
@unterstein @dcos @jaxcon

Container orchestrierung on Mesos DC/OS for Java developers



DC/OS

Micro-
Services

Docker

Scheduling

Mesos

Hadoop

Spark

DC/OS

Container Orchestration



Johannes Unterstein

Distributed Applications Engineer

 @unterstein



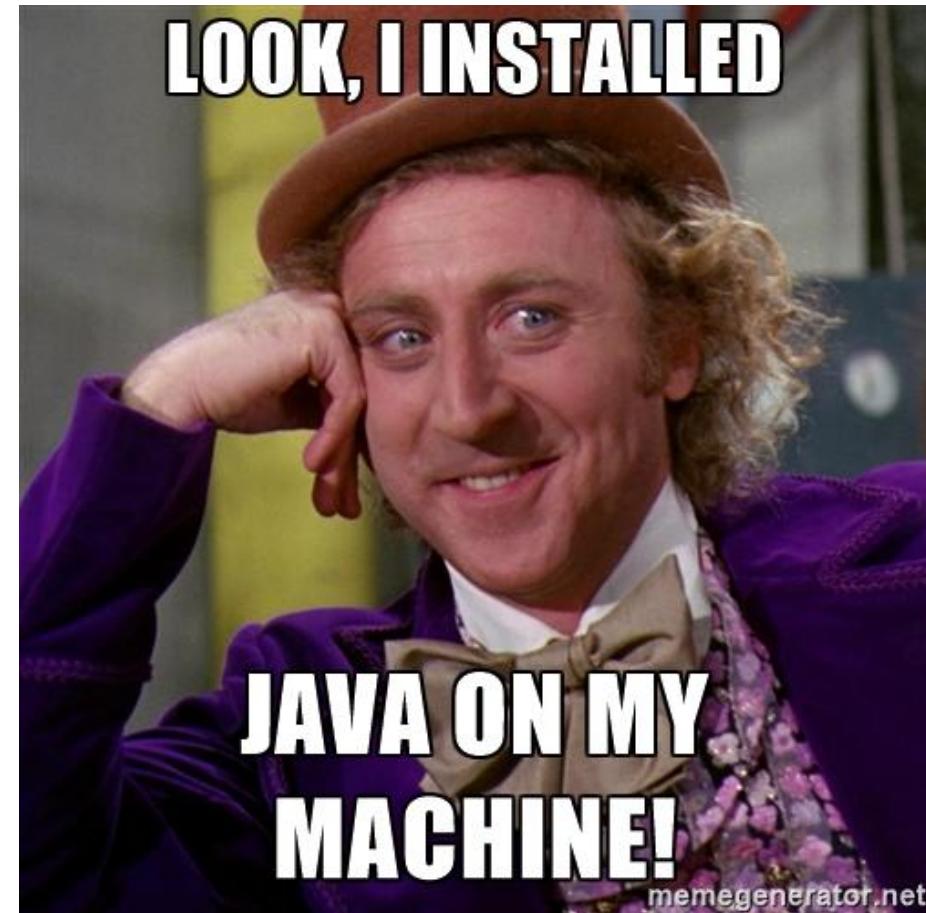
The story

The story

Java developer Alice builds new `App with Chuck Norris jokes` and put it into production in three easy steps.

<https://git.io/vXUoy>

Step 1 Runs on my machine



Step 2
Runs on my
container
engine

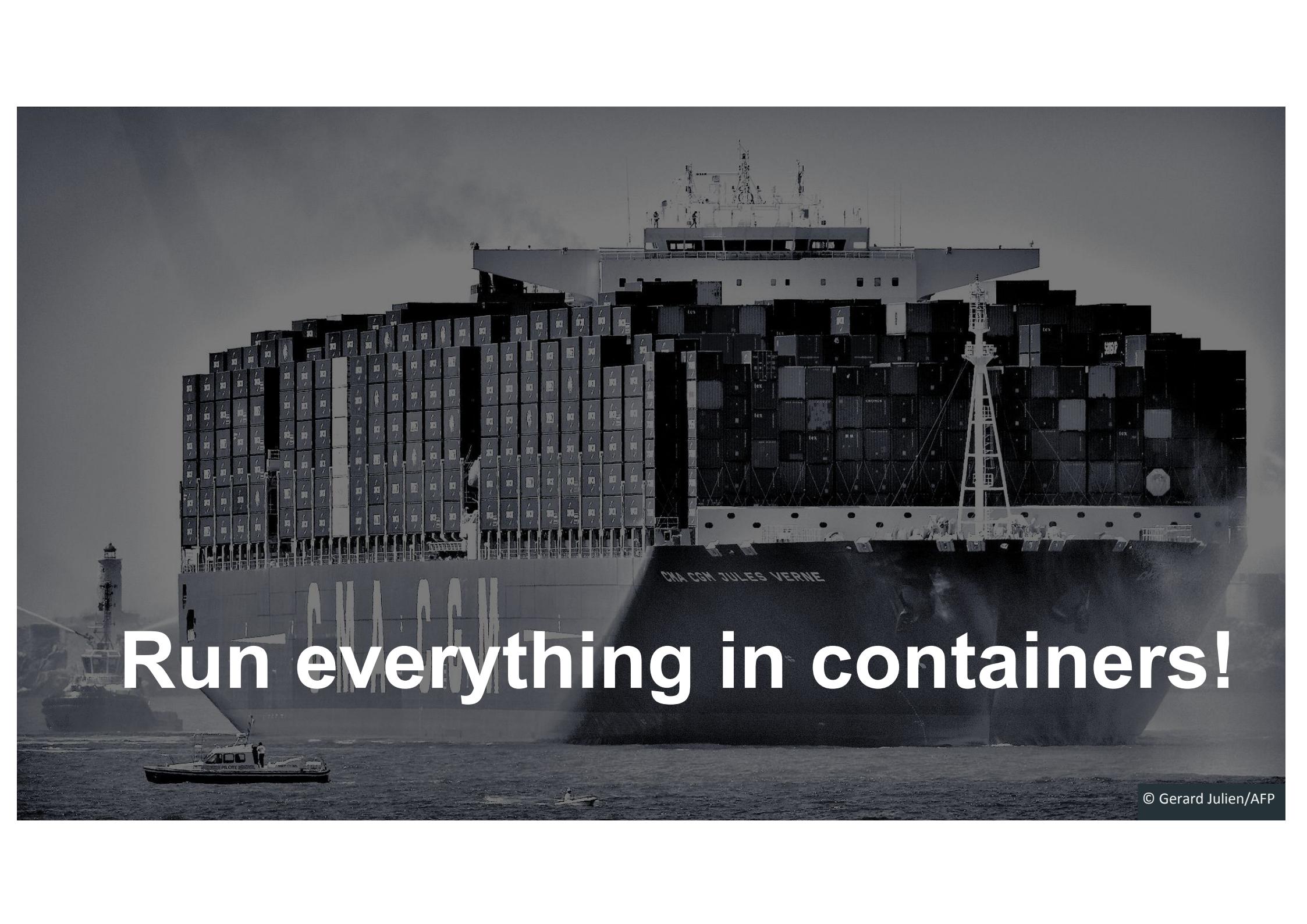


Step 3 Runs on my datacenter



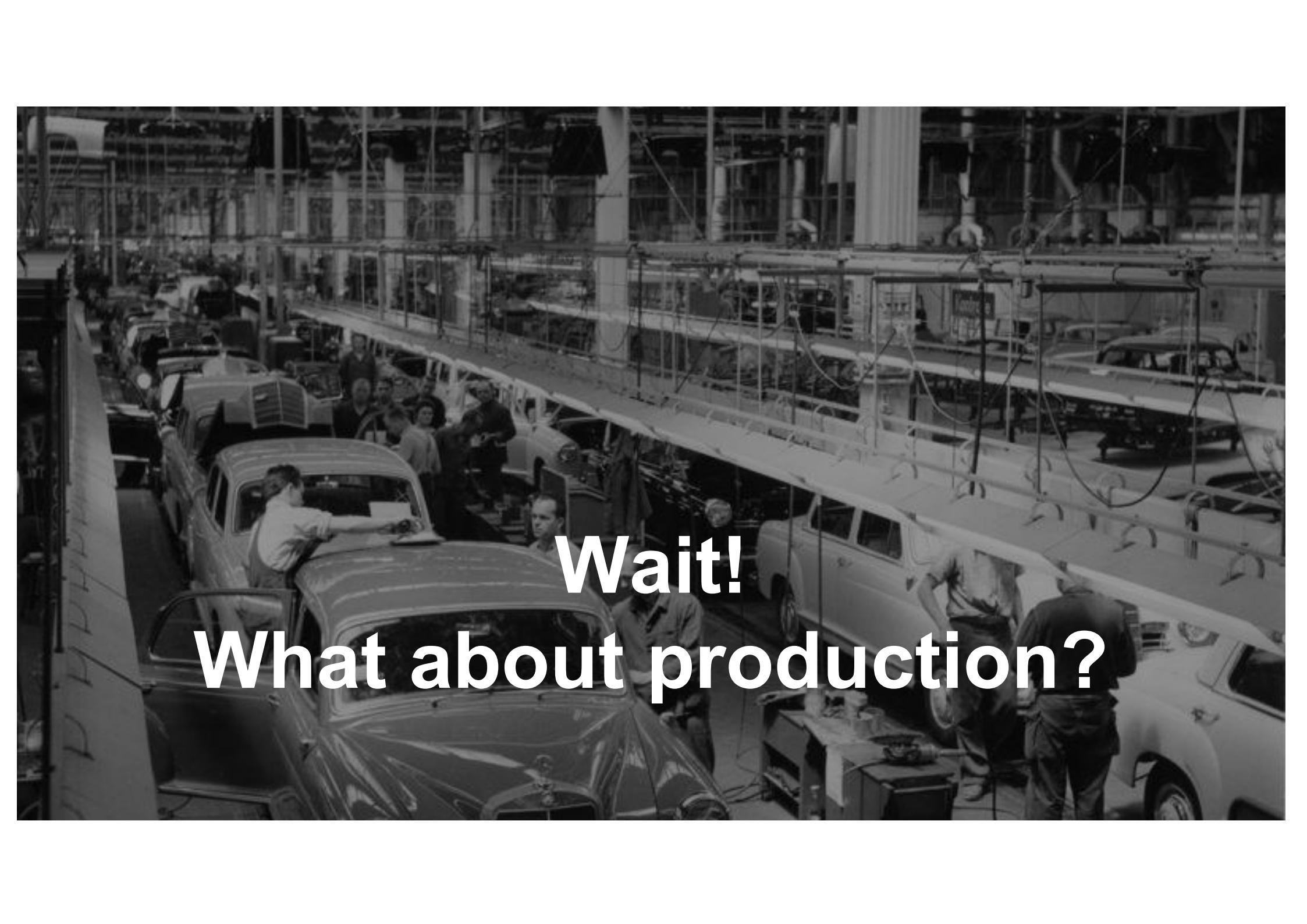
A black and white photograph of a man in a dark suit and glasses standing between two large, metallic reel-to-reel tape deck machines. He is holding a single reel of tape in his hands, looking down at it. The machines have multiple reels and control panels with numerous buttons and knobs. The background consists of vertical metal panels.

Works on my machine!



Run everything in containers!

© Gerard Julien/AFP



Wait!
What about production?

What about container scheduling?



What about resource management?



What about service management?



Orchestration

CONTAINER SCHEDULING



RESOURCE MANAGEMENT



SERVICE MANAGEMENT



Orchestration

CONTAINER SCHEDULING

- Placement
- Replication/Scaling
- Resurrection
- Rescheduling
- Rolling Deployment
- Upgrades
- Downgrades
- Collocation

RESOURCE MANAGEMENT

- Memory
- CPU
- GPU
- Volumes
- Ports
- IPs
- Images/Artifacts

SERVICE MANAGEMENT

- Labels
- Groups/Namespaces
- Dependencies
- Load Balancing
- Readiness Checking

Modern datacenter





Hadoop

Database

Services

Spark

memcached



Hadoop

Database

Services

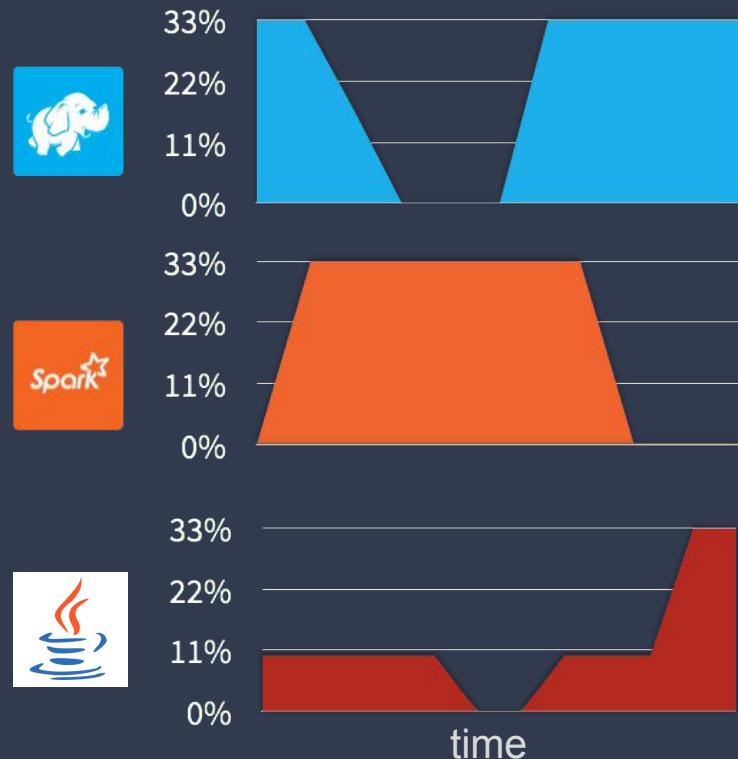
Spark

memcached

KEEP IT STATIC

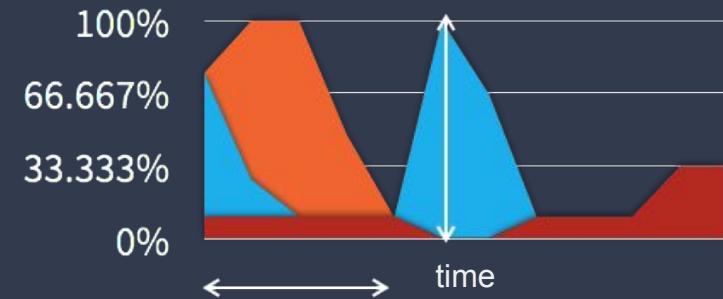
A naive approach to handling varied app requirements: **static partitioning**.

Maintaining sufficient headroom to handle peak workloads on all partitions leads to **poor utilization** overall.



SHARED RESOURCES

Multiple frameworks can use the same cluster resources, with their share adjusting dynamically.





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.

April 2016

DC/OS

TECHNOLOGY

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

Sharing resources between batch processing frameworks

- Hadoop
- MPI
- Spark

VISION

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

What does an operating system provide?

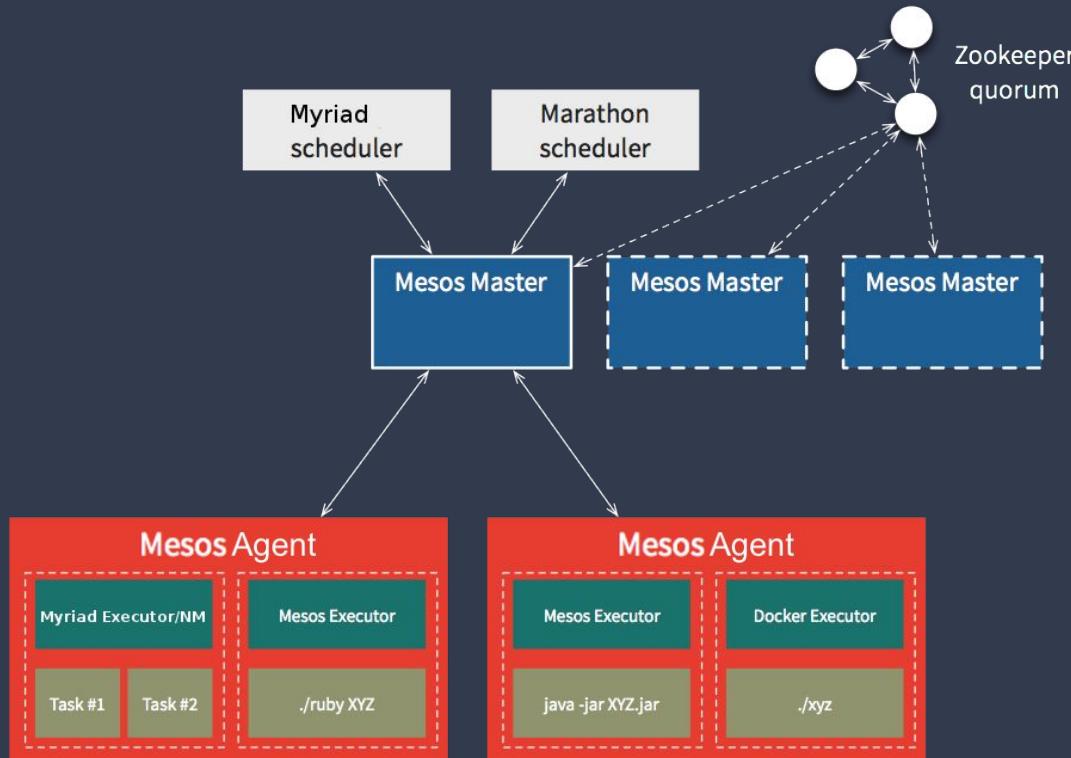
- Resource management
- Programming abstractions
- Security
- Monitoring, debugging, logging

Apache Mesos

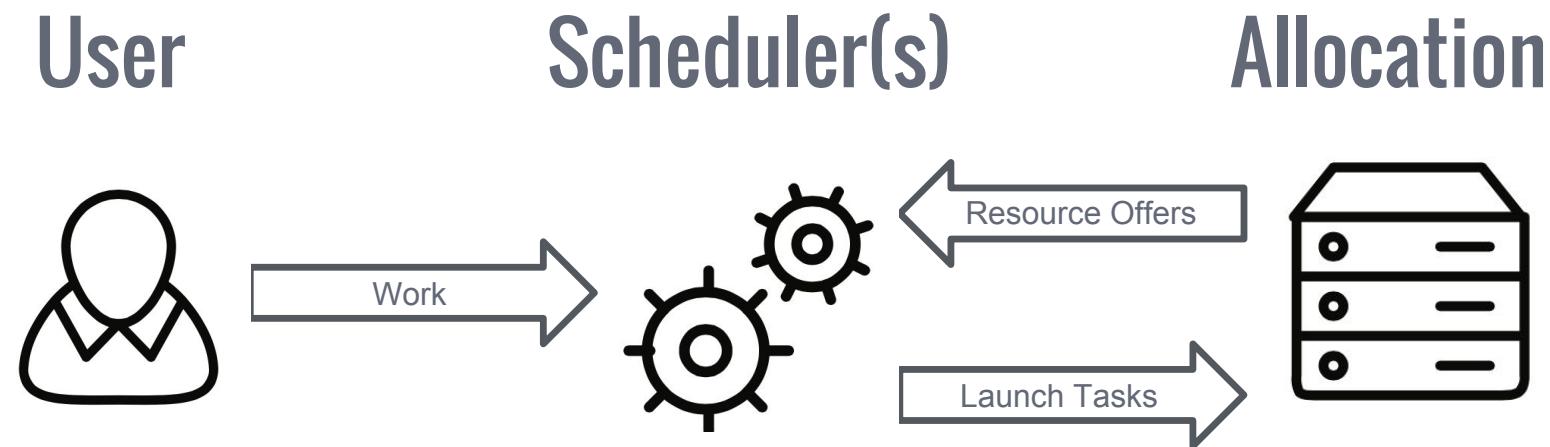
- A top-level Apache project
- A cluster resource negotiator
- Scalable to 10,000s of nodes
- Fault-tolerant, battle-tested
- An SDK for distributed apps
- Native Docker support



ARCHITECTURE

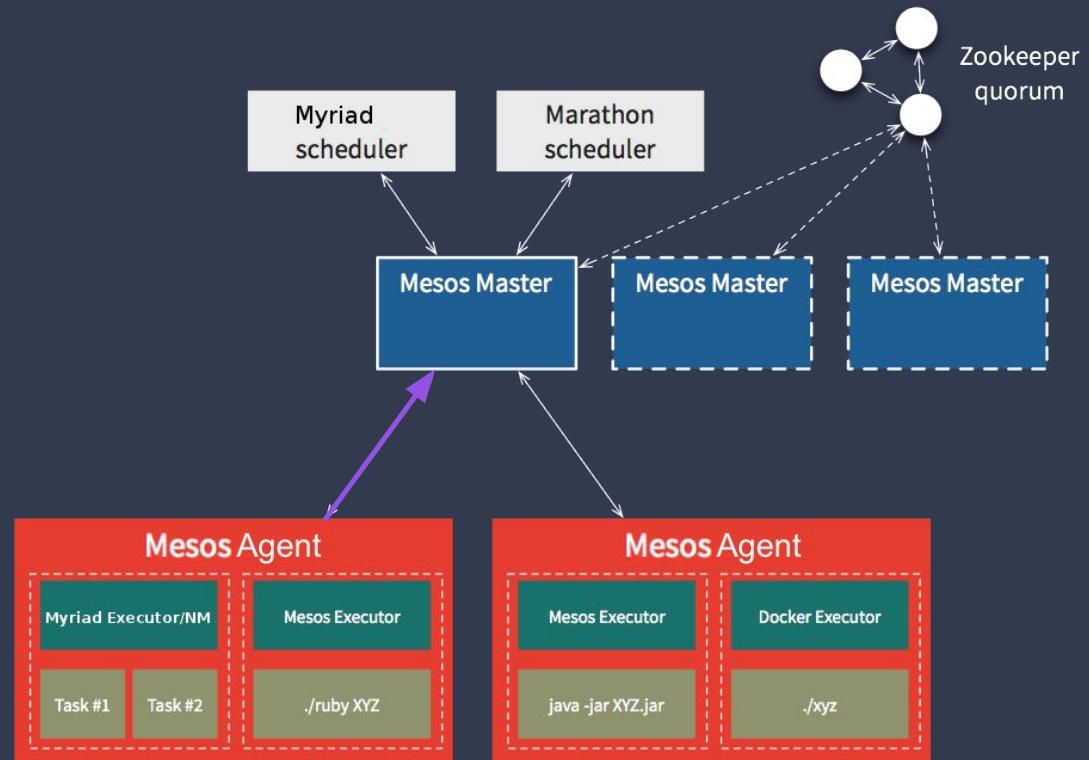


2-Level Scheduling



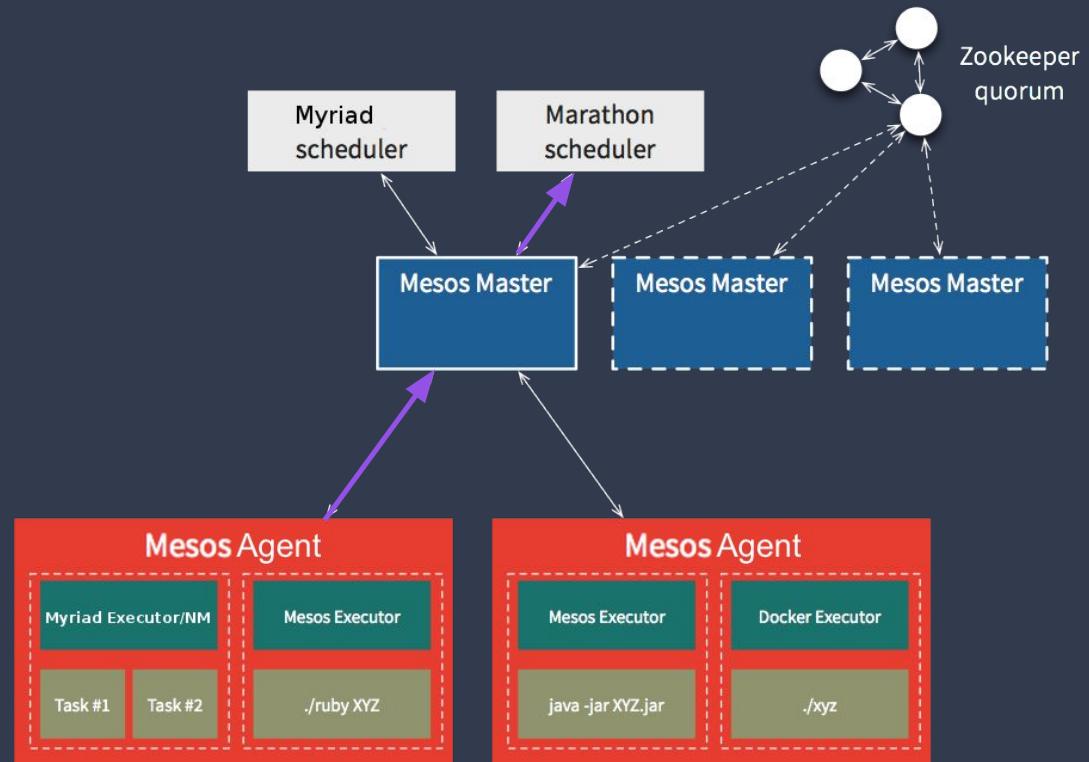
ARCHITECTURE

- Agents advertise resources to Master
- Master offers resources to Framework
- Framework rejects/uses resources
- Agents report task status to Master



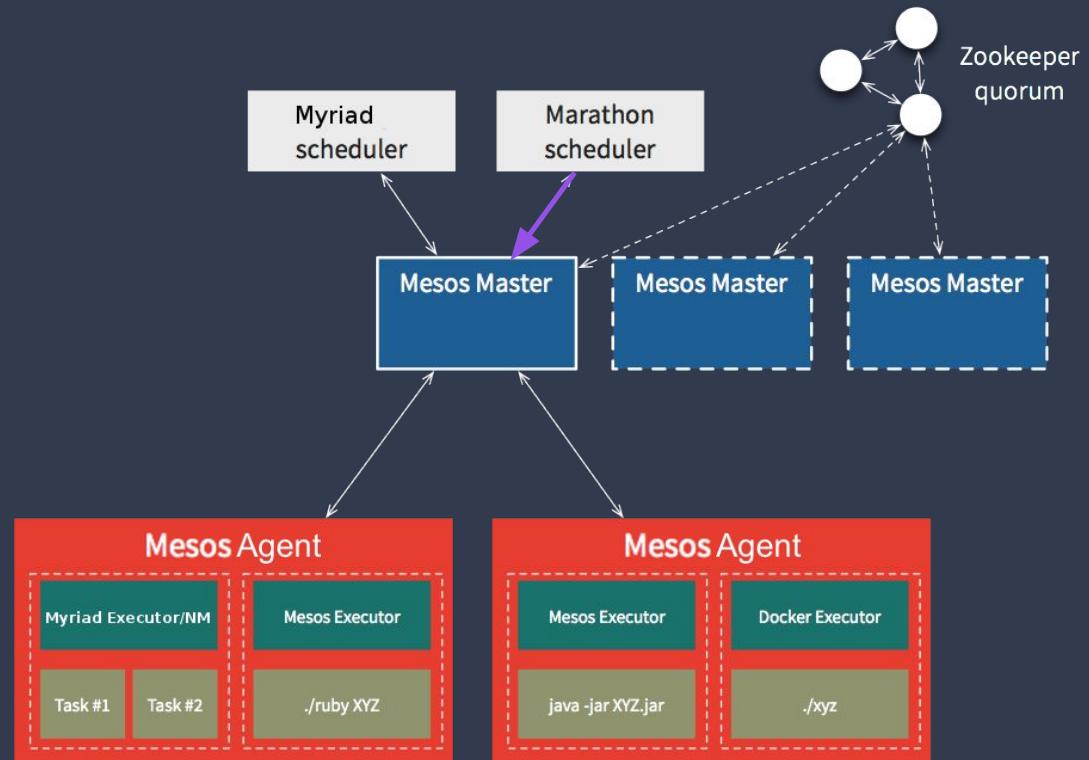
ARCHITECTURE

- Agents advertise resources to Master
- Master offers resources to Framework
- Framework rejects/uses resources
- Agents report task status to Master



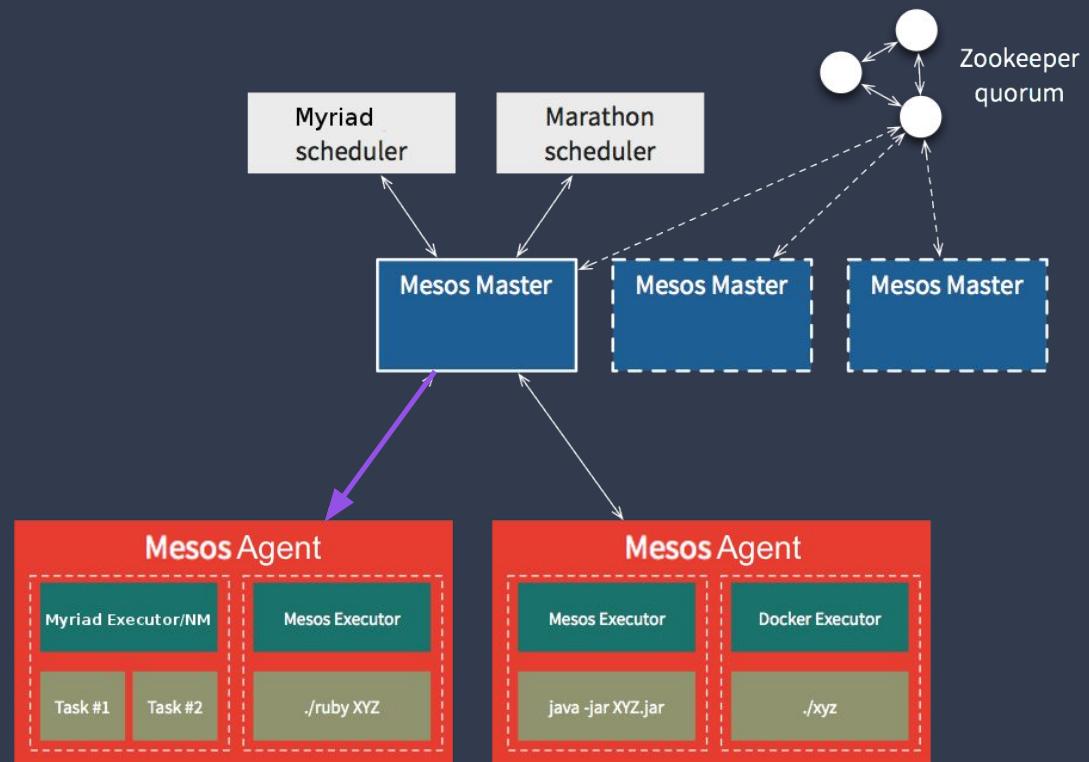
ARCHITECTURE

- Agents advertise resources to Master
- Master offers resources to Framework
- Framework rejects/uses resources
- Agents report task status to Master



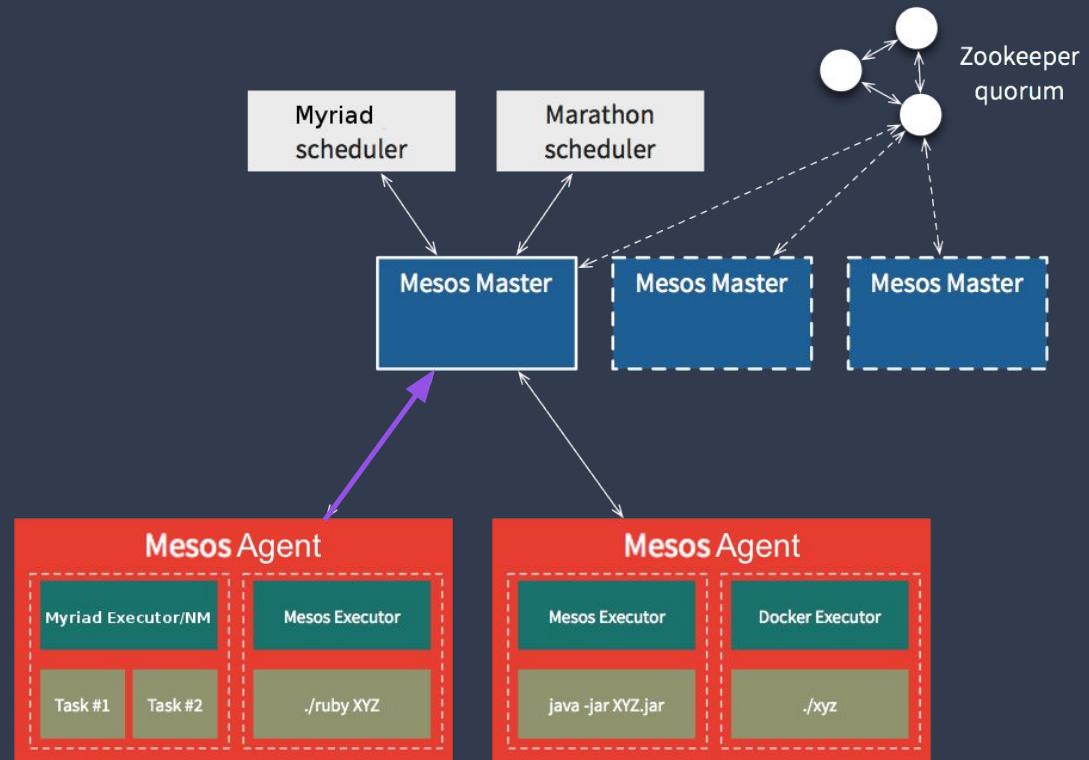
ARCHITECTURE

- Agents advertise resources to Master
- Master offers resources to Framework
- Framework rejects/uses resources
- Agents report task status to Master

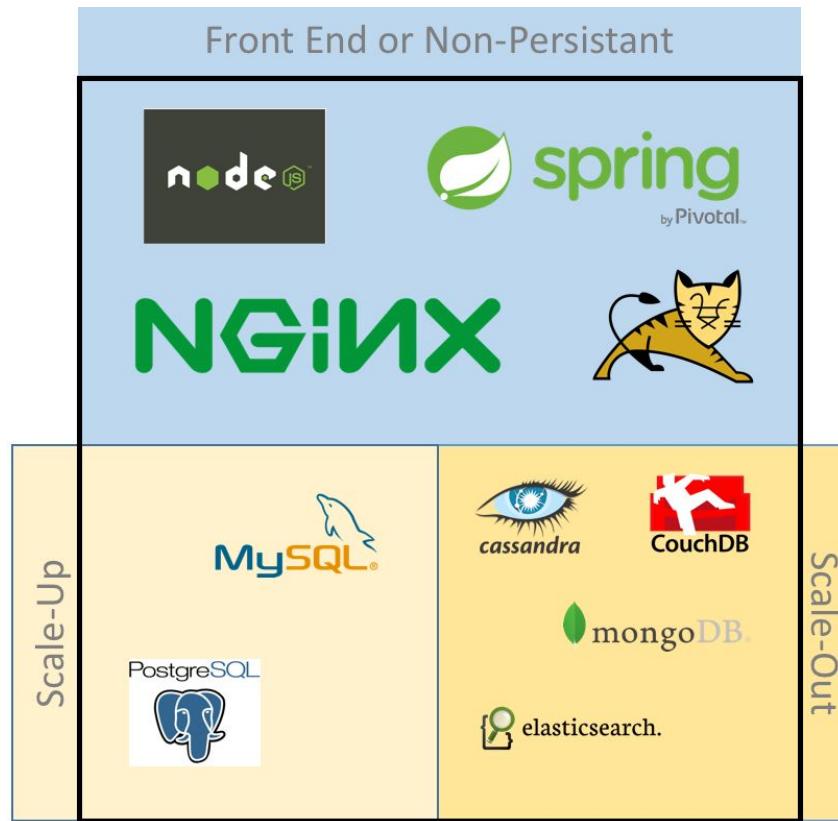


ARCHITECTURE

- Agents advertise resources to Master
- Master offers resources to Framework
- Framework rejects/uses resources
- Agents report task status to Master

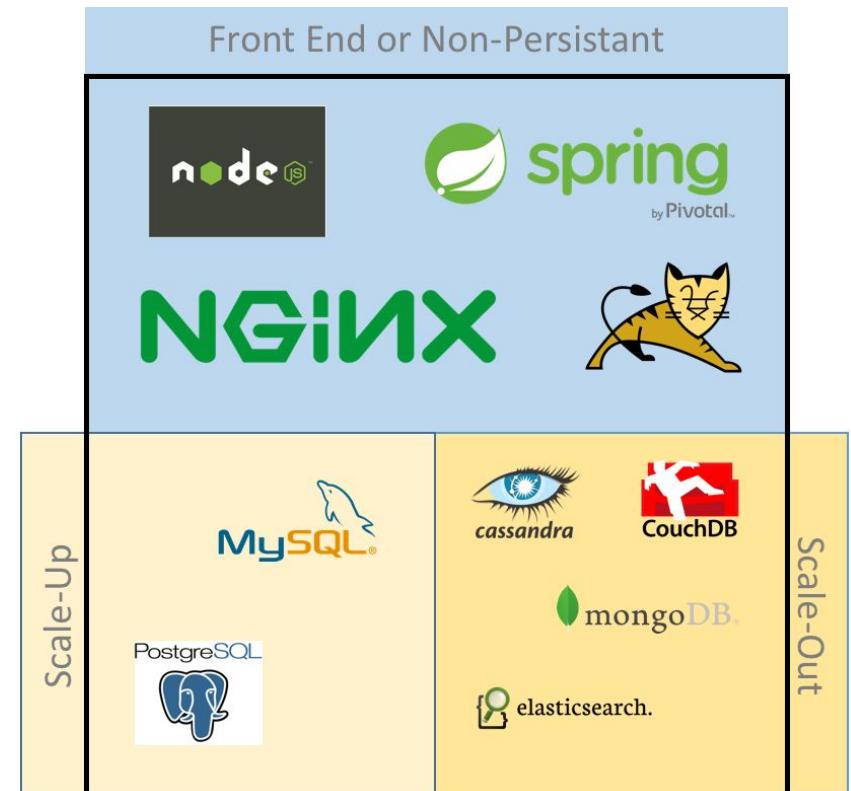


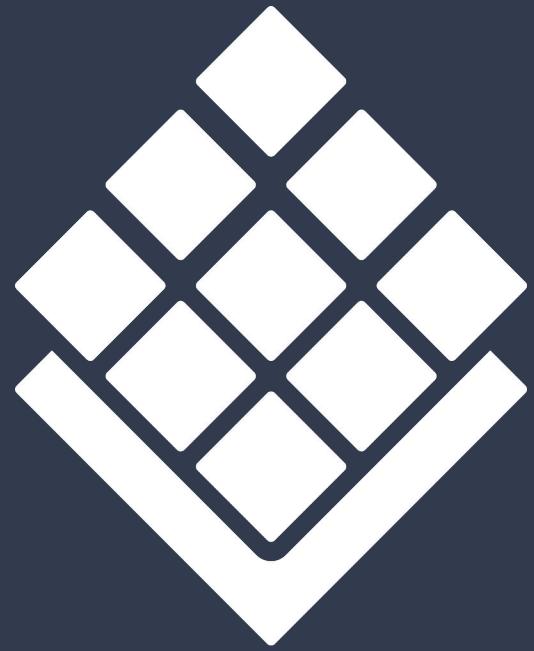
Persistence



Storage Options

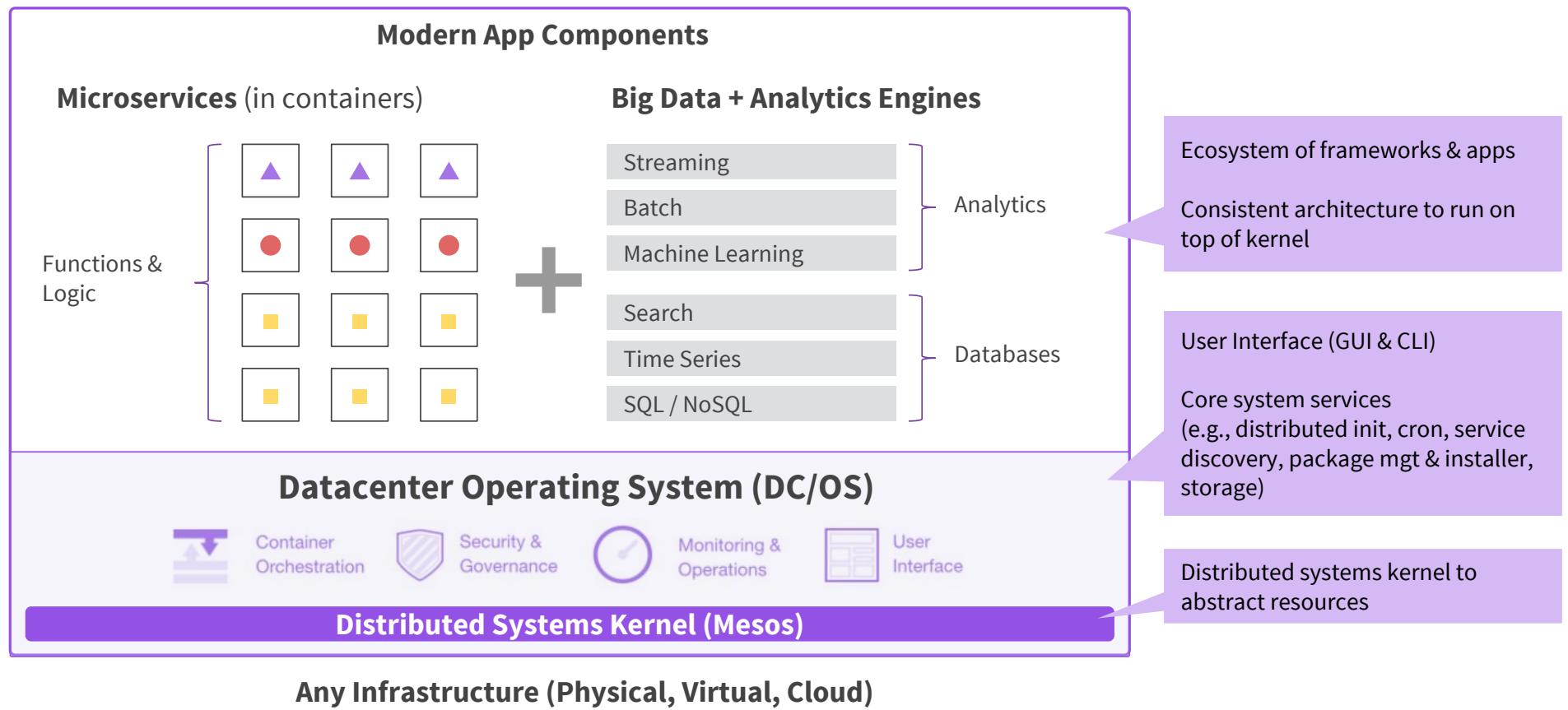
- Default Sandbox
 - ➡ Simple to use, Task failures
- Persistent Volumes
 - ➡ Task failures, (permanent) Node failures
- Distributed File System/External Storage
 - ➡ Node failures, non-local writes





DC/OS

DC/OS ENABLES MODERN DISTRIBUTED APPS



THE BASICS



DC/OS is ...

- 100% open source (ASL2.0)
 - + A big, diverse community
- An umbrella for ~30 OSS projects
 - + Roadmap and designs
 - + The build tool chain
 - + Docs and tutorials
- Not limited in any way
- Familiar, with a few new features

sf42-prod

52.36.125.131



Dashboard



Services



Nodes



Universe



System



DC/OS v.1.7-dev



THE UNIVERSE

Packages Installed

Community Packages

-  crate 0.1.0
-  datadog 5.4.3
-  elasticsearch 0.7.0
-  etcd 0.0.2
-  exhibitor 0.8.1
-  hdfs 0.1.8
-  hue 0.0.1
-  kubernetes v0.7.2-v1.1.5-alpha
-  marathon-lb 0.0.5-0.1
-  memsql 0.0.1
-  mr-redis 0.0.1
-  openvpn

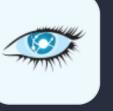
Packages Installed

Selected Packages



arangodb
0.3.0

[Install Package](#)



cassandra
0.2.0-2

[Install Package](#)



chronos
2.4.0

[Install Package](#)



jenkins
0.2.3

[Install Package](#)



kafka
0.9.4.0

[Install Package](#)



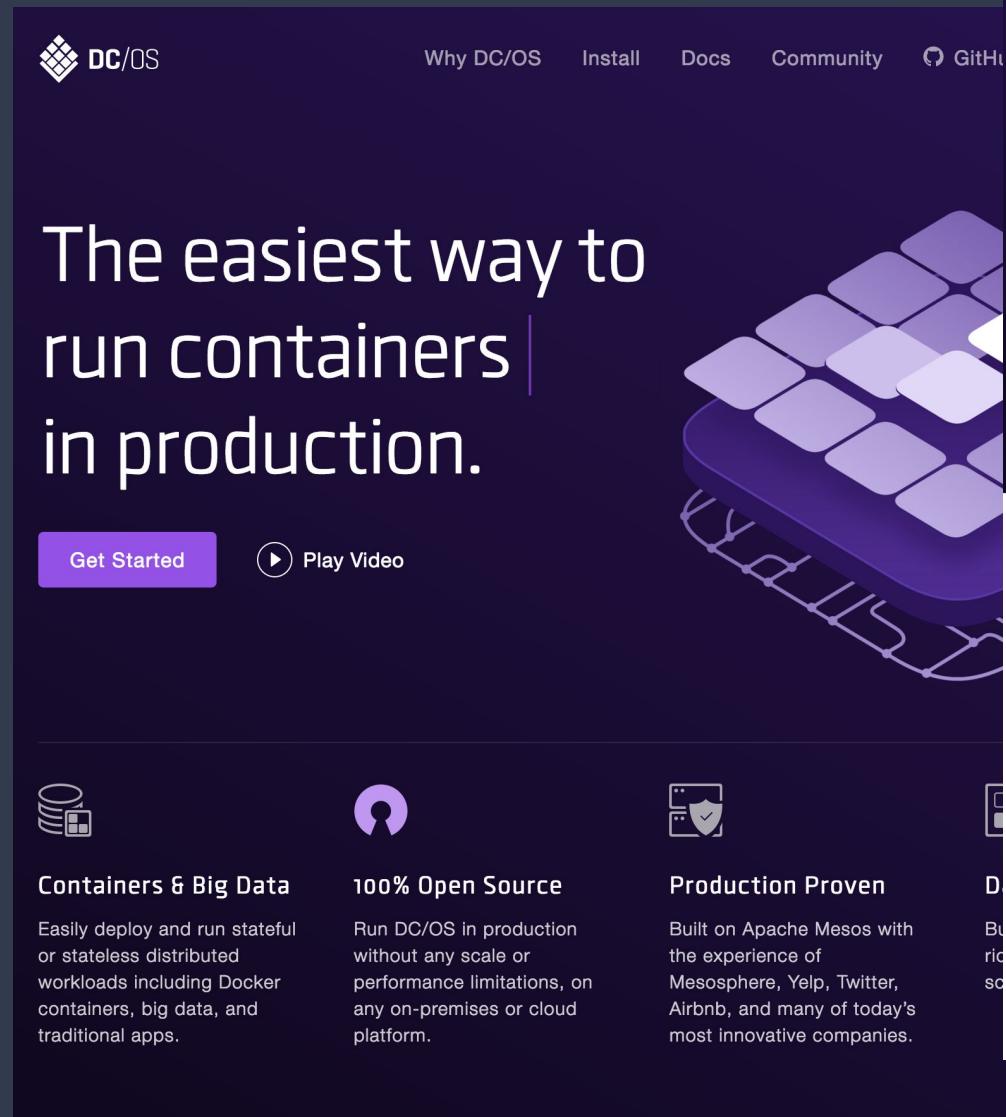
marathon
0.15.3

[Install Package](#)



spark
1.6.0

[Install Package](#)



The easiest way to run containers in production.

Get Started

Play Video

Containers & Big Data

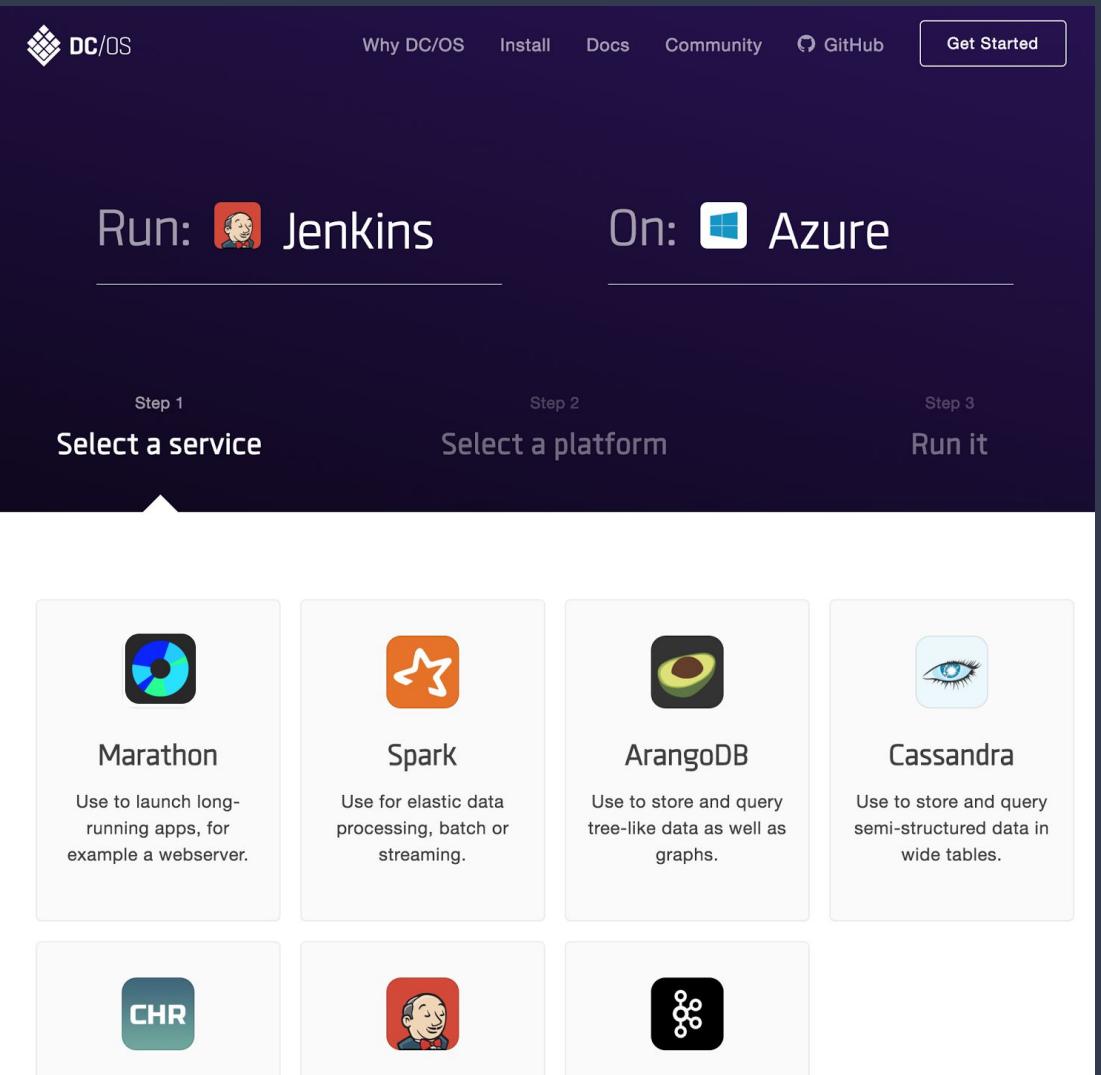
Easily deploy and run stateful or stateless distributed workloads including Docker containers, big data, and traditional apps.

100% Open Source

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

Production Proven

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



DC/OS

Why DC/OS Install Docs Community GitHub Get Started

Run: Jenkins On: Azure

Step 1 Select a service Step 2 Select a platform Step 3 Run it

 Marathon Use to launch long-running apps, for example a webserver.	 Spark Use for elastic data processing, batch or streaming.	 ArangoDB Use to store and query tree-like data as well as graphs.	 Cassandra Use to store and query semi-structured data in wide tables.
 CHR	 Jenkins	 Azure	

~~Micro-
Services~~

~~Docker~~

~~Scheduling~~

~~Mesos~~

~~Hadoop~~

~~Spark~~

~~DC/OS~~

~~Container Orchestration~~



Questions?

<https://git.io/vXUoy>



DC/OS

© 2016 Mesosphere, Inc. All Rights Reserved.