
APACHE MESOS NYC MEETUP | SEPTEMBER 22, 2016

CONTINUOUS DELIVERY WITH MESOS, DC/OS AND JENKINS



MESOSPHERE

WHO WE ARE



ROGER IGNAZIO

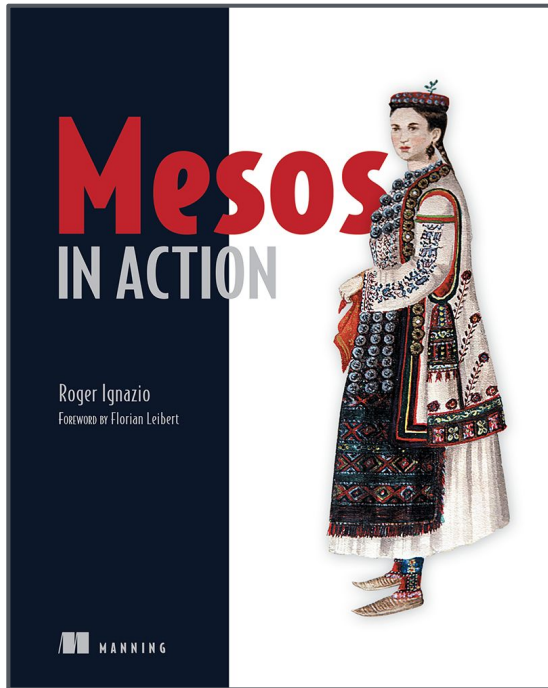
Tech Lead at Mesosphere
@rogerignazio



SUNIL SHAH

Product Manager at Mesosphere
@ssk2

BUY MY BOOK!



mesosinaction.com

Use the code **vecymes** for
42% off!

AGENDA

Background

- Introduction to Apache Mesos and DC/OS
- Components that make up modern infra.
- Running Jenkins as a service on DC/OS

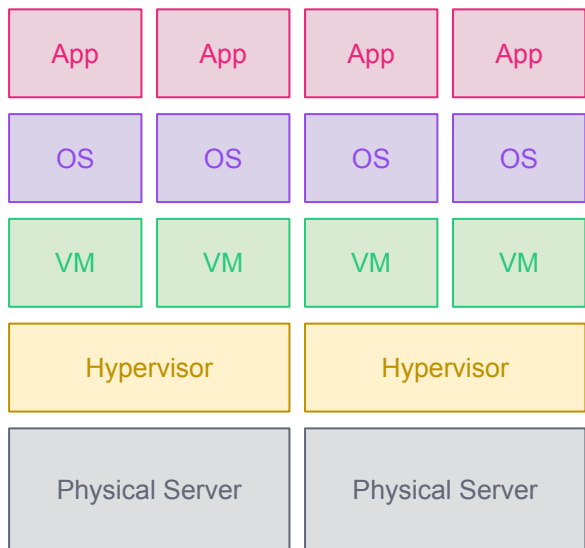
Demo

- Installing and configuring Jenkins
- Installing and configuring a load balancer
- Creating a new CI/CD pipeline
- Continuously deploying applications to DC/OS

INTRO TO APACHE MESOS AND DC/OS

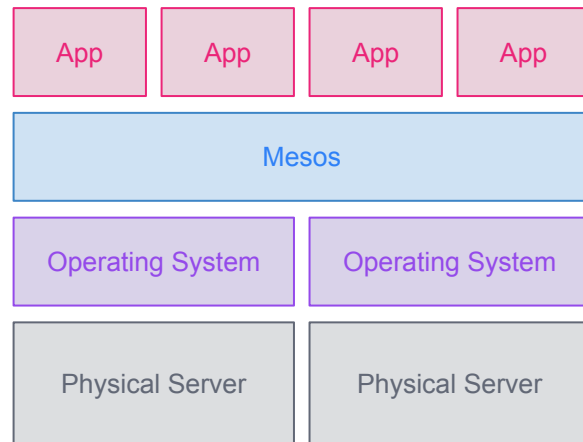
A QUICK PRIMER ON CONTAINERS

Virtual Machine–Based Application Deployment



Isolate apps by running multiple VMs per physical server; still need to manage each guest OS!

Container–Based Application Deployment



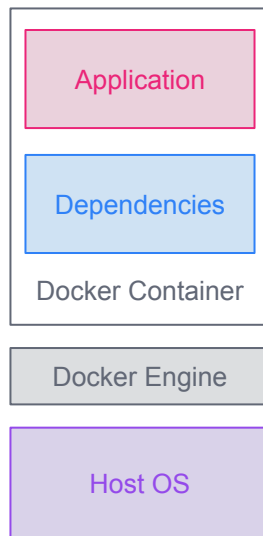
Isolate apps using features of the host OS, such as Linux cgroups.

A QUICK PRIMER ON CONTAINERS

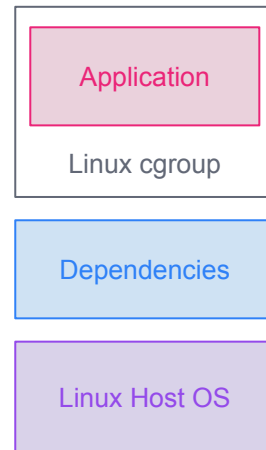
Virtual Machines



Docker Containers



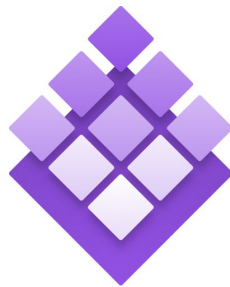
Linux cgroups



A BIT OF CLARIFICATION



Apache
MESOS™



DC/OS

<https://mesos.apache.org>

<https://dcos.io>

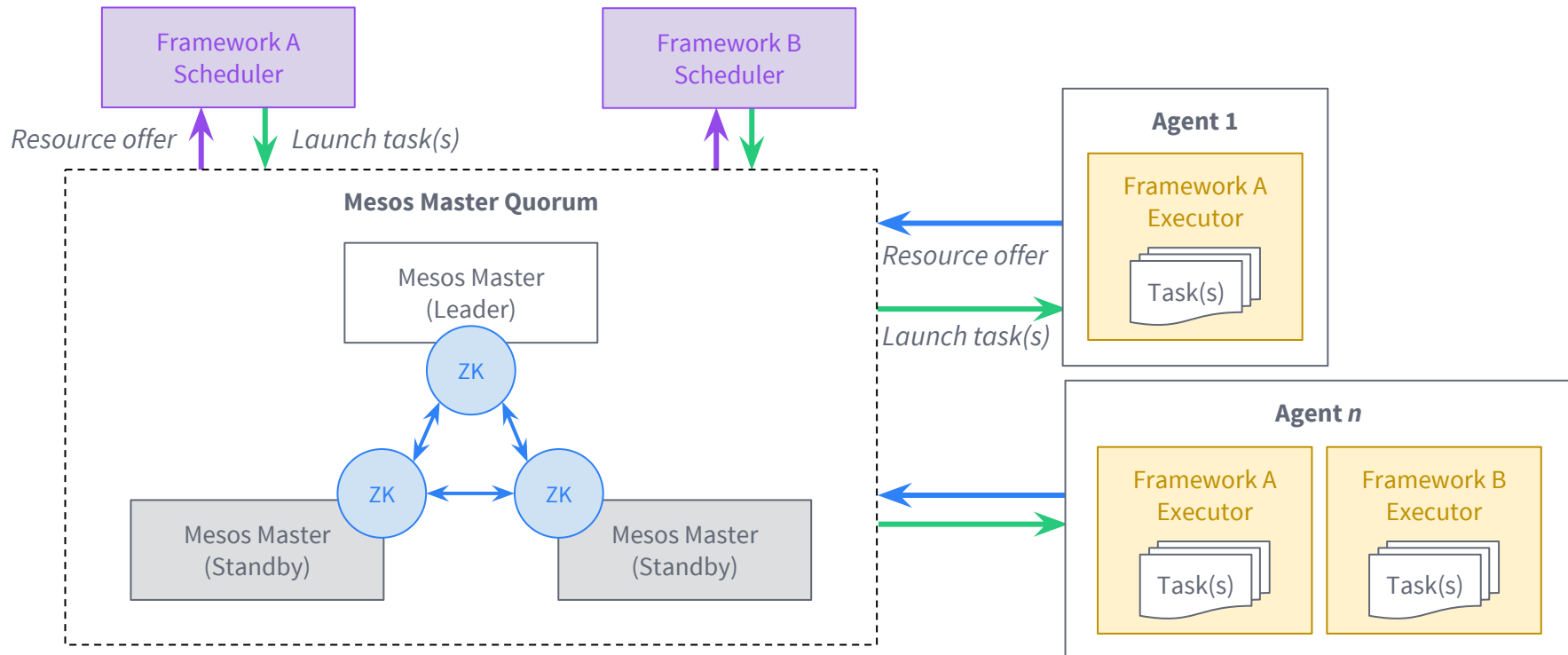
WHAT IS MESOS?

- General purpose cluster resource manager
- Represents many machines as a single entity
- Advertises resources directly to *frameworks*
- Works at scale: Apple, Twitter, Airbnb, Netflix, ...

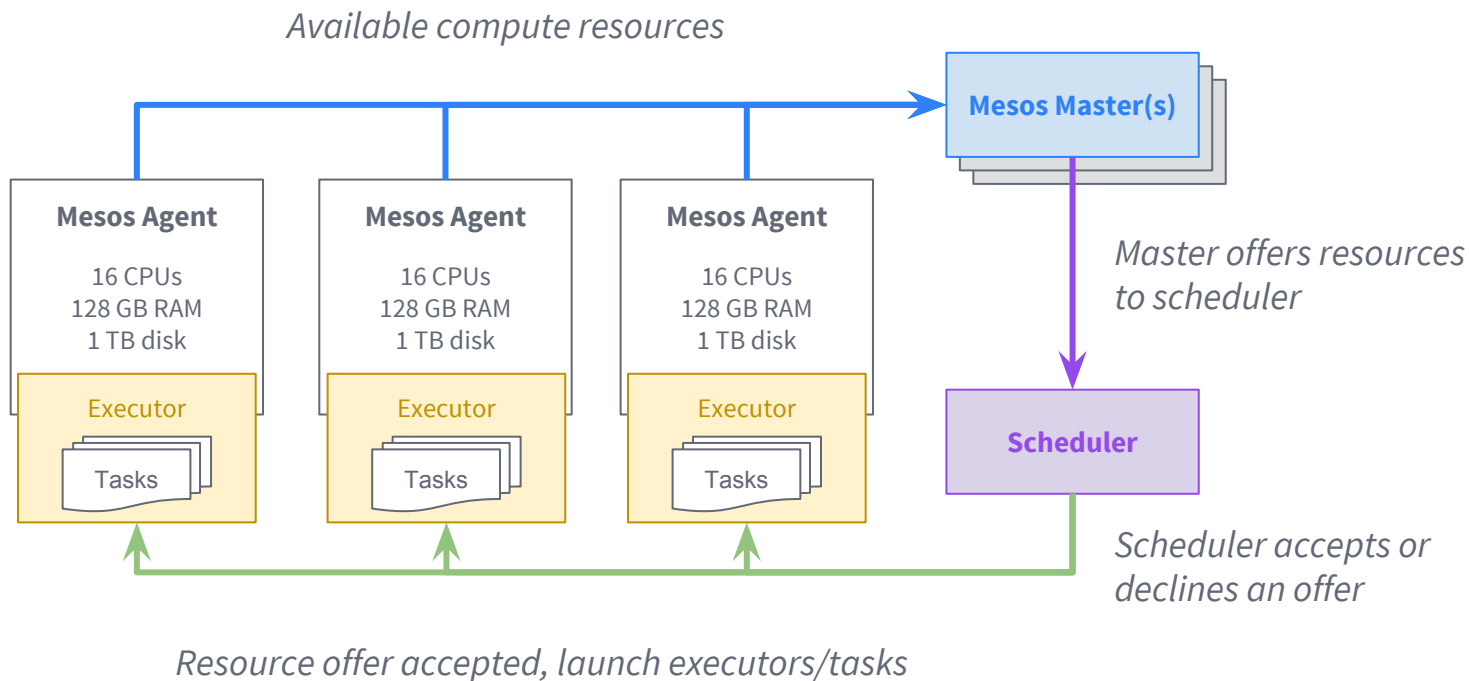
WHAT IS MESOS? (CONTINUED)

- Two-tier scheduling across resource types
 - cpus, mem, disk, and ports by default
- Masters are highly available, agents are fault tolerant
 - *Checkpointing, agent recovery*
- Resource isolation between processes
 - Linux cgroups, Docker, ...
- Language bindings: C++, Java, Python, Go, ...

MESOS ARCHITECTURE



ANATOMY OF A RESOURCE OFFER (TWO-TIER SCHEDULING)



NEW (OLD) PROBLEMS

- Service discovery and load balancing
 - BIND, Mesos-DNS, Consul-Mesos, Marathon-LB

NEW (OLD) PROBLEMS

- Service discovery and load balancing
 - BIND, Mesos-DNS, Consul-Mesos, Marathon-LB
- Monitoring and metrics collection
 - Collectd, Nagios, Prometheus, Snap

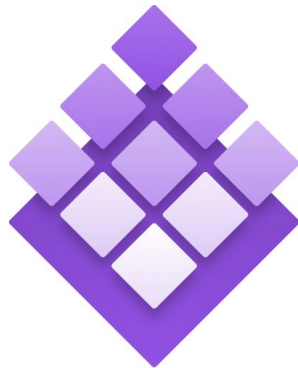
NEW (OLD) PROBLEMS

- Service discovery and load balancing
 - BIND, Mesos-DNS, Consul-Mesos, Marathon-LB
- Monitoring and metrics collection
 - Collectd, Nagios, Prometheus, Snap
- Persistent storage (filesystems, databases, etc)
 - Ceph, HDFS, Amazon EBS / EFS / S3, NFS, Cassandra

NEW (OLD) PROBLEMS

- Service discovery and load balancing
 - BIND, Mesos-DNS, Consul-Mesos, Marathon-LB
- Monitoring and metrics collection
 - Collectd, Nagios, Prometheus, Snap
- Persistent storage (filesystems, databases, etc)
 - Ceph, HDFS, Amazon EBS / EFS / S3, NFS, Cassandra
- Administration: named URIs vs. ports, IPAM
 - Nginx, HAProxy, Mesos-DNS, dnsmasq, Minuteman

DC/OS: BUILT ON MESOS



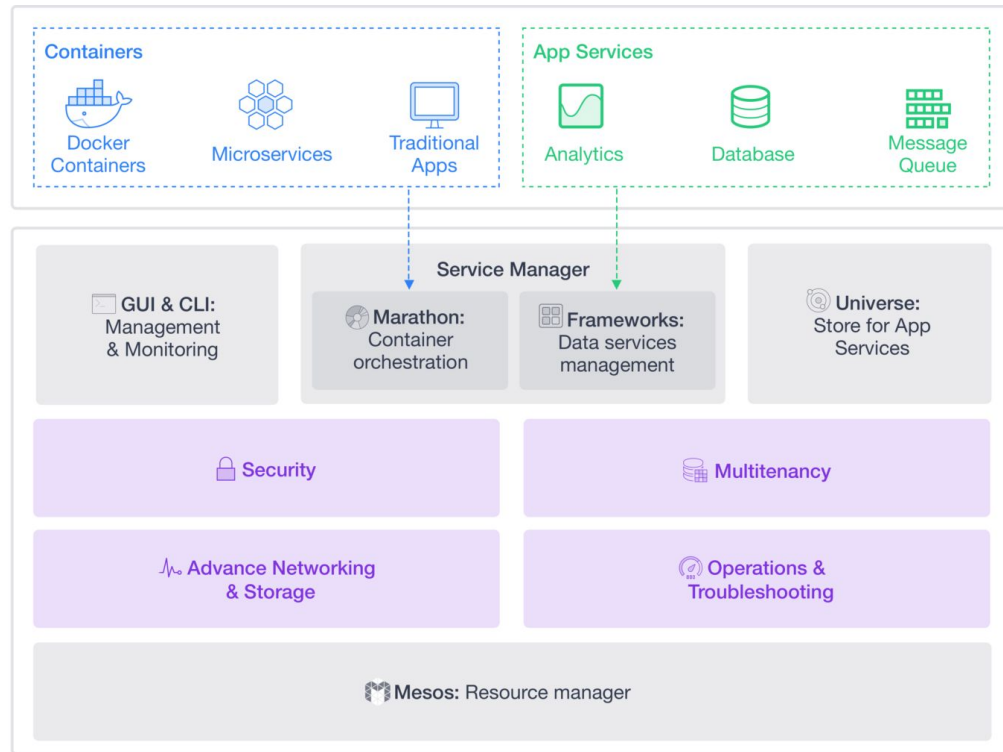
DC/OS

<https://dcos.io>

<https://github.com/dcos>

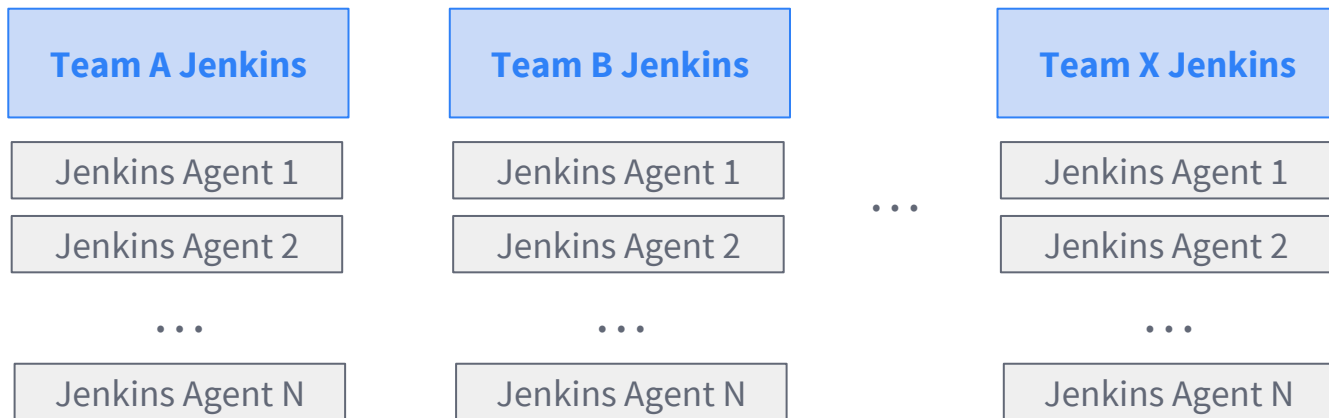
DC/OS: BUILT ON MESOS

MODERN APPS



JENKINS ON DC/OS

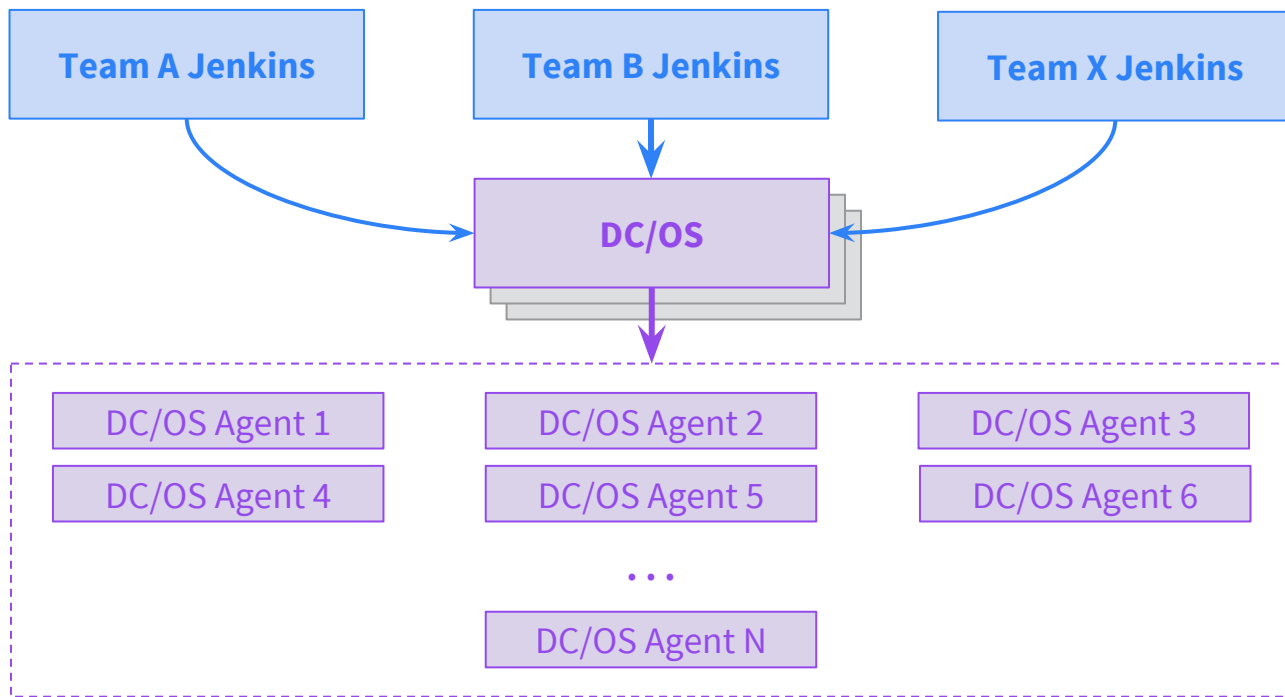
MULTIPLE JENKINS MASTERS



MULTIPLE JENKINS MASTERS → STATIC PARTITIONING



RUNNING JENKINS ON DC/OS → FAIR SHARING



CONTINUOUSLY DEPLOYING APPLICATIONS TO DC/OS

DEPLOYING APPLICATIONS: BASIC REQUIREMENTS

- ***Scheduling*** — advertising available compute resources
- ***Deployments*** — getting an application onto a node
- ***Health checks*** — ensuring the app/service is healthy
- ***Service discovery*** — connecting to dependent services
- ***Persistence*** — running stateful services in containers

DEPLOYING APPLICATIONS: SCHEDULING

Before DC/OS

A sysadmin provisions one or more physical/virtual servers to host the app

With DC/OS

Mesos resource offers (two-tier scheduling) offers available resources directly to frameworks

DEPLOYING APPLICATIONS: DEPLOYMENTS

Before DC/OS

By hand or using Puppet / Chef / Ansible

Jenkins SSHing to the machine and running a shell script

Note: all dependencies must also be present!

With DC/OS

Marathon deploys containers, ideally using a CI/CD tool to create/update app definitions

Docker containers packages app and dependencies

DEPLOYING APPLICATIONS: HEALTH CHECKS

Before DC/OS

Nagios pages a sysadmin

With DC/OS

Marathon performs health checks,
restarts unhealthy/failed instances

DEPLOYING APPLICATIONS: SERVICE DISCOVERY

Before DC/OS

Static hostnames / IP addresses in a spreadsheet or config management

A sysadmin configures a load balancer manually or with Puppet / Chef / Ansible

With DC/OS

Mesos-DNS provides DNS resolution for running services (hostname / IP address, ports, etc)

Load balancer configs built dynamically using cluster state

DEPLOYING APPLICATIONS: PERSISTENCE

Before DC/OS

Individual servers with RAID 1/5/6/10, expensive SANs, NFS, etc.

Dedicated, statically partitioned Ceph or Gluster storage clusters

With DC/OS

Mesos external/persistent volumes (REX-Ray), HDFS, etc.

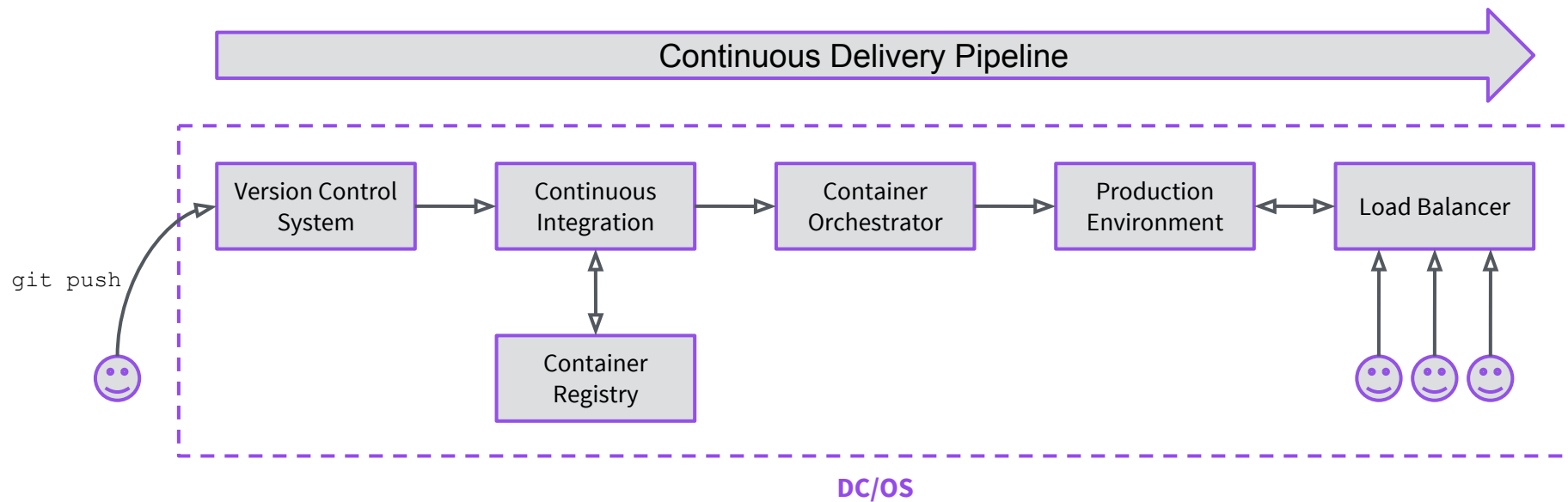
Self-healing Ceph or Gluster on Mesos / DC/OS

Q: Given that we are all engineers,
which tools would we like to work with?

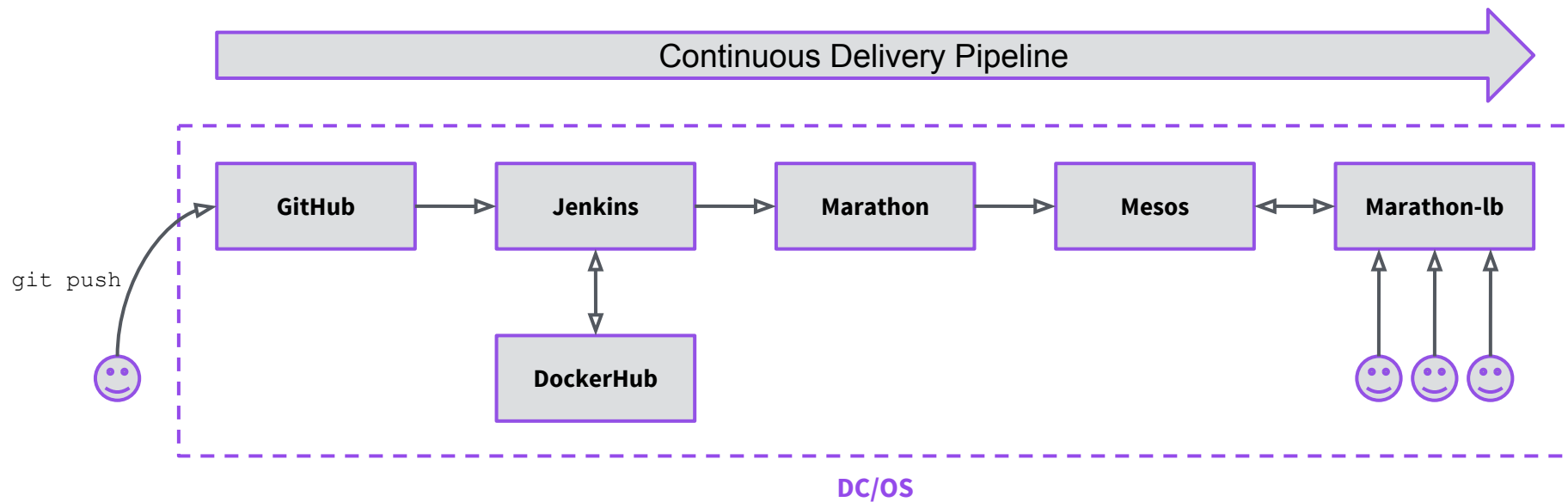


DEMOS

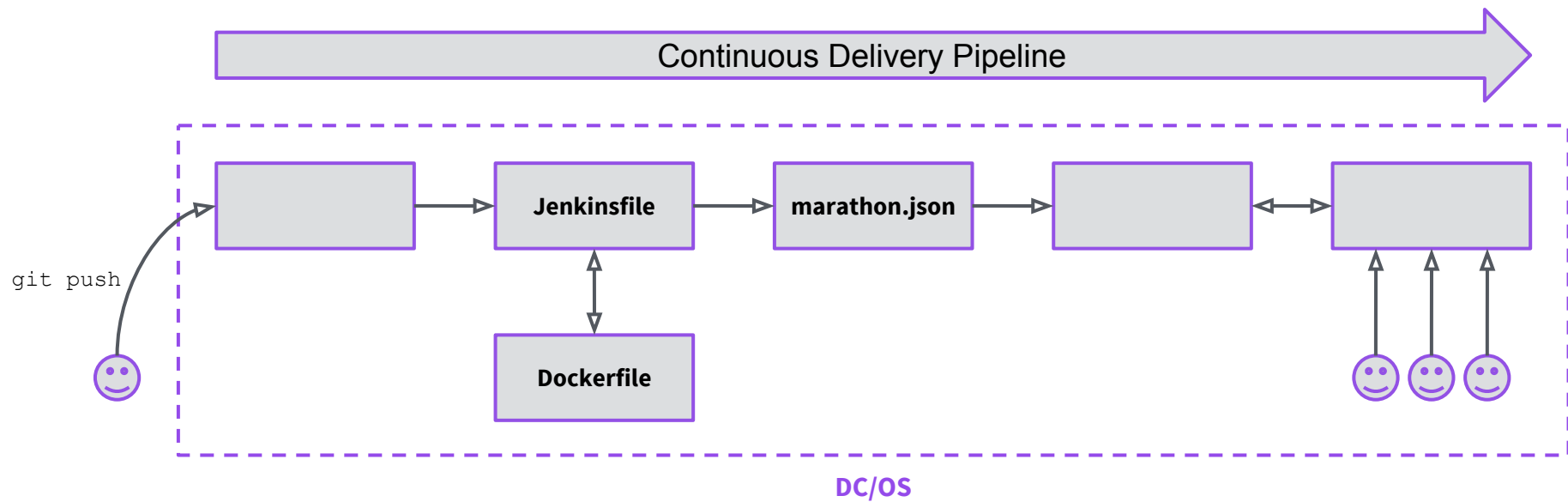
PIPELINE COMPONENTS



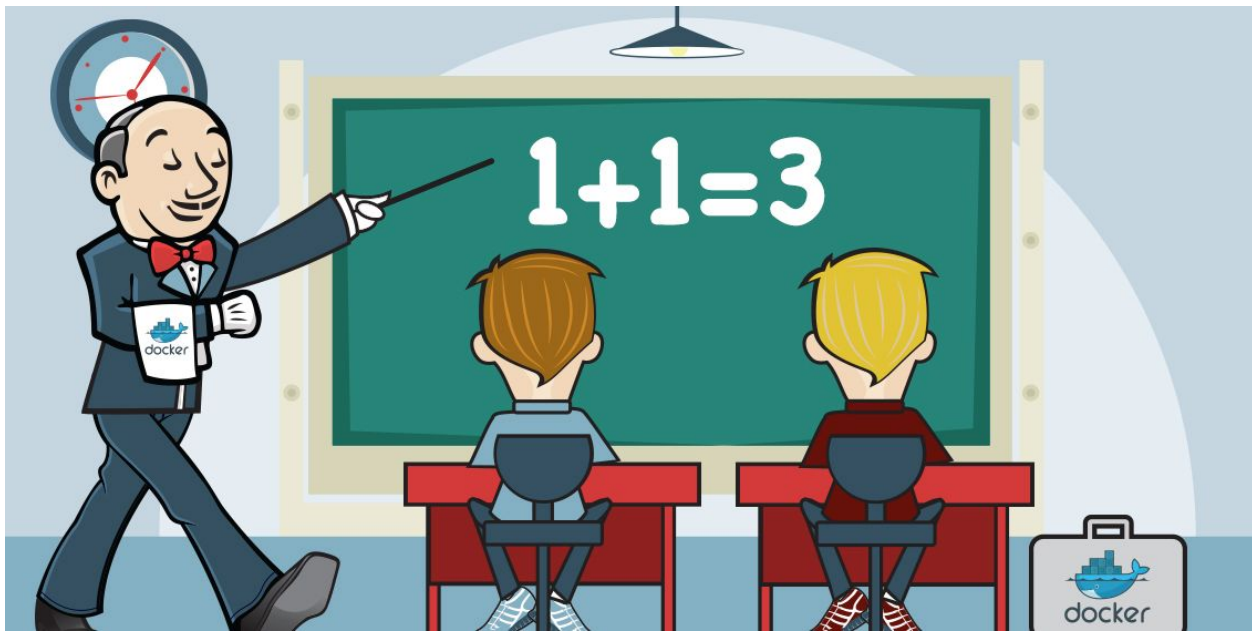
PIPELINE COMPONENTS



PIPELINE CONFIGURATION



DEMOTIME





THANK YOU!

Roger Ignazio
roger@mesosphere.com
@rogerignazio

Sunil Shah
sunil@mesosphere.com
@ssk2

Learn more by visiting [DCOS.io](https://dcos.io) and [Mesosphere.com](https://mesosphere.com)