
Continuous Deployment at Enterprise Scale with Artifactory and DC/OS

swampUP 2017
Elizabeth K. Joseph
@pleia2

Elizabeth K. Joseph, Developer Advocate, Mesosphere

- 15+ years working in open source communities
- 10+ years in Linux systems administration and engineering roles
- 4 years working on CI/CD for the OpenStack project
- Author of [The Official Ubuntu Book](#) and [Common OpenStack Deployments](#)

Confession Time

Infrastructure vs. Applications

Containerization with DC/OS creates abstraction between infrastructure and applications so teams can focus

Infrastructure

Infrastructure admins can focus on providing a fast, efficient and dynamic environments for application developers to deploy on

- Manage underlying “physical” system (whether bare metal or management of a cloud service)
- Keep underlying infrastructure operating system patched and upgraded
- Maintain resource abstraction clusters (like Apache Mesos)

Applications

Application developers can focus on building and maintaining the applications they're running

- Launch and run applications in containers
- Instead of upgrading container environments running in production, create and set up a rolling upgrade to deploy the new images
- Independently monitor and maintain just the application layer without worrying about maintaining the underlying infrastructure (network, storage, etc)

Introducing DC/OS

Built on Apache Mesos, DC/OS abstracts your data center into a single computer, pooling distributed workloads and simplifying rollout and operations.

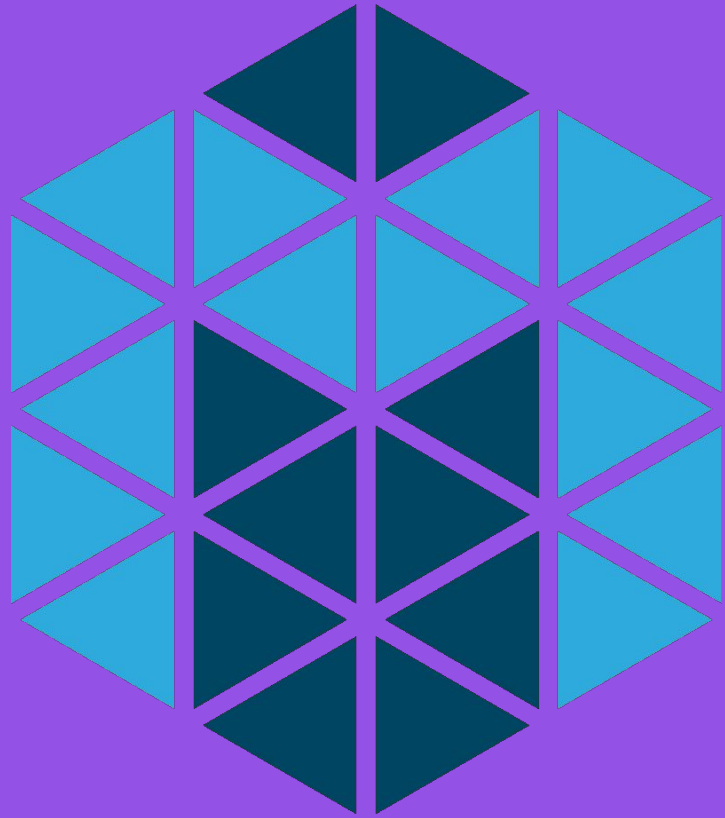
Open Source and Enterprise versions available.



DC/OS

Some of what DC/OS gets you

- Container orchestration
- Resource and network management
- Task scheduling
- Local DNS configuration for simplified discovery
- Unified logging and metrics
- “Universe” catalog of pre-configured services, including Artifactory and Jenkins

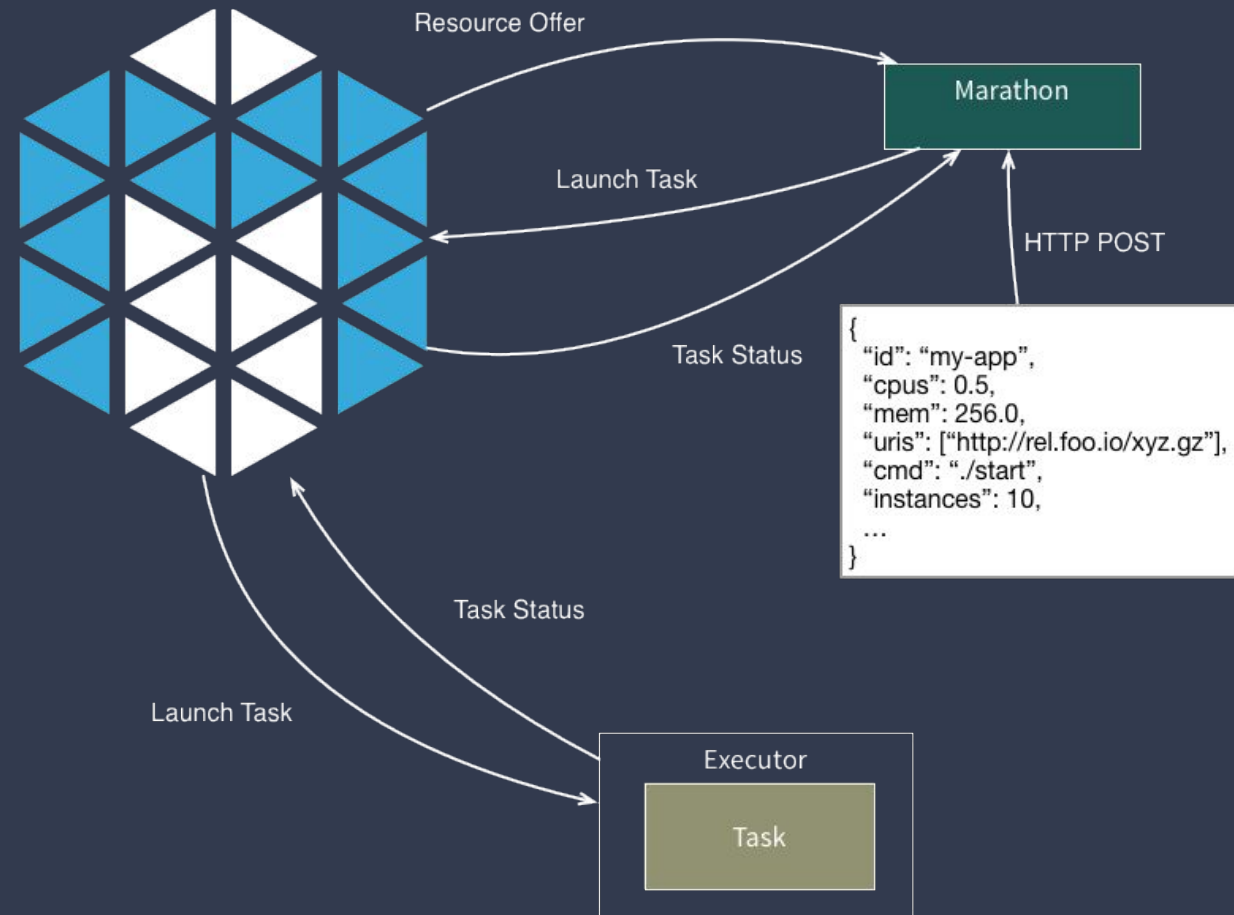


Apache Mesos: The datacenter kernel

<http://mesos.apache.org/>

Marathon

- Mesos can't manage running applications on its own.
- A Mesos framework is a distributed system that has a scheduler.
- Schedulers like Marathon start and keep your applications running. A bit like a distributed init system.
- Mesos mechanics are fair and HA
- Learn more at <https://mesosphere.github.io/marathon/>



DC/OS Architecture Overview

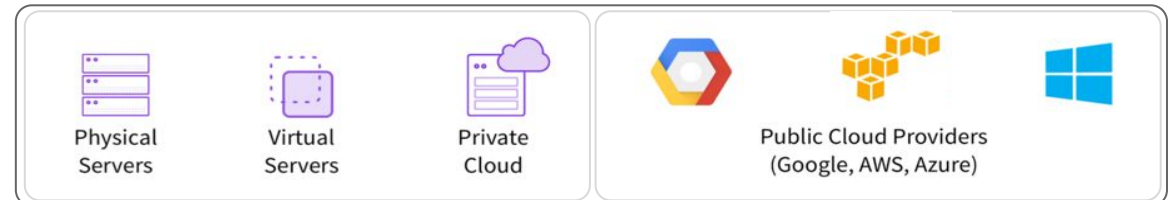
Services & Containers



DC/OS



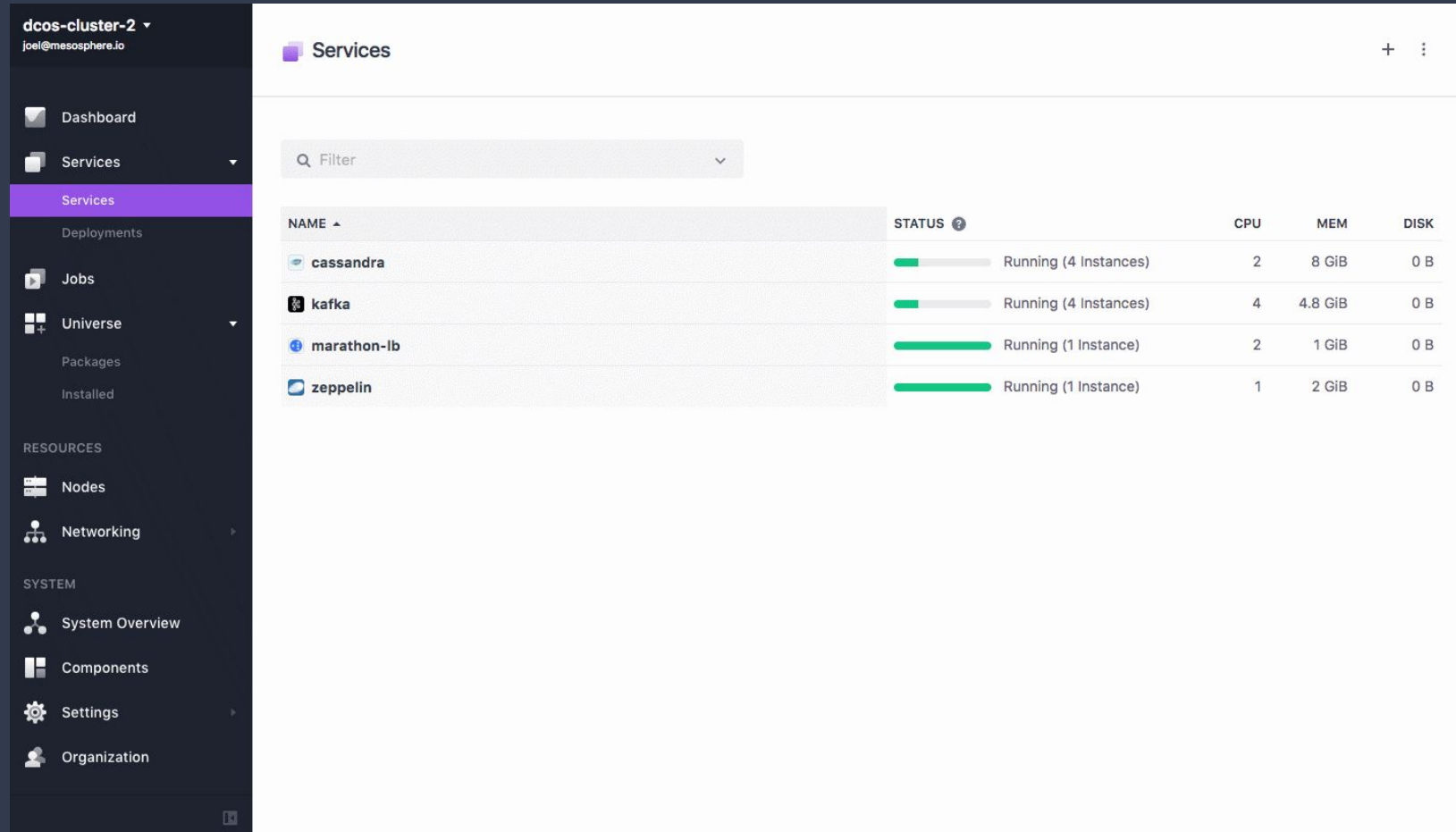
ANY INFRASTRUCTURE



Interact with DC/OS (1/2)

Web-based GUI

<https://dcos.io/docs/latest/usage/webinterface/>



The screenshot shows the DC/OS web interface for a cluster named 'dcos-cluster-2'. The left sidebar is dark-themed and contains navigation options: Dashboard, Services (selected), Deployments, Jobs, Universe, Packages, Installed, RESOURCES (Nodes, Networking), and SYSTEM (System Overview, Components, Settings, Organization). The main content area is titled 'Services' and features a search filter. Below the filter is a table listing services with their status, instance count, and resource usage.

NAME	STATUS	CPU	MEM	DISK
cassandra	Running (4 Instances)	2	8 GiB	0 B
kafka	Running (4 Instances)	4	4.8 GiB	0 B
marathon-lb	Running (1 Instance)	2	1 GiB	0 B
zeppelin	Running (1 Instance)	1	2 GiB	0 B

Interact with DC/OS (2/2)

CLI tool

<https://dcos.io/docs/latest/usage/cli/>

API

<https://dcos.io/docs/latest/api/>

Jenkins + Mesos

The Mesos plugin for Jenkins allows Jenkins to dynamically launch Jenkins slaves on a Mesos cluster.

“Put simply, whenever the Jenkins Build Queue starts getting bigger, this plugin automatically spins up additional Jenkins slave(s) on Mesos so that jobs can be immediately scheduled! Similarly, when a Jenkins slave is idle for a long time it is automatically shut down.”

Source: <https://github.com/jenkinsci/mesos-plugin>

+ Mesos DNS: <https://mesosphere.github.io/mesos-dns/>

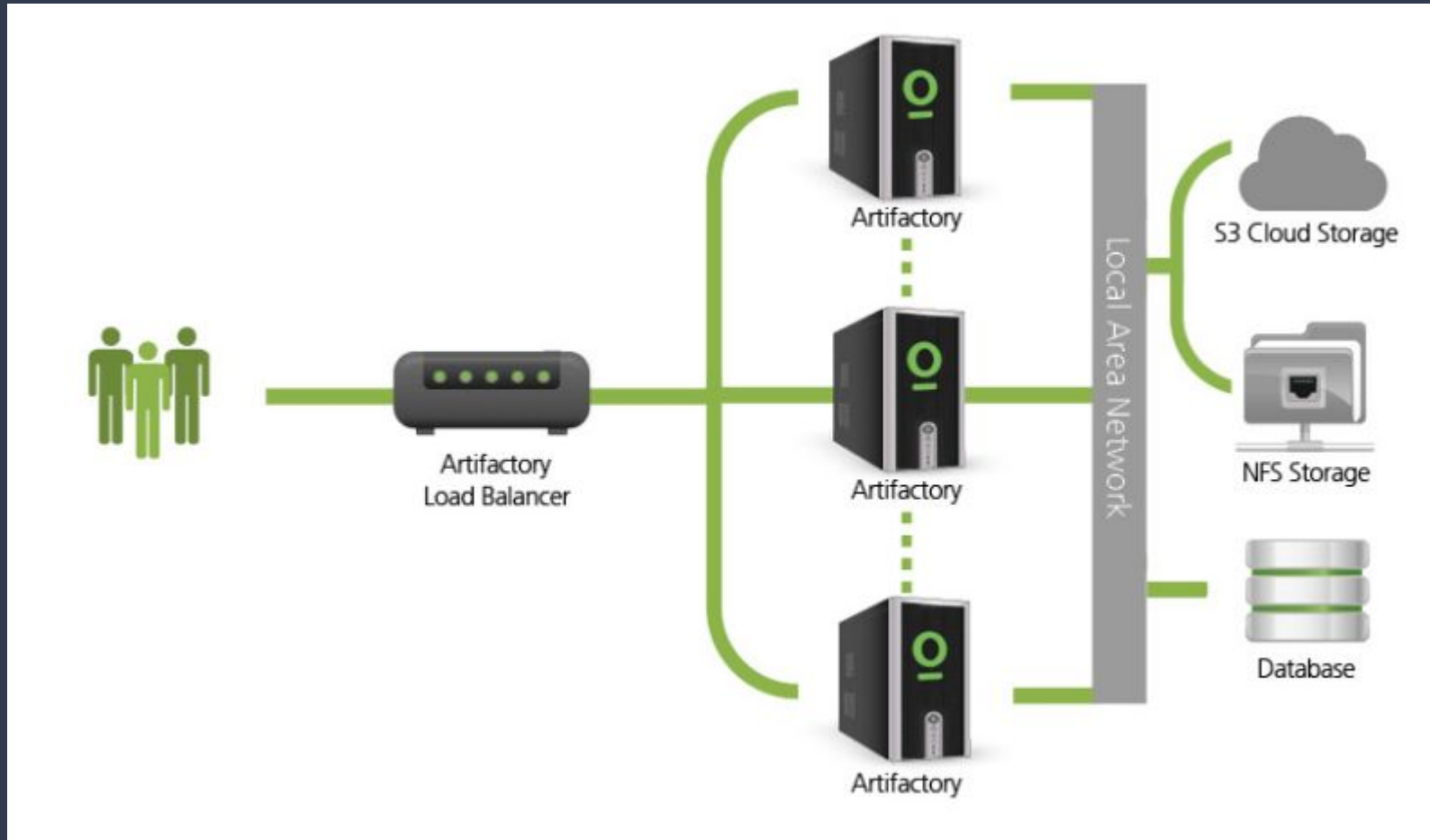
+ Artifactory

DC/OS supports simple installation of the Pro and Enterprise versions of Artifactory, basic installation example for both versions at:

<https://github.com/dcos/examples/tree/master/artifactory>

Jenkins in DC/OS also comes with the Artifactory plugin

Artifactory Enterprise HA on DC/OS



Artifactory Enterprise Configuration

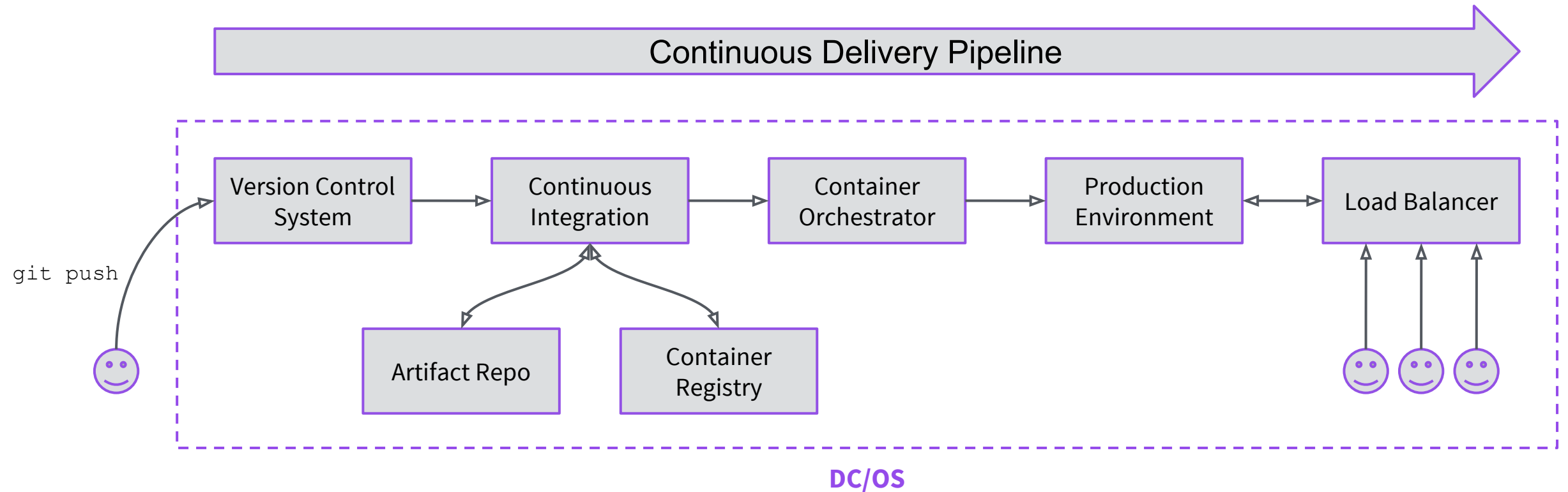
The screenshot shows the Artifactory configuration page for version 5.1.4. On the left, a sidebar lists 'service', 'pro', and 'enterprise' (selected). The main content area is titled 'enterprise' and contains the following sections:

- enterprise**
Configuration for Artifactory Enterprise (a highly-available deployment of Artifactory). If enabled, this will override any Artifactory Pro configuration.
 ENABLED * ?
- NETWORK** ?
[Empty text input field]
- secondary**
Configuration properties for the Artifactory Enterprise secondary node.
 ENABLED * ?
- UNIQUE-NODES** ?
- NODES** ?
[Input field containing '1']
- NAME** ?
[Empty text input field]

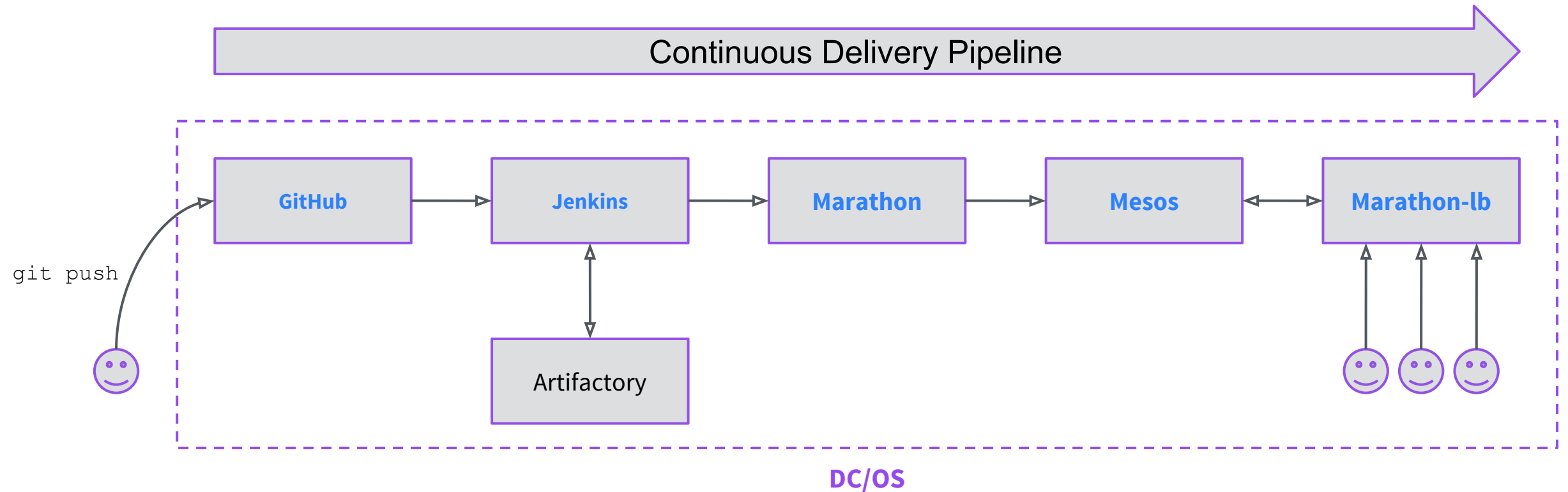
At the bottom, there are two buttons: 'BACK' on the left and 'REVIEW AND INSTALL' on the right.

CI/CD Demo!

CI/CD PIPELINE (general)



CI/CD PIPELINE (specifics)



Demo

1. Clone the git repository
2. Build a Docker container based off the Jekyll Docker image that includes the content stored in /site and push it to the Artifactory Docker Registry.
3. Run the newly created container and a Linkchecker container that runs a basic integration test against the container, checking that the web server comes up correctly and that all links being served are valid (i.e. no 404s).
4. Manually trigger a Marathon deployment of the newly created container, served from the Artifactory Docker Registry, to the DC/OS base Marathon instance. If the application already exists, Marathon will simply upgrade it.
5. Make the application available on a public agent at port 80 using Marathon-lb.

Demo

Already have Open Source DC/OS cluster running with:

- Jenkins
- MySQL (with user + empty DB for Artifactory)
- Docker on agents configured to accept an insecure Docker registry

Add:

- Artifactory Pro
- Artifactory load balancer for DC/OS
- SSL exception for Artifactory in Jenkins
- Jenkins CI/CD pipeline that runs the demo

- Dashboard
- Services
- Services**
- Deployments
- Jobs
- Universe
- RESOURCES
- Nodes
- Networking
- SYSTEM
- System Overview

Services



Filter

NAME	STATUS	CPU	MEM	DISK
artifactory	Running (1/1)	2	2 GiB	0 B
artifactory-lb	Running (1/1)	1	1 GiB	0 B
jenkins	Running (1/1)	4.6	6.1 GiB	0 B
jenkins-deployed-app	Running (1/1)	0.1	128 MiB	0 B
marathon-lb	Running (1/1)	2	1 GiB	0 B
mysql	Running (1/1)	0.5	512 MiB	0 B



@dcos



chat.dcos.io



users@dcos.io



/dcos

/dcos/examples

/dcos/demos

Thank you!

(and thanks Sunil Shah)

Elizabeth K. Joseph

Twitter: @pleia2

Email: lyz@princessleia.com