Sunil Shah

SECURE, FLEXIBLE CONTINUOUS DELIVERY PIPELINES WITH GITLAB AND DC/OS

MESOSPHERE

Introduction

MOBILE, SOCIAL & CLOUD ARE RAISING CUSTOMER EXPECTATIONS

> *“We need a way to deliver software so fast that our customers don’t have time to change their minds”*

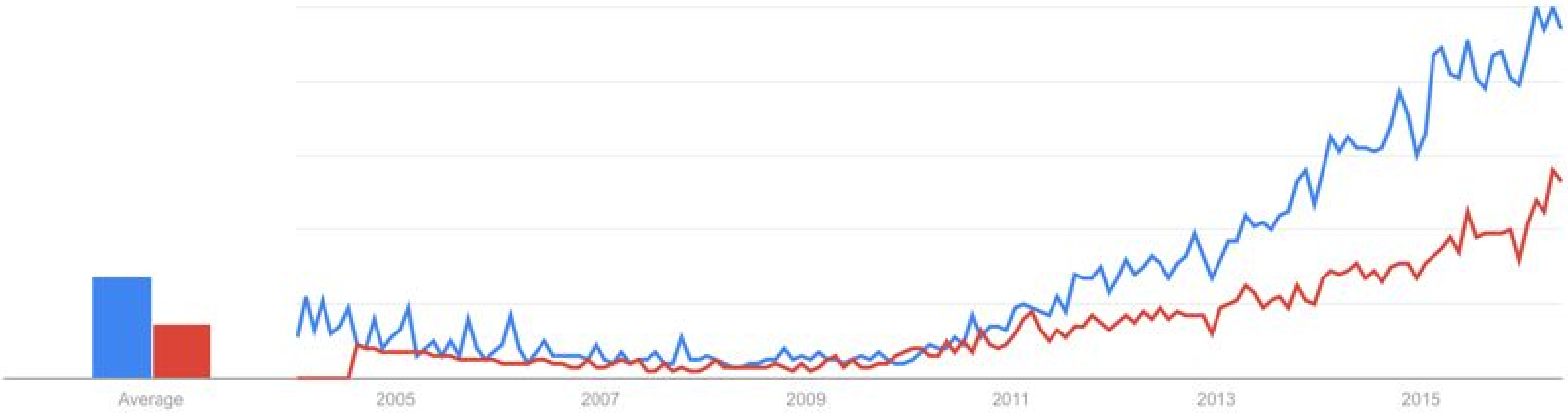
Application development and DevOps teams are under increasing pressure.

- Releasing stable quality code (much) more frequently, with similar or less resources
- Maintaining uptime while supporting faster release cycle

Introduction

WHY BOTHER?

Why is continuous delivery interesting now?



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

Introduction

WHY BOTHER?

1. **It’s much easier to get compute resources nowadays!**
    - Doesn’t cost much
    - **EVERY** platform has an API

amazon web services™

Google Cloud Platform

Microsoft Azure

Introduction

WHY BOTHER?

1. It’s much easier to get compute resources nowadays!
2. **Container orchestration lets you sleep**
   - 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!

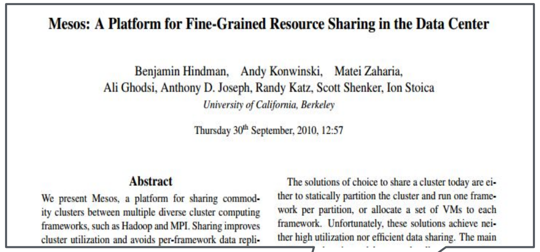
WHY BOTHER?

1. It’s much easier to get compute resources nowadays!
2. Container orchestration lets you sleep
3. **Containers mean you can!**
   - Containers encapsulate everything your application needs to run
   - No need for painful and tedious manual intervention anymore

ABOUT DC/OS

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

DC/OS CAPABILITIES ENABLING DEVELOPER AGILITY

1. One platform for next-gen development technologies
2. Reduced risk of failed deployments
3. Reliable, simplified CI/CD integration
4. One API, deploy anywhere

DC/OS capabilities simplify and accelerate application development lifecycle

MESOSPHERE ENTERPRISE DC/OS, SIMPLIFYING THE OPERATION OF NEXT GENERATION TECHNOLOGIES, AT MASSIVE SCALE

Services & Containers

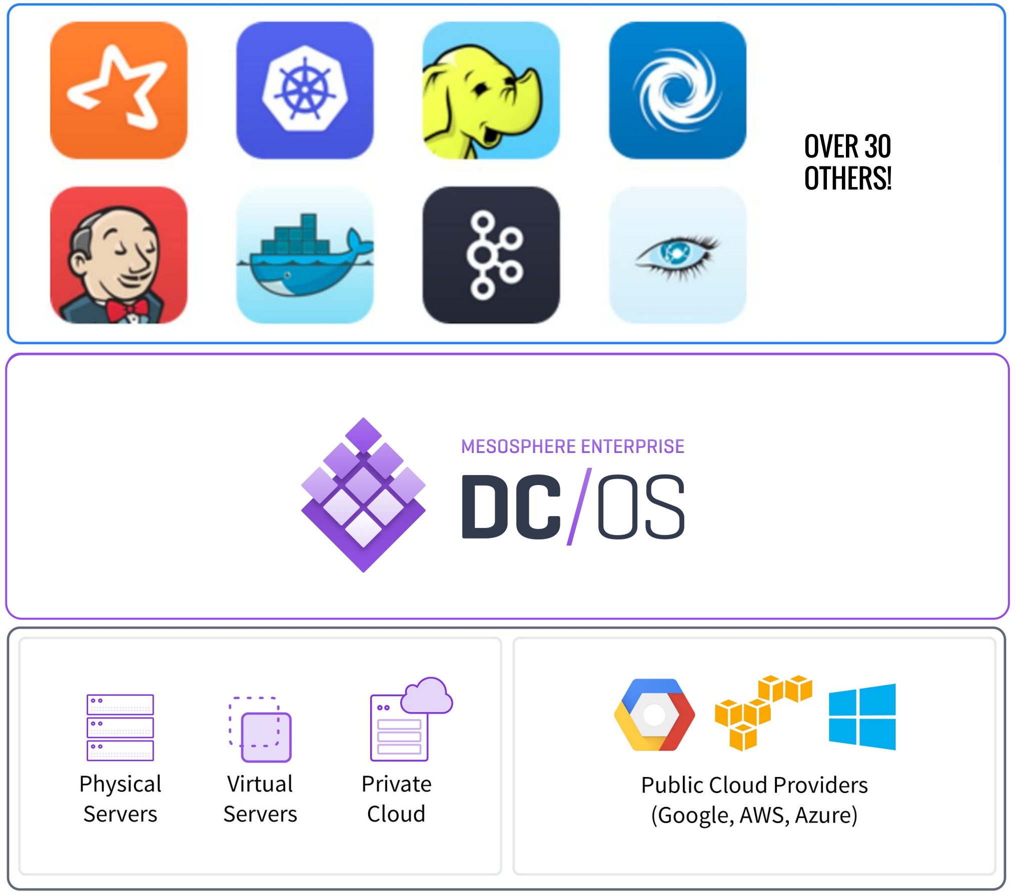
Your favorite services, container formats, and those yet to come.

Mesosphere Enterprise DC/OS

Runs distributed apps anywhere as simply as running apps on your laptop.

Any Infrastructure

Build Apps once in DC/OS, and run it anywhere



OVER 30 OTHERS!

MESOSPHERE ENTERPRISE DC/OS

Physical Servers

Virtual Servers

Private Cloud

Public Cloud Providers (Google, AWS, Azure)

MESOSPHERE ENTERPRISE DC/OS IS THE BEST WAY TO BUILD AND RUN MODERN APPLICATIONS

The Modern Enterprise App

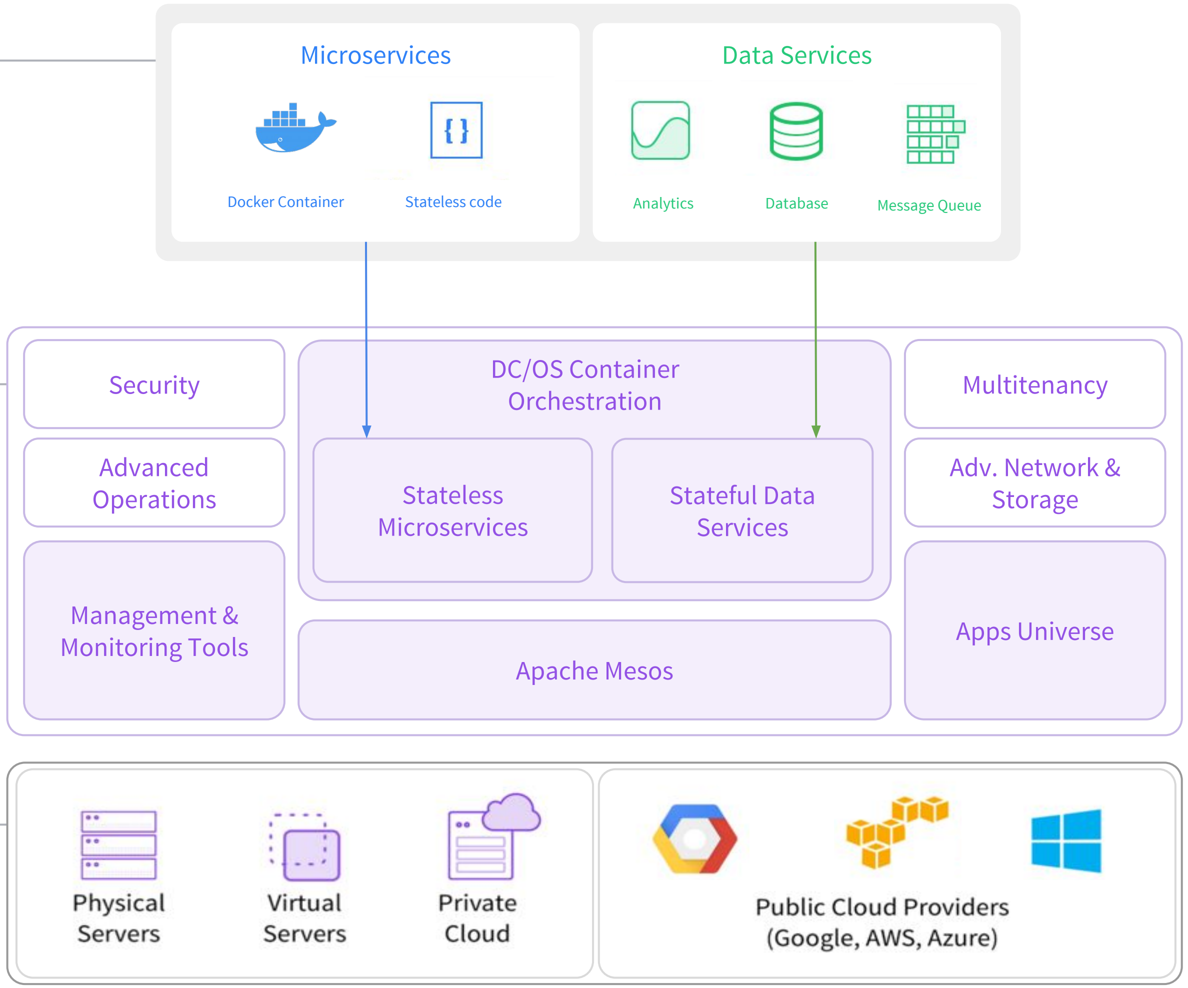
Composed of stateless processes & stateful backing services.

Enterprise DC/OS

The only supported platform that elastically runs the full Modern Enterprise App.

Any Infrastructure

Containerized workloads are scheduled across resources in your private datacenter or in the cloud.



Microservices

Docker Container

Stateless code

Data Services

Analytics

Database

Message Queue

Security

DC/OS Container Orchestration

Multitenancy

Advanced Operations

Stateless Microservices

Stateful Data Services

Adv. Network & Storage

Management & Monitoring Tools

Apache Mesos

Apps Universe

Physical Servers

Virtual Servers

Private Cloud

Public Cloud Providers (Google, AWS, Azure)

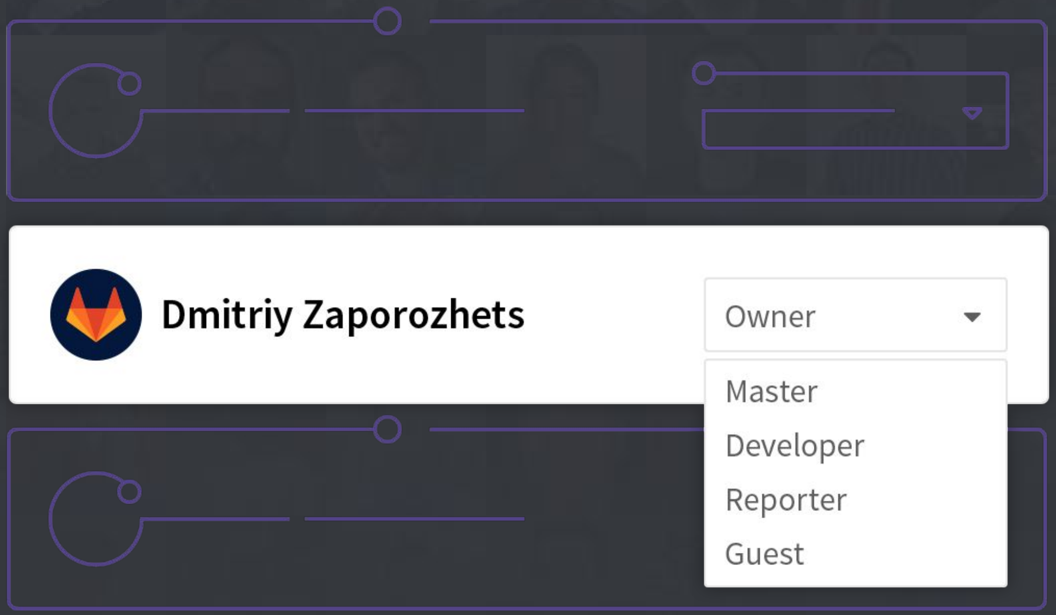
About GitLab

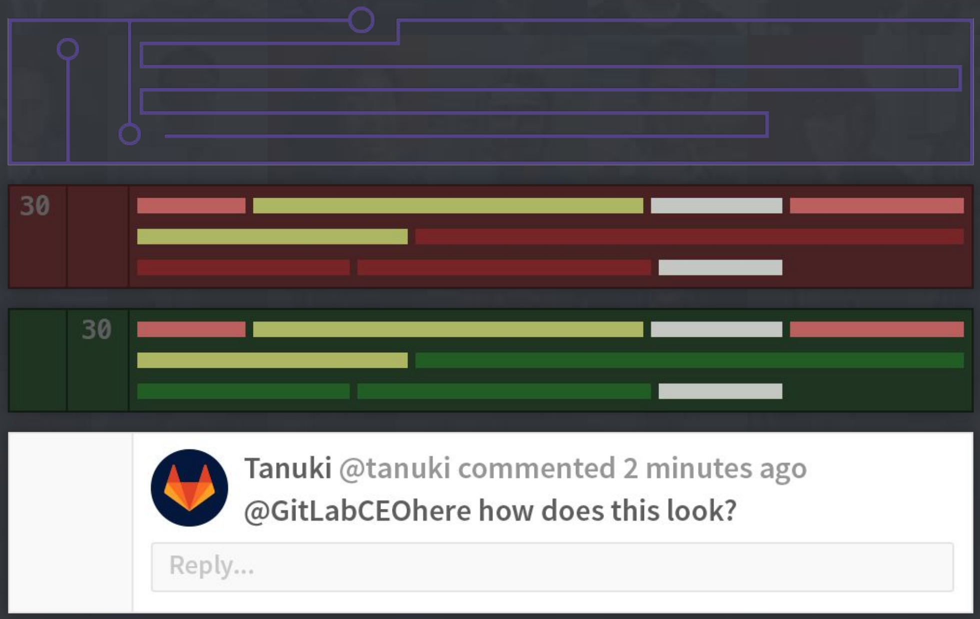
Collaborating with Git

- Modern Platform
- Distributed version control
- Code and commit offline
- Workflow designed around open-source

An integrated set of tools

- Repository Management
- Access Control
- Code Review Tools
- Issue Tracker and Wiki

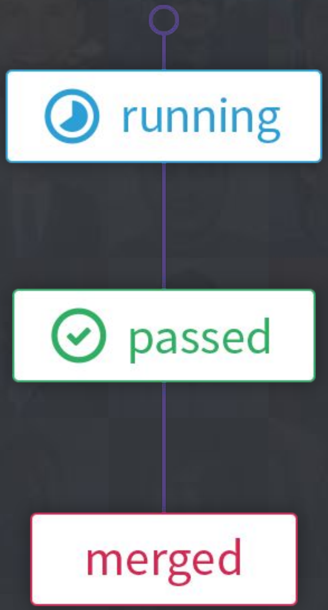


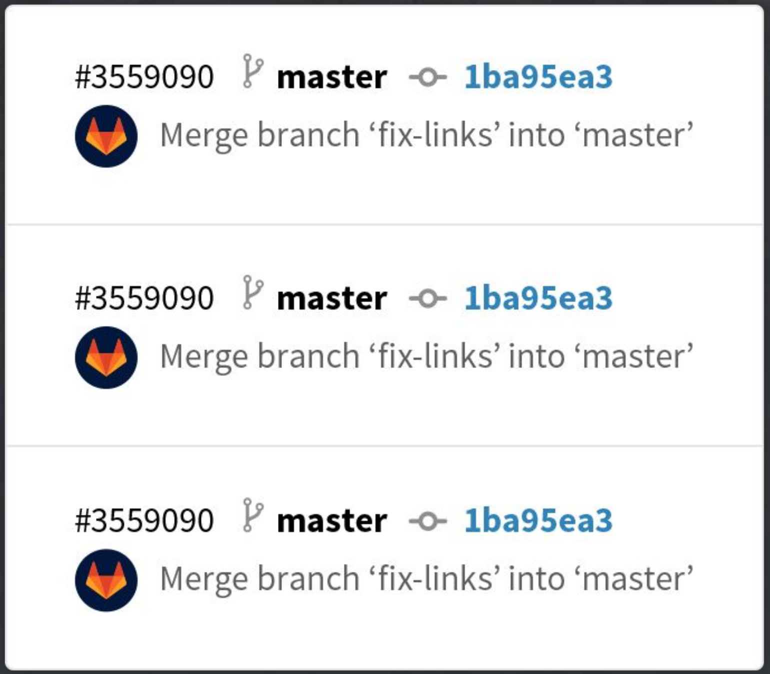


Tanuki @tanuki commented 2 minutes ago
@GitLabCEOhere how does this look?

An integrated set of tools

- Merge Conflict Resolution
- Built-in Continuous Integration
- Built-in Container Registry




#3559090 master 1ba95ea3
Merge branch ‘fix-links’ into ‘master’

#3559090 master 1ba95ea3
Merge branch ‘fix-links’ into ‘master’

#3559090 master 1ba95ea3
Merge branch ‘fix-links’ into ‘master’

History of GitLab

The Product

- Started in 2011

The Website

- Started in 2012

The Company

- Joined forces in 2013, incorporated in 2014

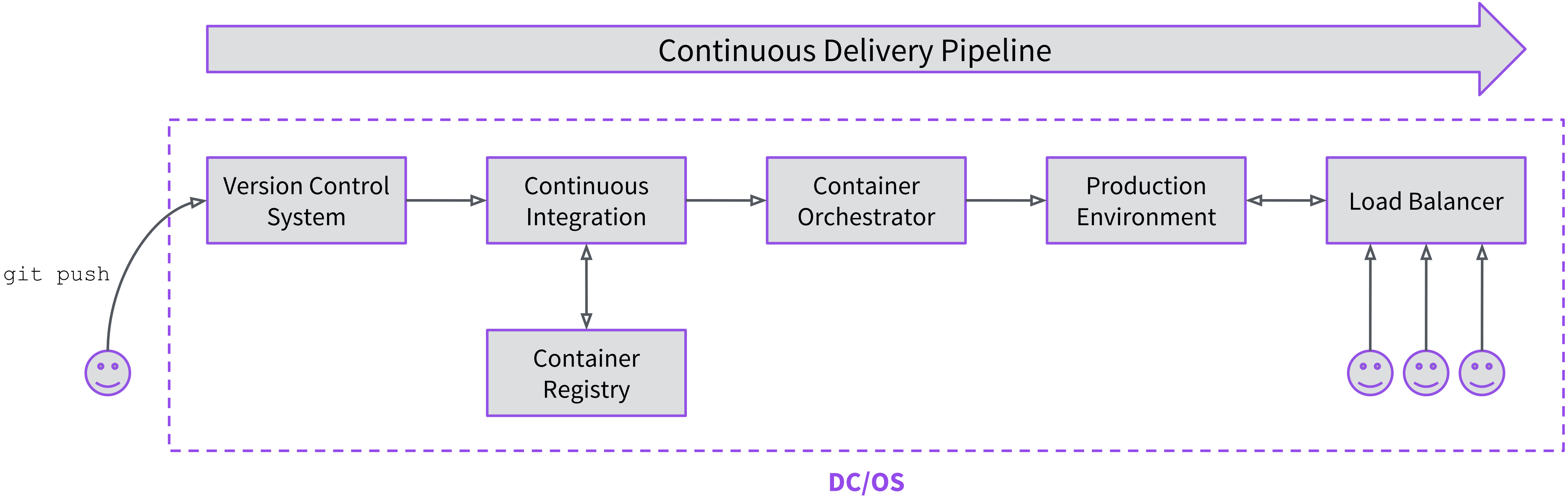
Benefits of GitLab

- An integrated set of tools that scale
- Open Source and updated monthly
- Easy to Install and supports many environments
- Great technical support
- Tons of features and a public roadmap:

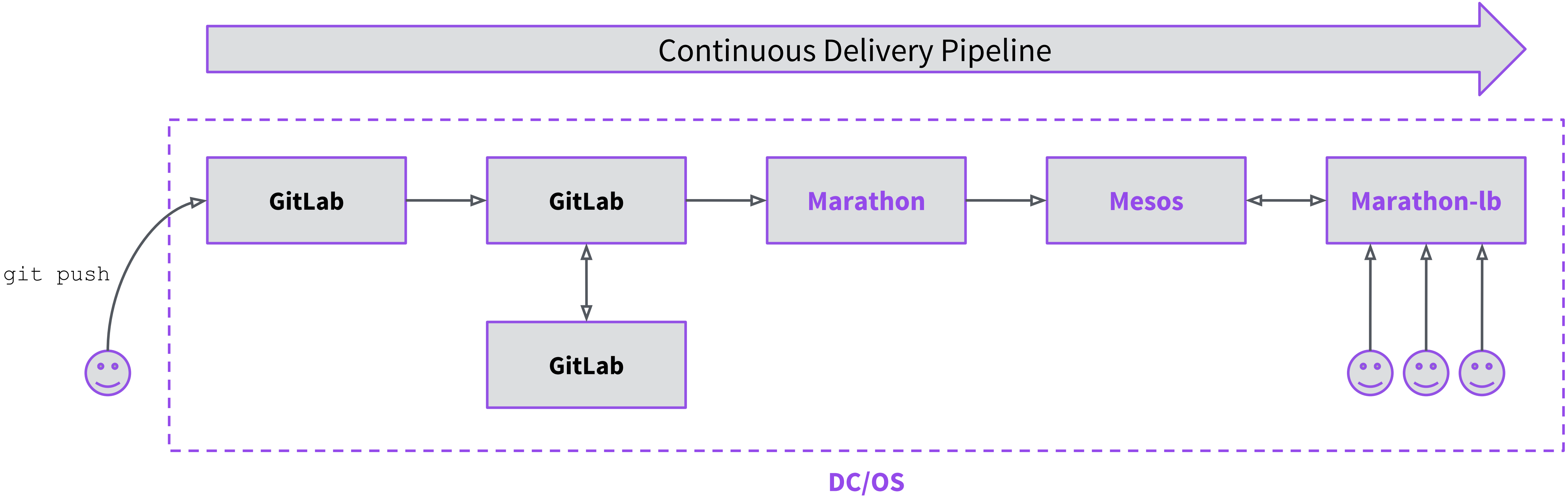
  https://about.gitlab.com/direction/

GITLAB ON DC/OS

CONTINUOUS DELIVERY: MAIN COMPONENTS



GITLAB AS SCM, CI & REGISTRY



GitLab on DC/OS

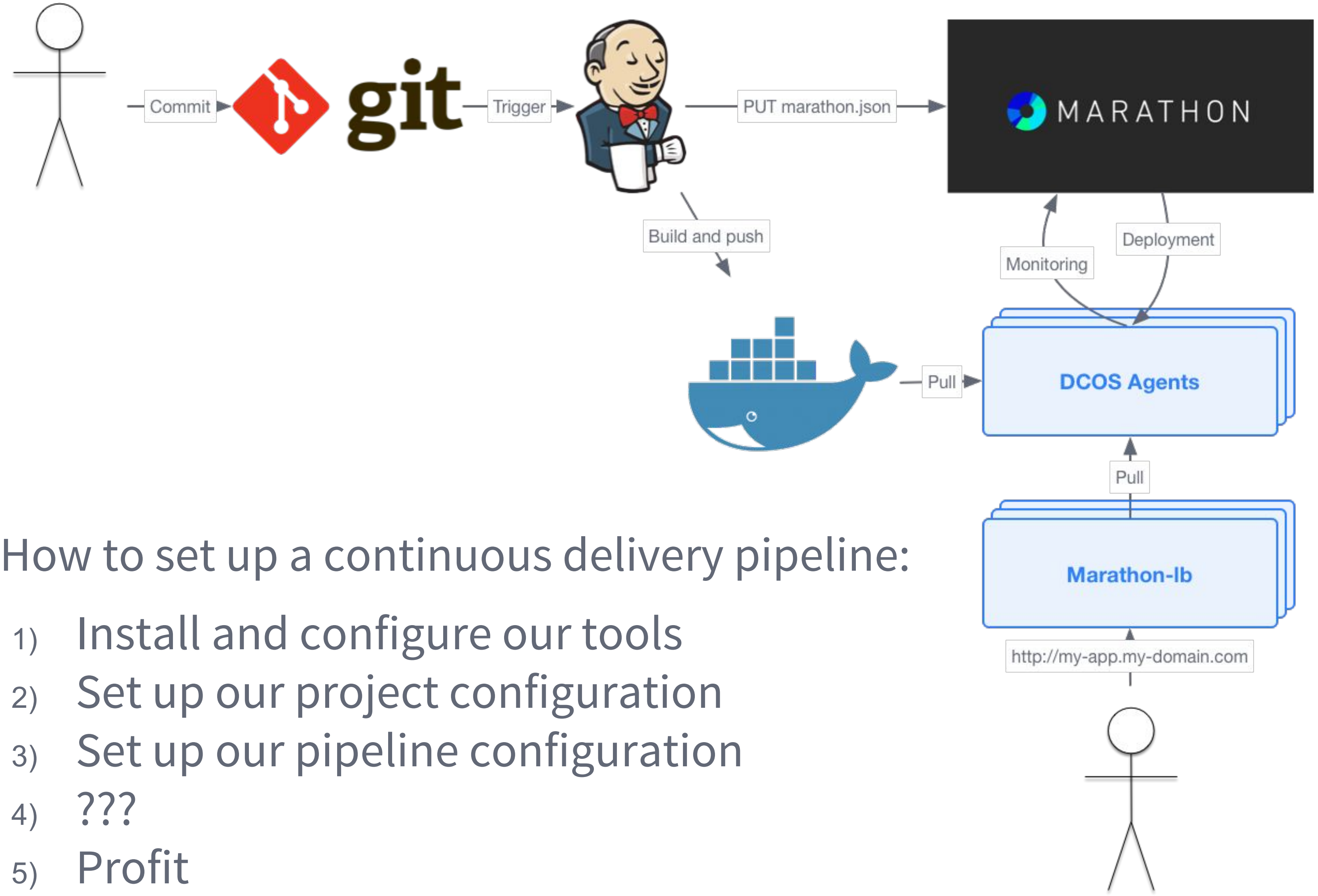
BENEFITS

- Easy installation: set up a highly available deployment within minutes
- Deploy multiple instances of GitLab with ease
- Fault-tolerant: DC/OS keeps GitLab running and your data safe
- Run all of your CI/CD infrastructure in one place

CONTINUOUS DELIVERY USING GITLAB ON DC/OS

CD using GitLab on DC/OS

BASICS

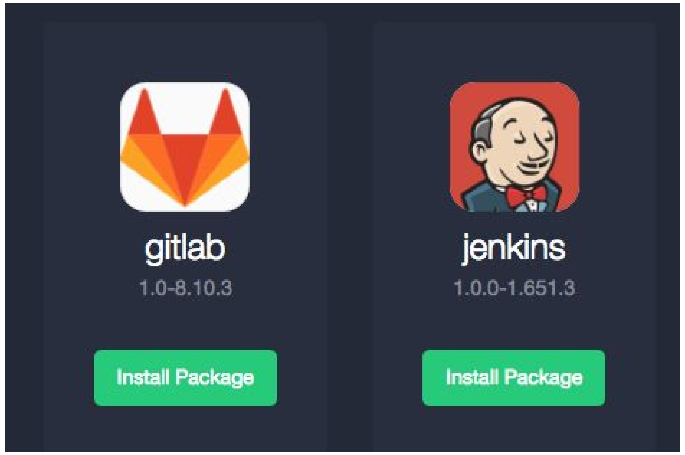


How to set up a continuous delivery pipeline:

1) Install and configure our tools
2) Set up our project configuration
3) Set up our pipeline configuration
4) ???
5) Profit

CD using GitLab on DC/OS

1. INSTALL & CONFIGURE TOOLS



We will use GitLab and Jenkins:

1. Set up CNAMEs for each service
   - jenkins-demo.mesosphere.com
   - gitlab-demo.mesosphere.com
2. Install each of the packages from the DC/OS Universe and configure them to use these CNAMEs
3. (Optional) Configure Jenkins to talk to an insecure registry

CD using GitLab on DC/OS

2. SET UP PROJECT

Our project contains two files that describe how to build and run our application.

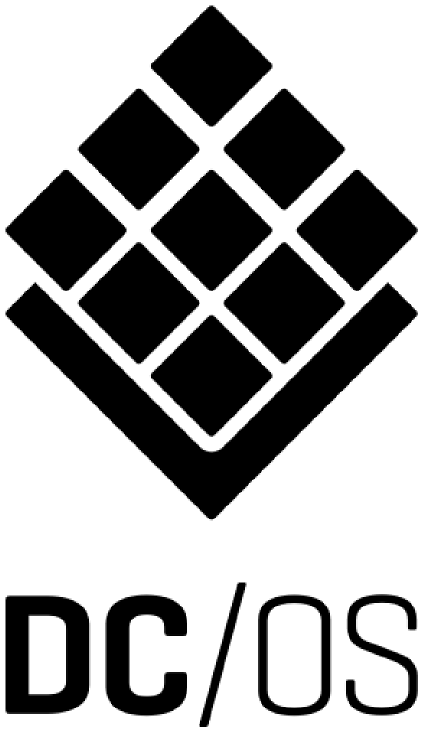
1. A Dockerfile
   - Encapsulates the dependencies required by your application
2. A marathon.json
   - Describes resources required
   - How many instances to run
   - What command to run
   - How to check health

3. CONFIGURE PIPELINE

Finally, we can set up the pipeline itself.

1. Set up GitLab to trigger a Jenkins build
2. Jenkins should be configured with the credentials and certificates required to talk to GitLab
3. Add a post build step that triggers a Marathon deployment
   - Marathon will perform a rolling upgrade of your application so that instances remain available while being upgraded

GITLAB ON DC/OS: AVAILABLE 15/9



Q & A

- Email me at sunil@mesosphere.io
- See https://mesosphere.com and https://dcos.io