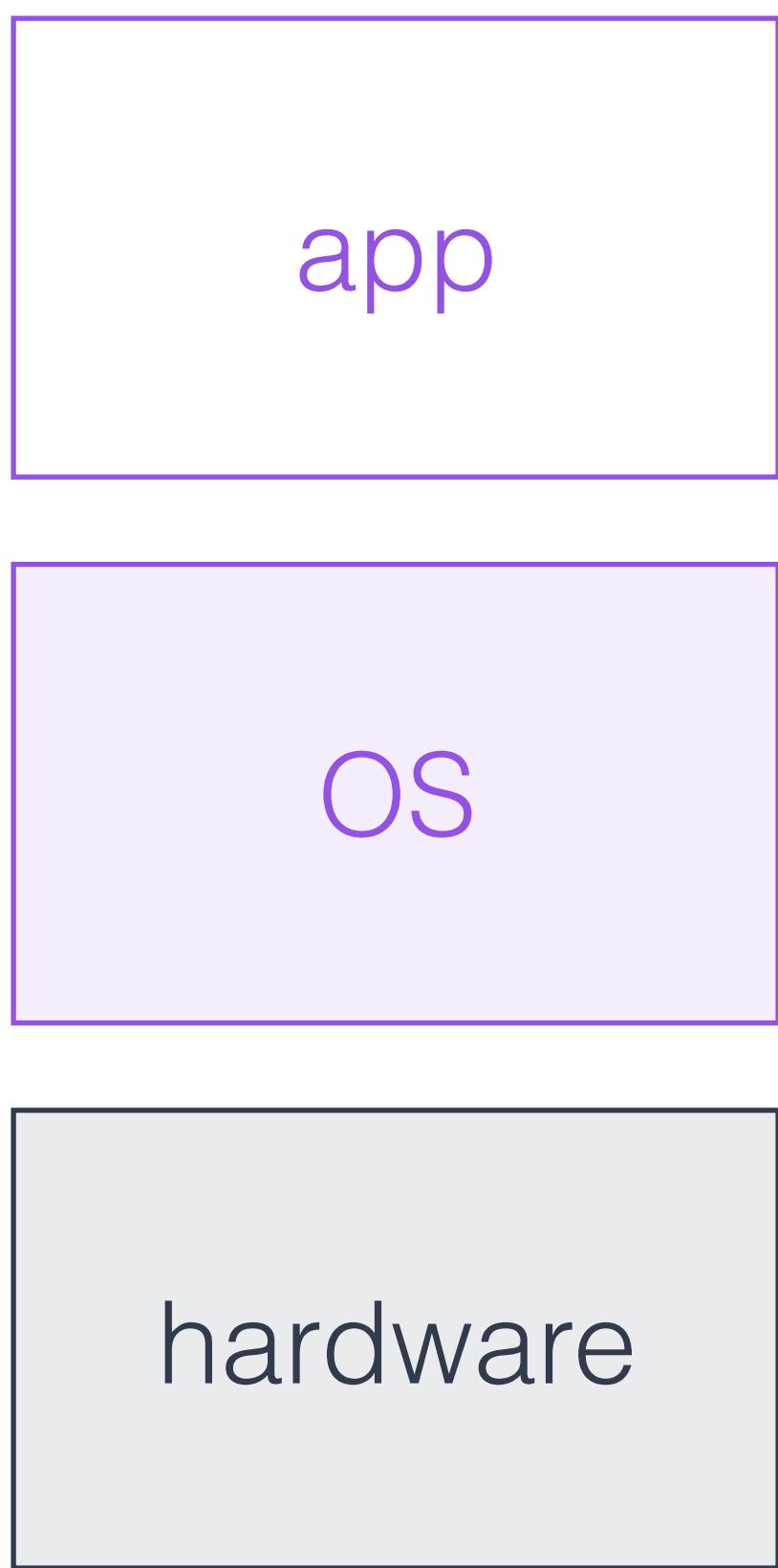


HOW DO
THINGS FIT
TOGETHER?

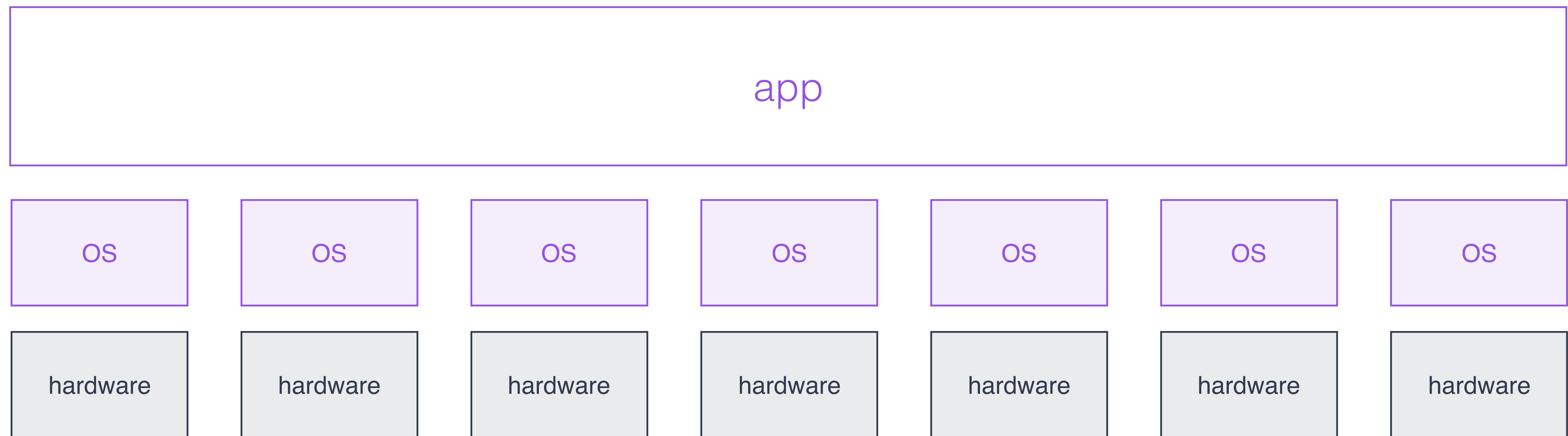


DC/OS

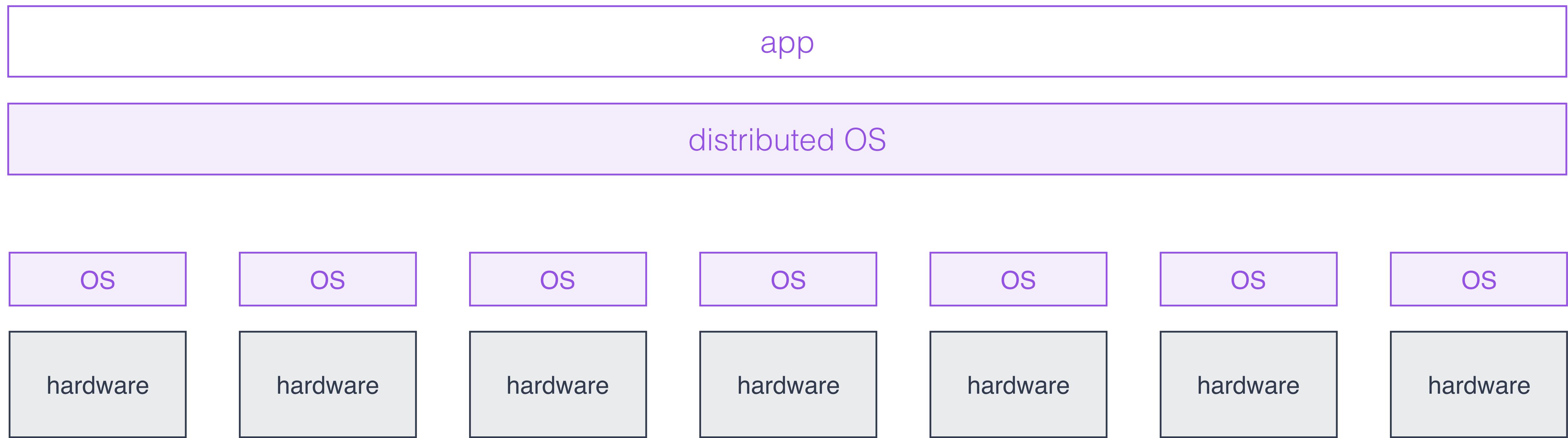
SINGLE MACHINE APPLICATION



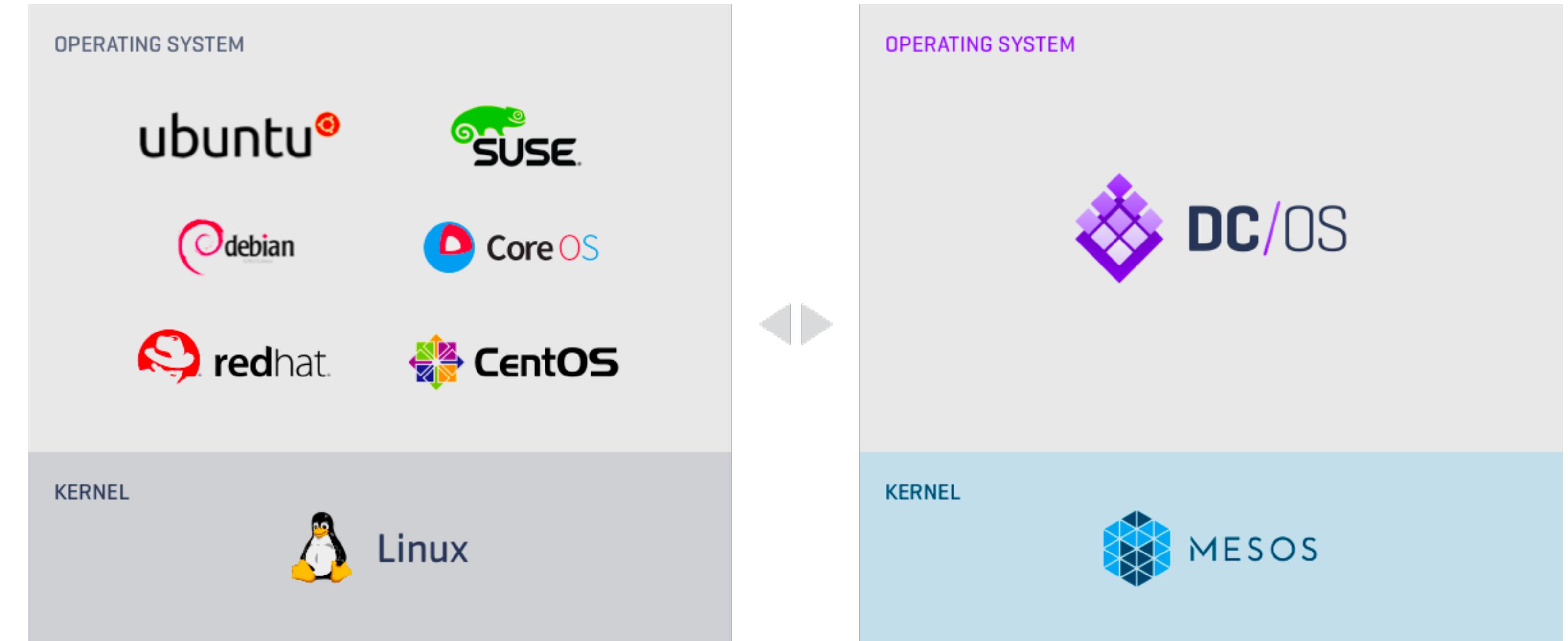
DISTRIBUTED APPLICATION



DISTRIBUTED OS + DISTRIBUTED APP



LOCAL OS VS DISTRIBUTED OS



The screenshot shows the DC/OS website homepage. At the top, there's a navigation bar with tabs for DC/OS, Marathon, Mesos, Weave Scope, and "The Definitive Platform for Modern Apps |...". Below the navigation is a dark purple header with the DC/OS logo, a search bar, and links for Why DC/OS, Install, Docs, Community, and Blog. A GitHub star icon with the number 593 is also present. A prominent white text area on the left says "The easiest way to run containers in production." with a "Get Started" button and a "Play Video" button. To the right is a large, stylized purple cube composed of smaller cubes, resting on a track. Below this are four sections: "Containers & Big Data", "100% Open Source", "Production Proven", and "Datacenter Apps", each with a small icon and a brief description.

The easiest way to run containers in production.

Get Started

Play Video

Containers & Big Data

Easily deploy and run stateful or stateless distributed workloads including Docker containers, Big Data, and

100% Open Source

Run DC/OS in production without any scale or performance limitations, on any on-premises or cloud

Production Proven

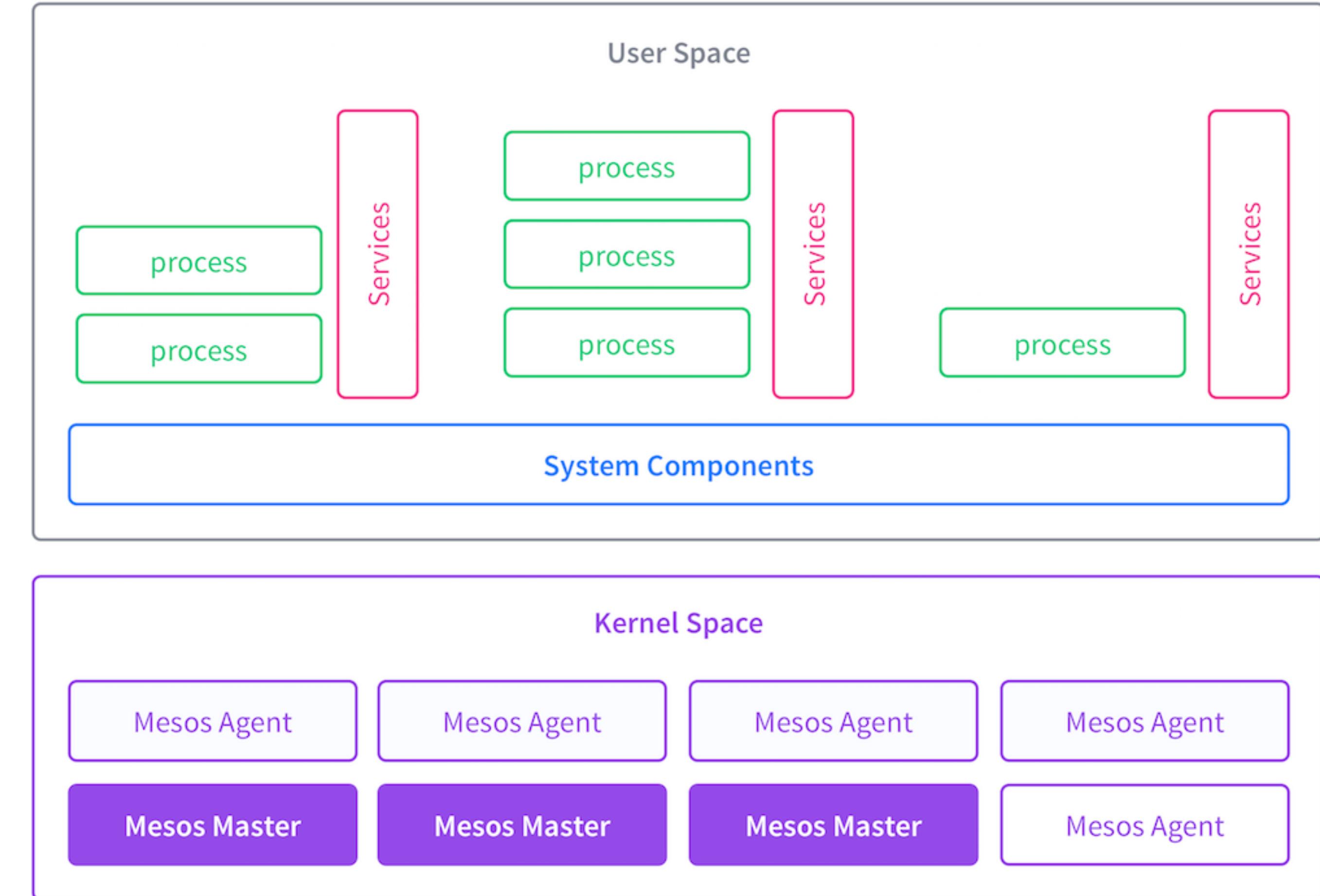
Built on Apache Mesos with the experience of Mesosphere, Yelp, Twitter, Airbnb, and many of today's most innovative

Datacenter Apps

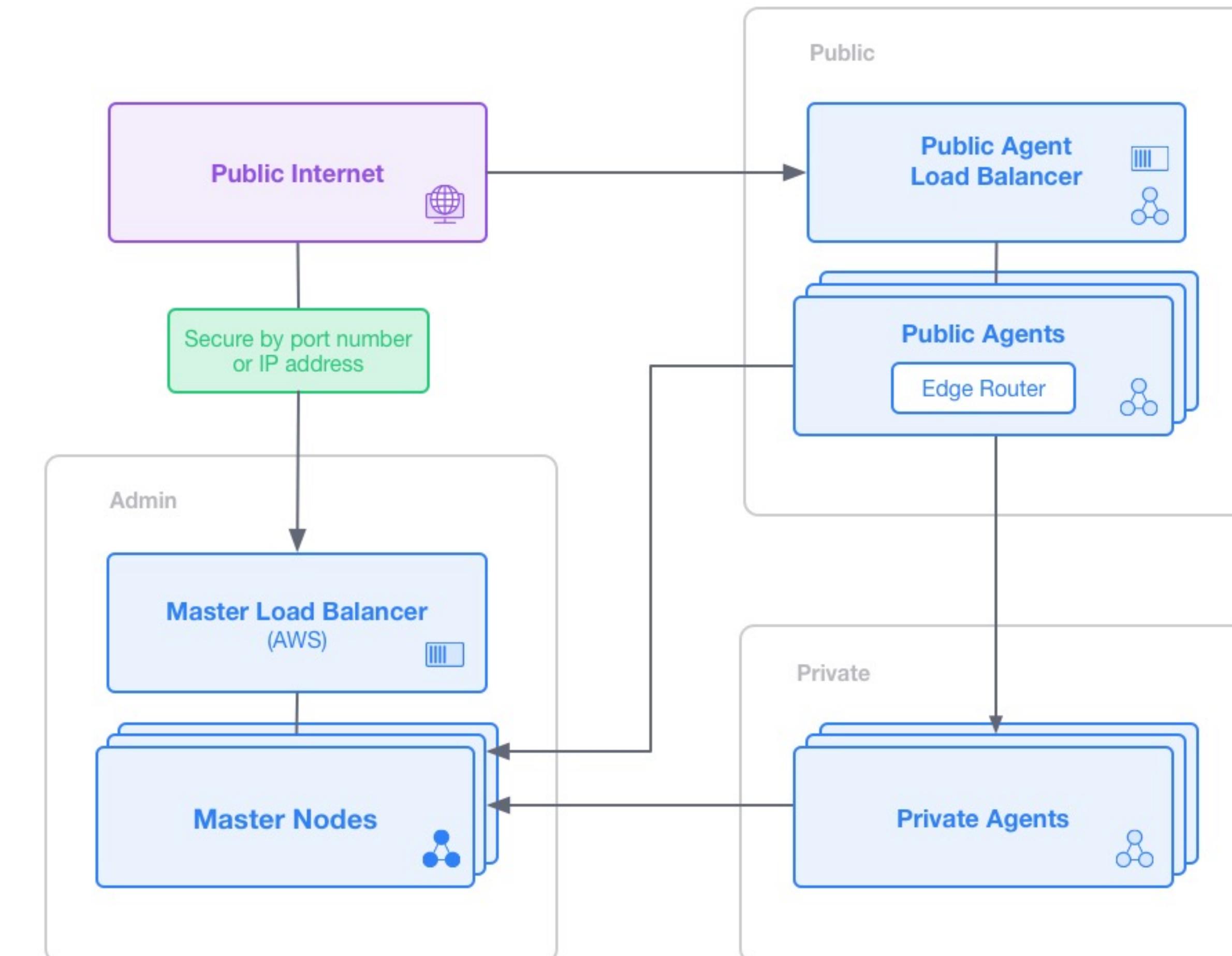
Build apps quickly using a rich ecosystem of datacenter-scale application services.

<https://dcos.io>

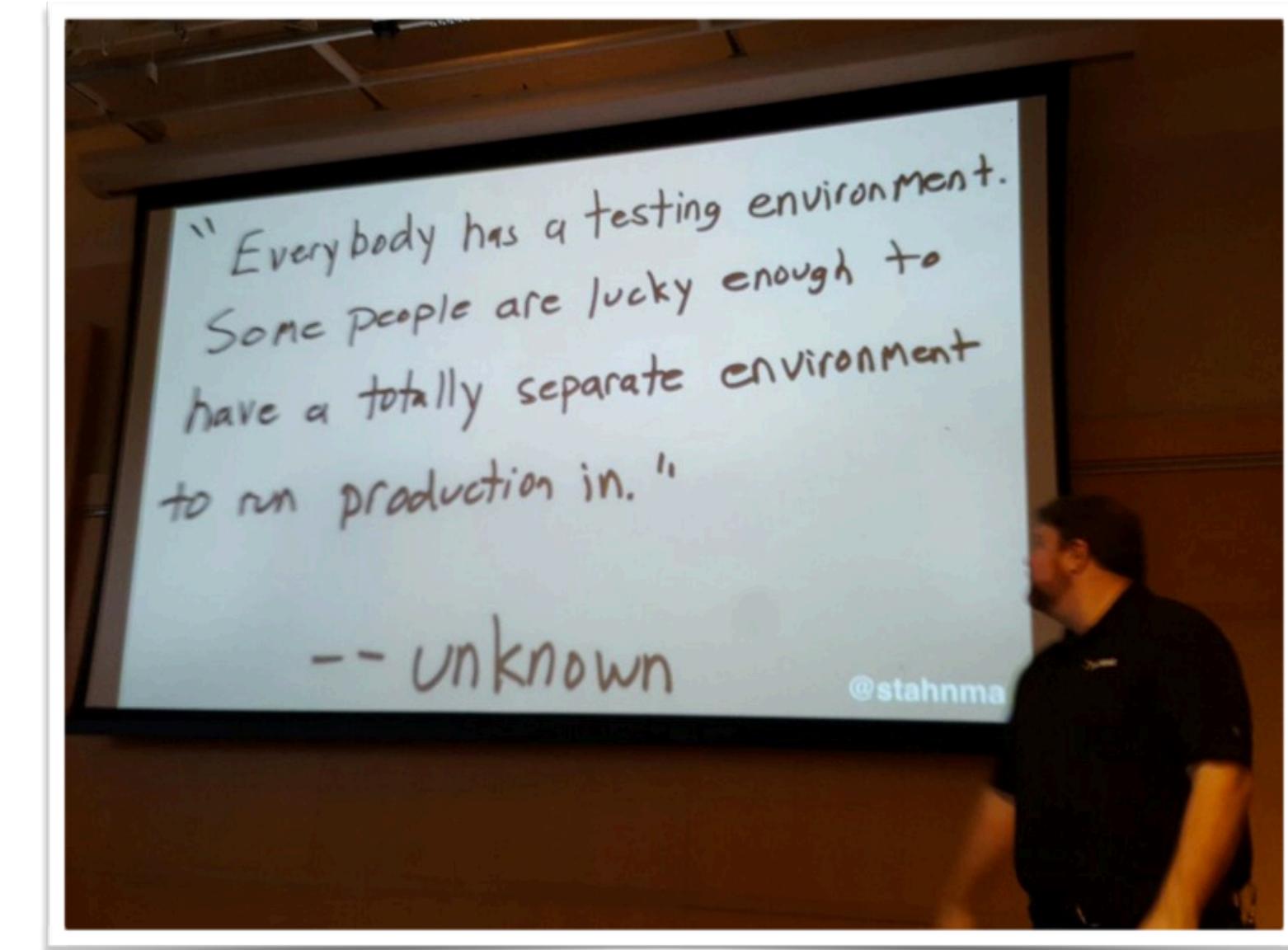
DC/OS ARCHITECTURE



DC/OS ARCHITECTURE



DEVELOPING IN A DISTRIBUTED SETUP



ALTERNATIVE WAYS TO DEVELOP IN A DISTRIBUTED SETUP



ALTERNATIVE WAYS TO DEVELOP IN A DISTRIBUTED SETUP

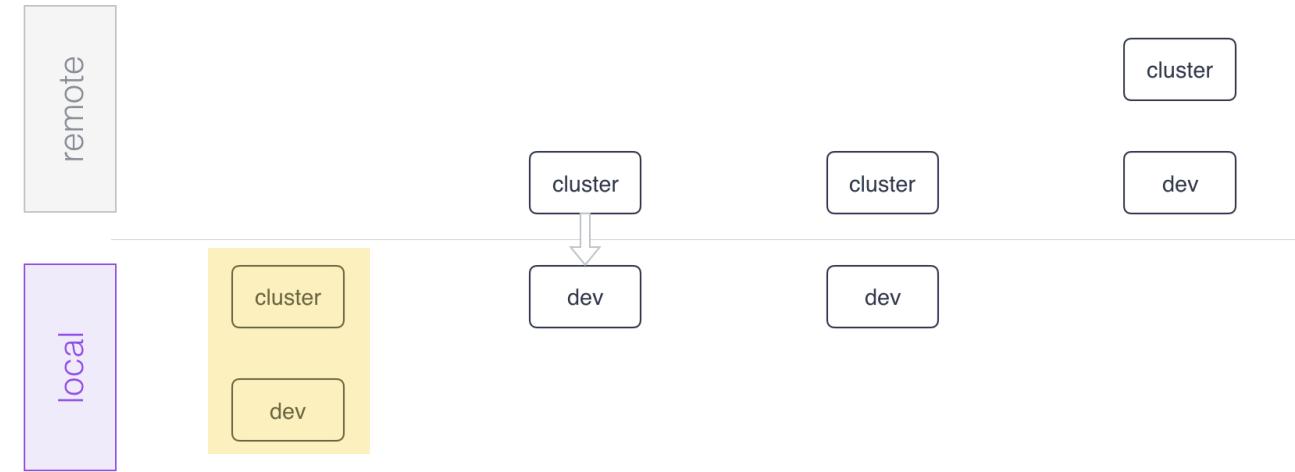
Minikube, Minishift

Docker Mac/Windows

DC/OS Vagrant

pure off-line

- 😊 no need to maintain cluster
- 😊 no costs
- 😊 off-line and fast dev cycles
- 😢 limited scale
- 😢 not really a distributed system
- 😢 not all functionality might be available



ALTERNATIVE WAYS TO DEVELOP IN A DISTRIBUTED SETUP

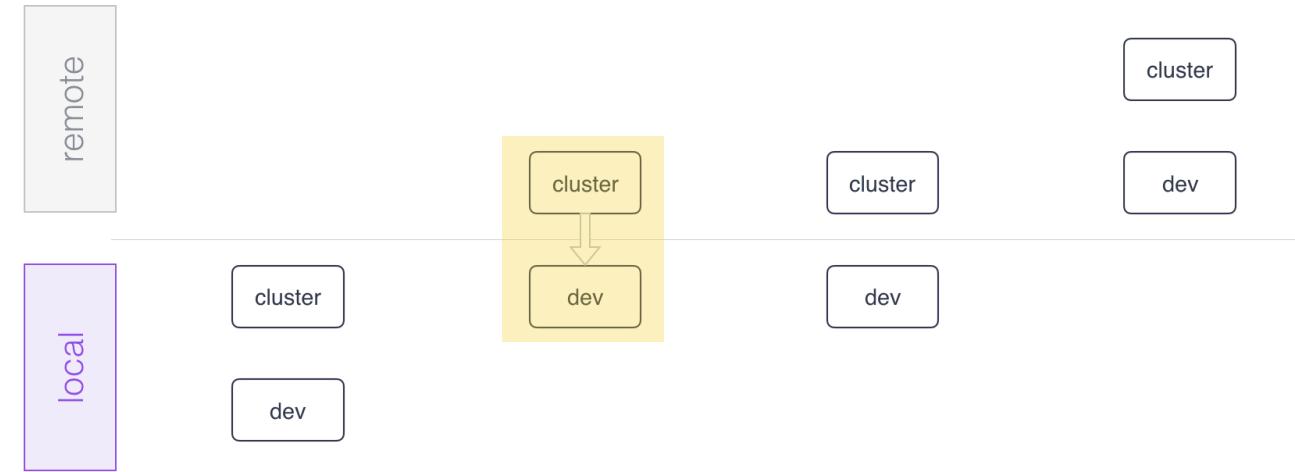
kubectl proxy/apiserver proxy

DC/OS Tunnel

proxied

- 😊 almost the real thing (distributed system)
- 😊 fast dev cycles

- 🙁 have to maintain cluster/cost
- 🙁 limited off-line development



ALTERNATIVE WAYS TO DEVELOP IN A DISTRIBUTED SETUP



Quay.io + TeamCity + Tectonic

😊 the real thing (distributed system)

GitLab CR + Jenkins + OpenShift

- :(have to maintain cluster/cost
- :(very limited off-line development
- :(slow dev cycles

Artifactory + Bamboo + Marathon

ALTERNATIVE WAYS TO DEVELOP IN A DISTRIBUTED SETUP

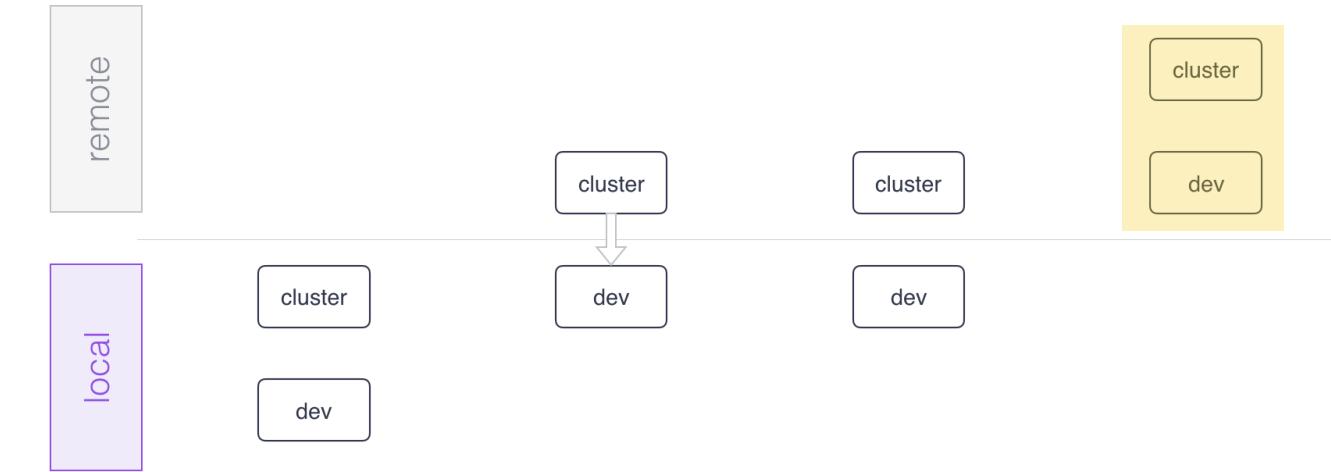
Google Cloud Shell/GCR/GKE

Katacoda

Koding

pure on-line

- 😊 the real thing (distributed system)
- 😊 low/no costs
- 😊 scales well
- 😢 no off-line development
- 😢 limited customization/control



CHALLENGES & OPPORTUNITIES



CHALLENGES

- No or long feedback cycles
- Creating new stuff more valued than keeping it alive
- Need to move fast
 - software is eating the world
 - 24/7 online mentality
 - expectations from consumer apps determine enterprise apps

appops

The person who writes an app is also the person responsible for operating the app in prod.



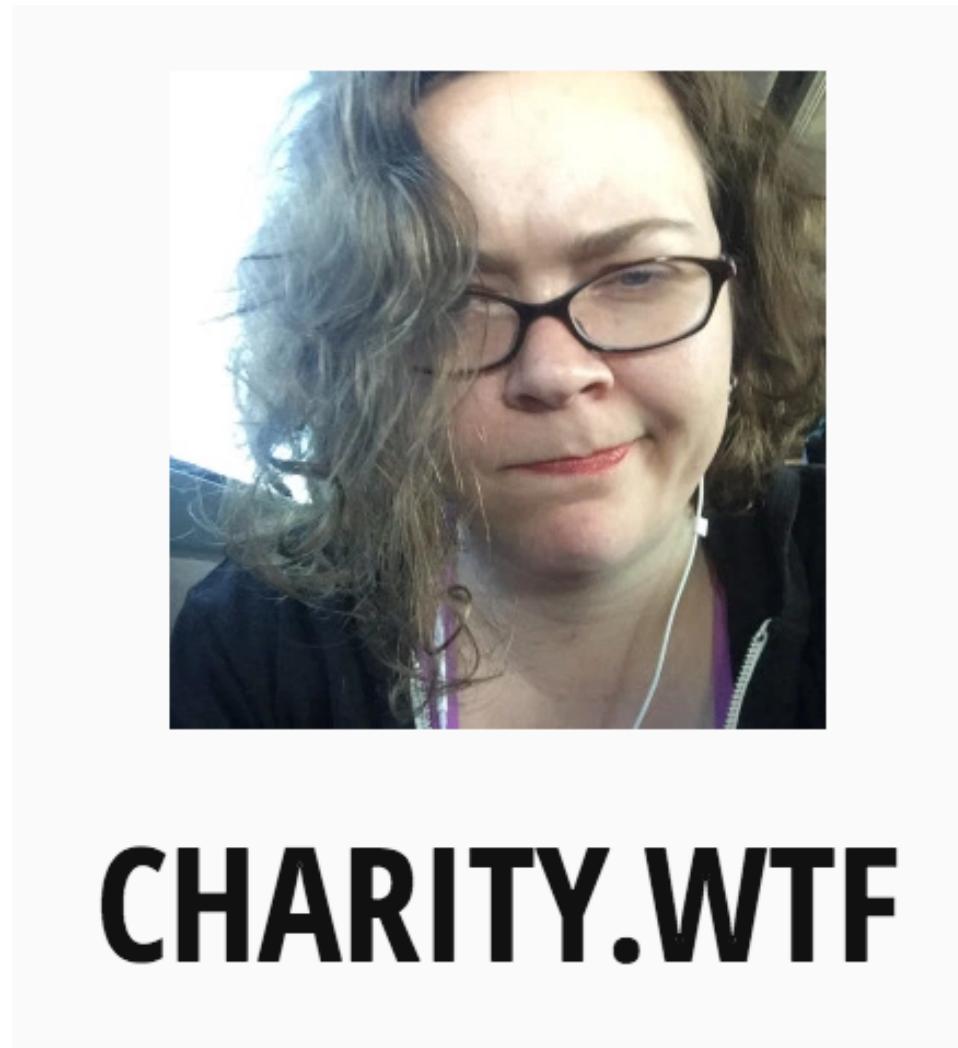
appops

It's **not** about provisioning
a VM or installing a DC/OS
cluster or replacing a faulty
HDD ...

... this would be on the
infrastructure team.



appops



speakerdeck.com/charity/devops-for-developers-building-an-effective-ops-org-1

ELASTIC DATA PIPELINES

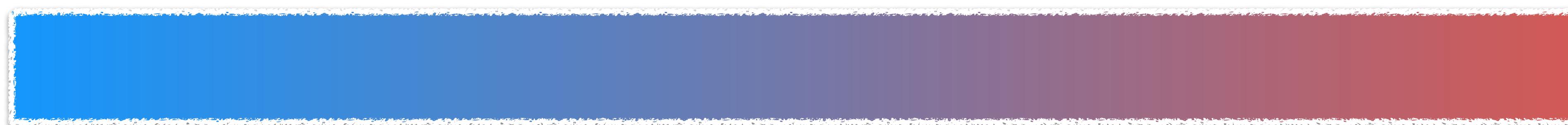


LET'S TALK ABOUT WORKLOADS* ...

batch

streaming

PaaS



CHRONOS



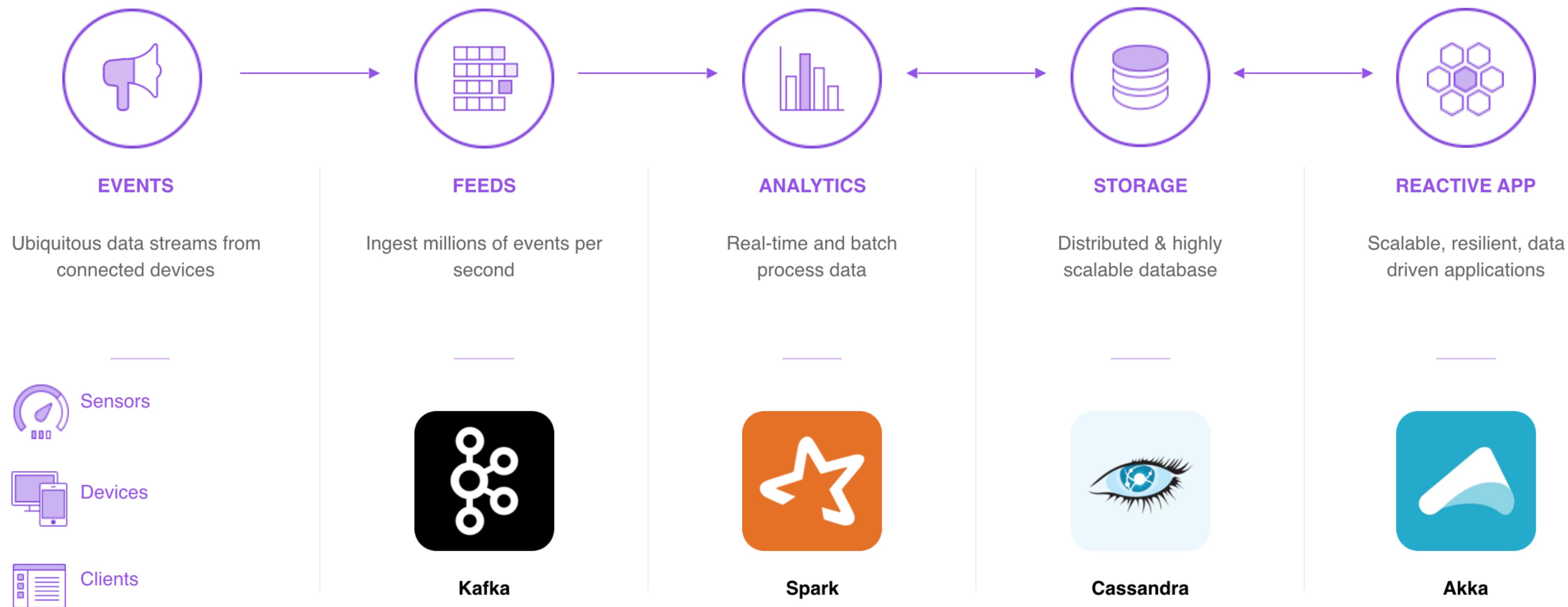
samza



CHALLENGES

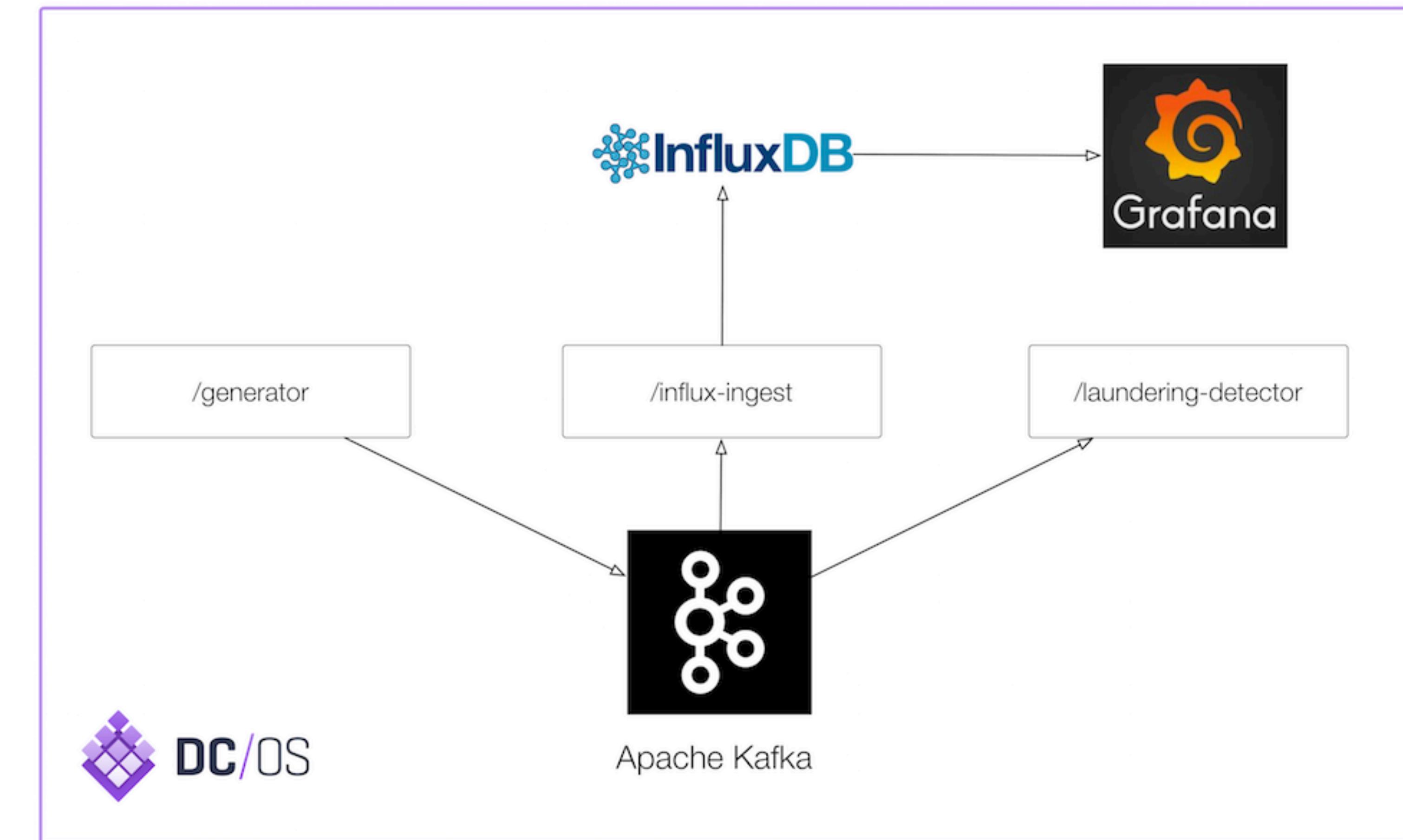
- Set up and **operation** of components
- **Elasticity**: static vs. dynamic partitioning
- **Efficient usage** of resources (utilization/TCO)

ELASTIC DATA PIPELINES WITH DC/OS

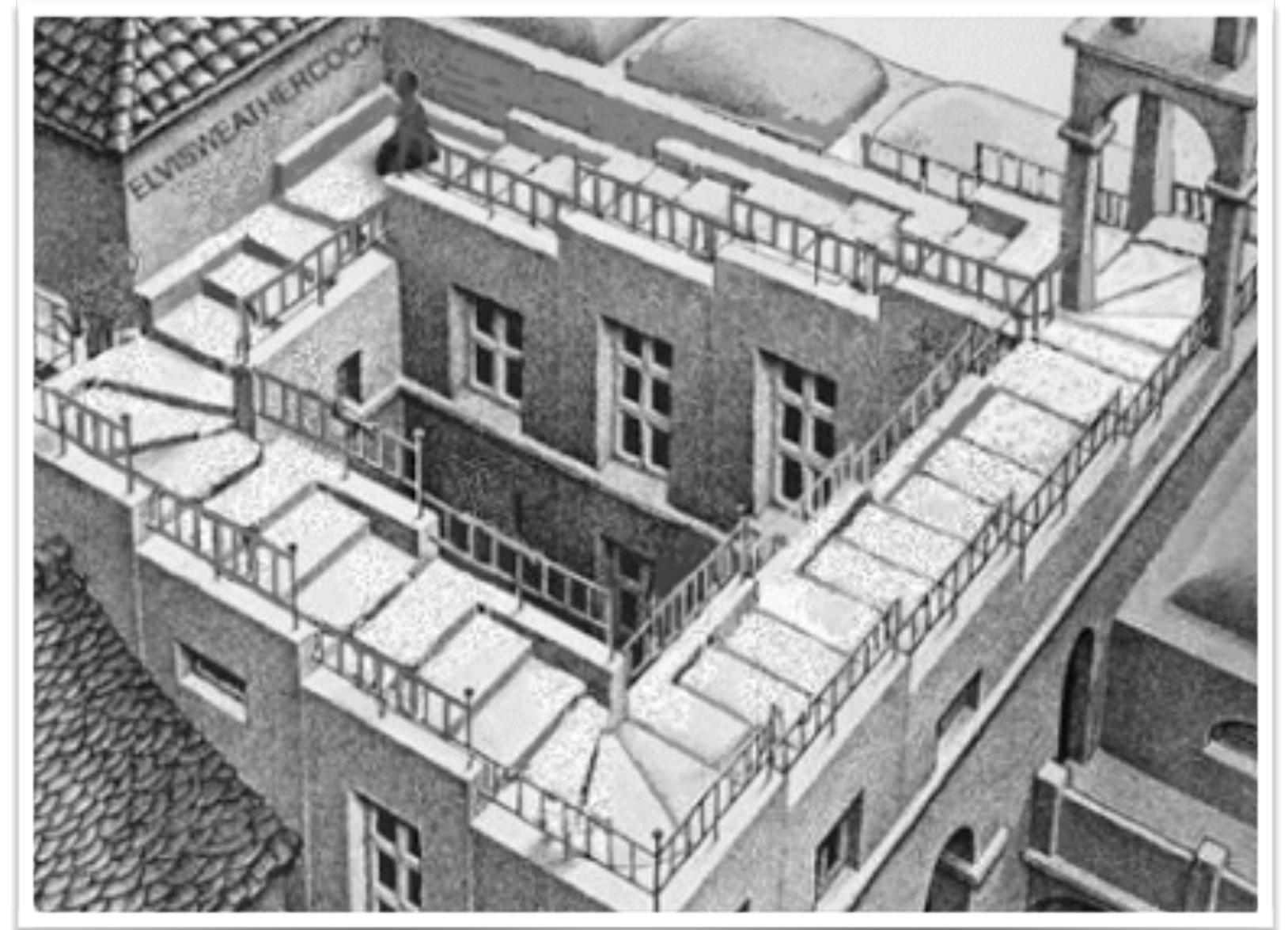


EXAMPLE: FINANCIAL DATA PROCESSING

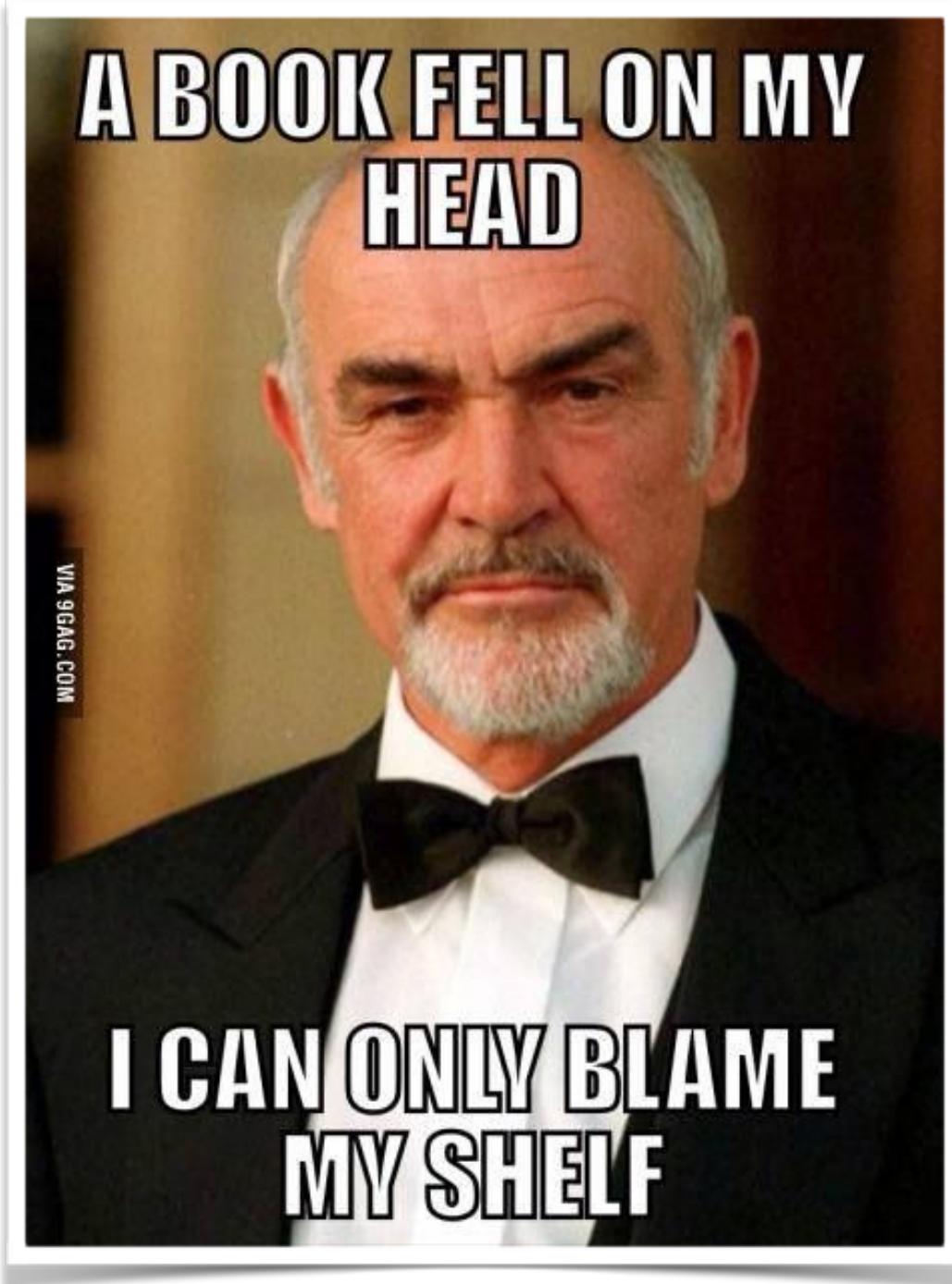
github.com/dcos/demos/tree/master/1.9/fintrans



HANDS-ON

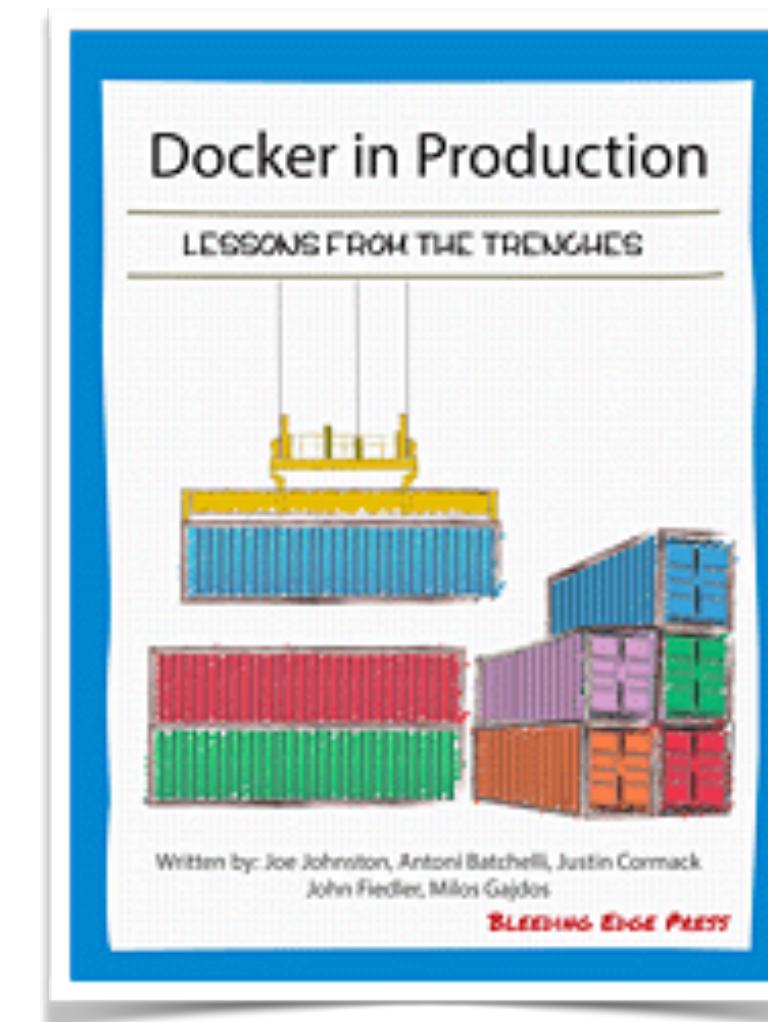


LEARNING RESOURCES

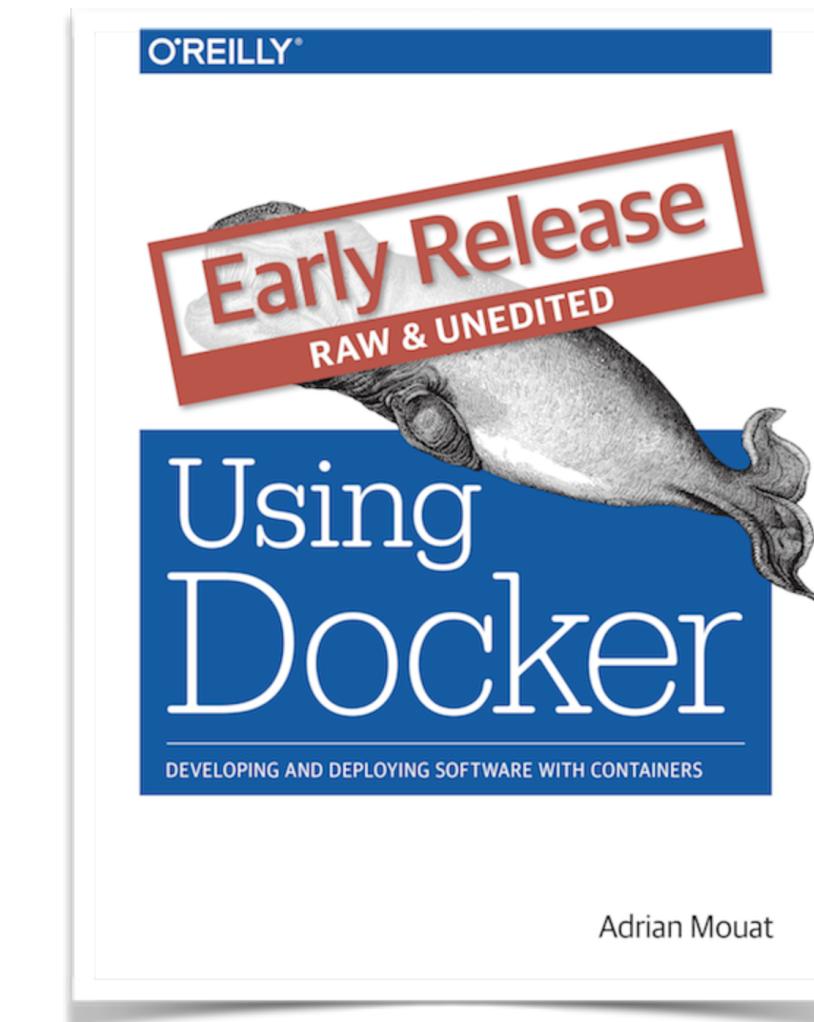


WHERE CAN I LEARN MORE?

<http://shop.oreilly.com/product/9781939902184.do>



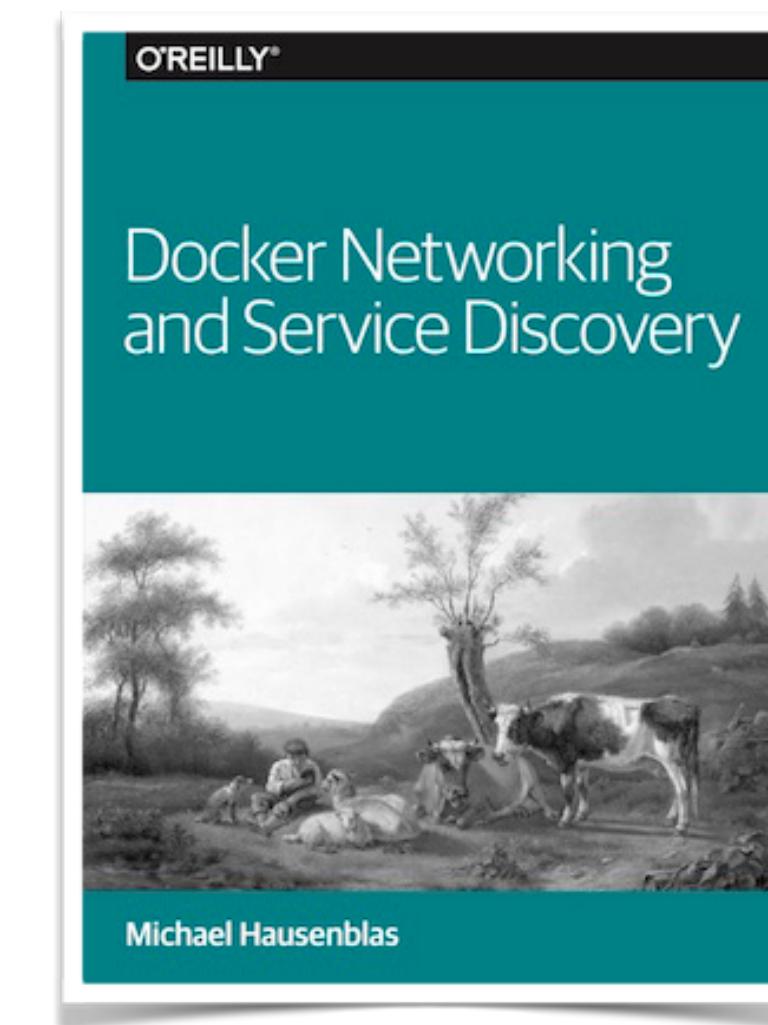
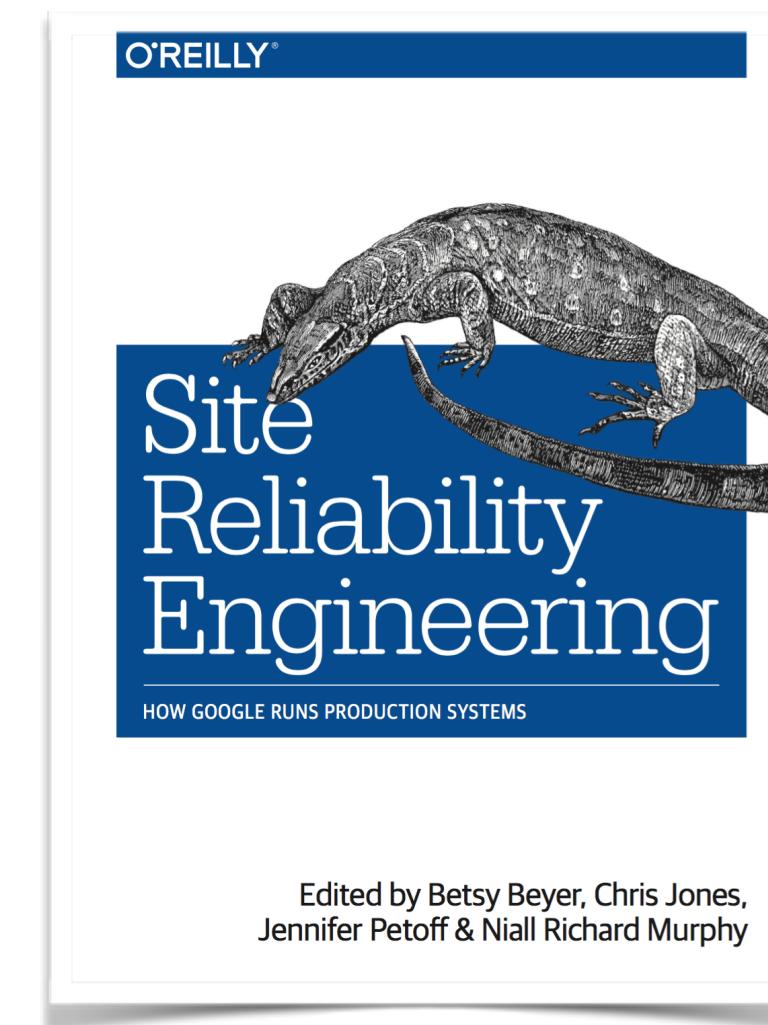
<http://shop.oreilly.com/product/0636920035671.do>



WHERE CAN I LEARN MORE?

<http://301.sh/ora2016-dnsd>

<http://301.sh/ora2016-dnsd>

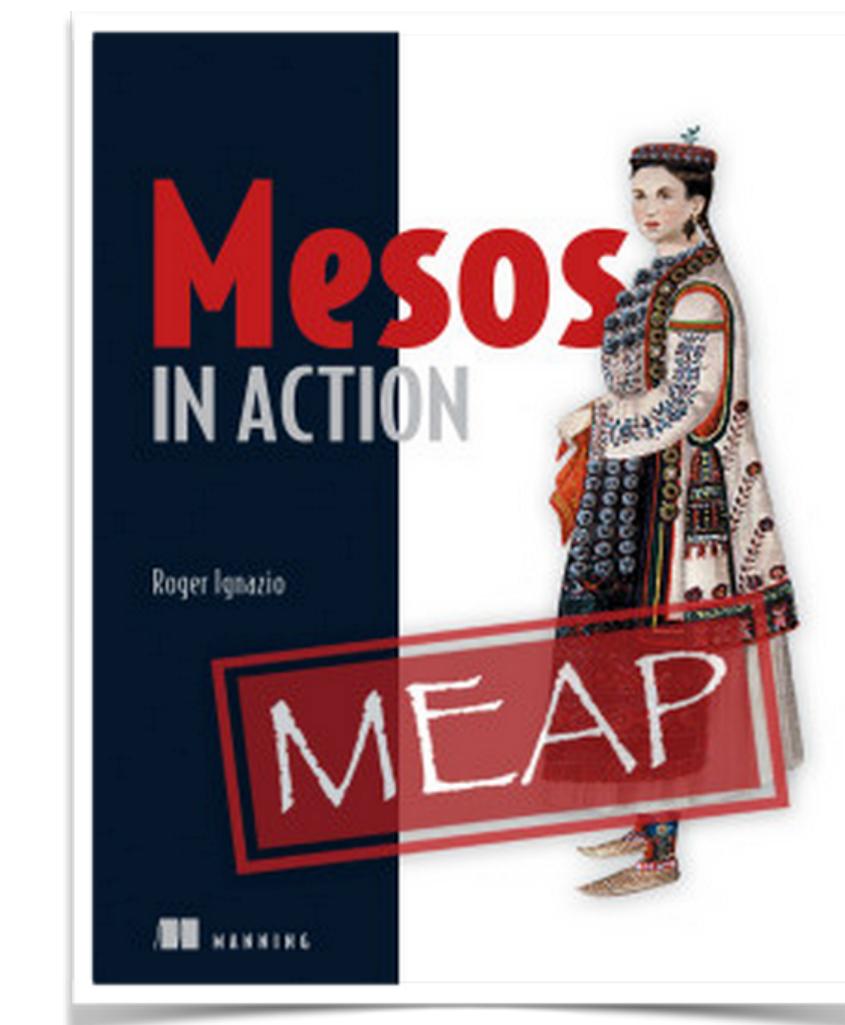


WHERE CAN I LEARN MORE?

<http://shop.oreilly.com/product/0636920039952.do>

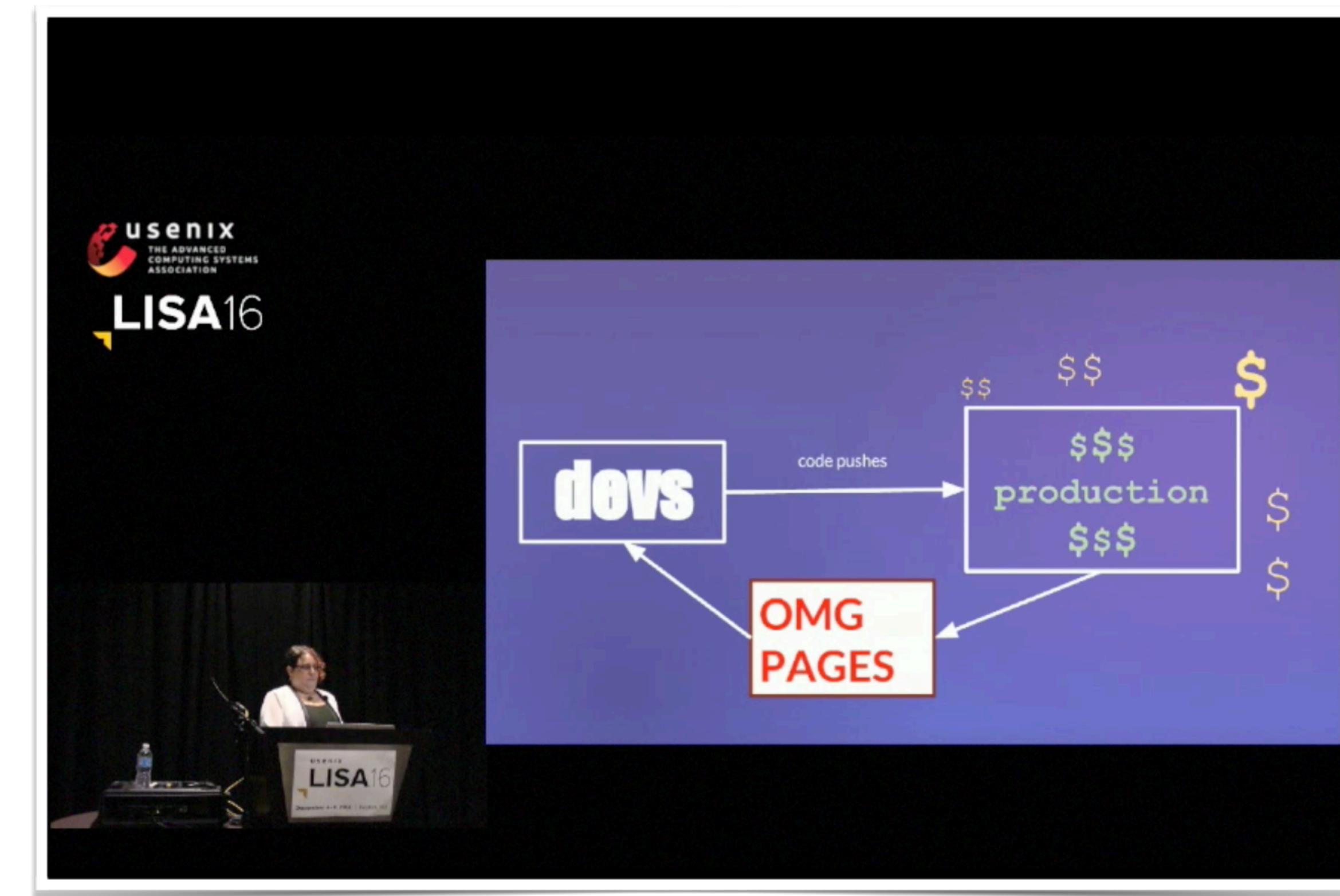


<https://manning.com/books/mesos-in-action>



WHERE CAN I LEARN MORE?

usenix.org/conference/lisa16/conference-program/presentation/eckhardt



Courtney Eckhardt, Heroku

WHERE CAN I LEARN MORE?

some.ops4devs.info

The screenshot shows the homepage of the 'Operations For Developers' website. At the top, there is a navigation bar with the site's name 'Operations For Developers' on the left and links for 'CONTAINERS', 'STORAGE', 'MONITORING', 'LOGGING', 'DEPLOYMENTS', 'NETWORKING', 'SECURITY', 'BACKUPS', and 'FUNCTION-AS-A-SERVICE' on the right. The main title 'Operations For Developers' is prominently displayed in a large, bold, dark font, underlined. Below the title is a subtitle 'you wrote it, we all run it!'. A featured article titled 'Welcome!' is shown, posted on Dec 09, 2016. The text of the article discusses the challenges of learning modern operations skills for developers and reassures them that the website provides resources to help them strengthen their ops skills.

Operations For Developers

CONTAINERS STORAGE MONITORING LOGGING DEPLOYMENTS NETWORKING SECURITY BACKUPS FUNCTION-AS-A-SERVICE

Operations For Developers

you wrote it, we all run it!

Welcome!

Posted on Dec 09, 2016

Hello my dear developer. I'm sure with all that cloud native and container stuff these days you've been asking yourself: What do I need to learn to deploy and run 'modern' services? Which operations—or ops for short—skills are required of me? Do not despair, we got you covered. This website is dedicated to collecting resources for developers to strengthen their ops skills. You might have noticed, it's really [\[Read More\]](#)

Q & A

- [@mhausenblas](#)
- [mhausenblas.info](#)



<https://dcos.io>