开源社
kaiyuanshe

2016中国开源年会

China Open Source Conference 2016

时间：2016年10月15日-16日

地点：北京航空航天大学

基于DC/OS的现代应用实践

Sam Chen

2016.10

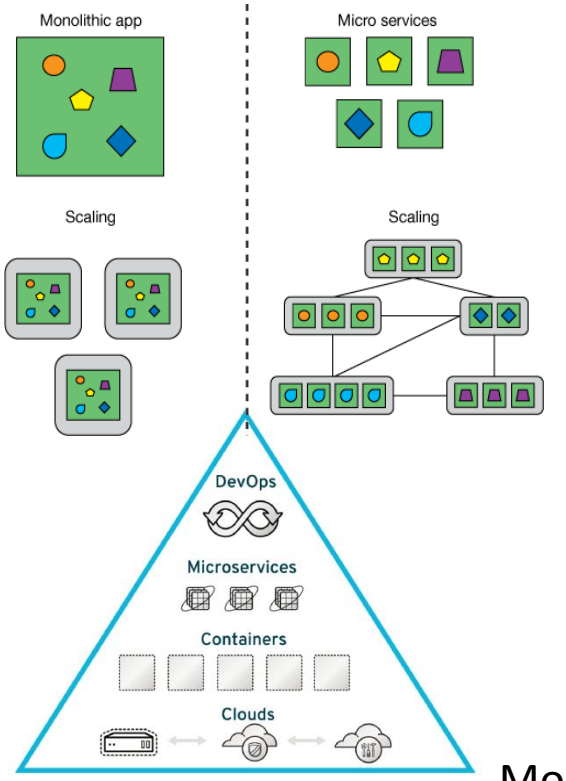
个人简介

- DC/OS中国社区创始人和社区项目孵化者。
- 陈冉(Sam)先生曾任职Linker Networks首席技术官和技术副总裁，主要负责Linker 的技术路线制定，新技术孵化和产品线推广。在这之前，在惠普云事业部担任首席架构师和首席技术官。
- Mesos/OpenStack/Docker/CF社区共同负责人。
- OSCAR DCOS PL/Core。
- 开源布道者。
- Linux 基金会中国协调人

演讲内容

- 概念解释
- 微服务
- 现代应用
- DC/OS是什么
- DC/OS vs. Docker vs. K8S vs. Openstack
- 如何现实现代应用在DC/OS架构
- DC/OS为什么“与众不同”
- 落地最佳实践
- 案例分析

概念介绍



Services & Applications

Easily deploy and run datacenter-wide app services such as Docker, Cassandra, and Spark pooled on a single platform

DC/OS Powered by Apache Mesos

Runtime, tools and best practices built-in to simplify operations and deliver a production self-healing infrastructure

Run Anywhere

Bare-metal, virtual, cloud or hybrid - DC/OS runs on it all - only requirement is a modern Linux distro; Windows support coming soon

Modern Apps= (*DC/OS +Micro services +CI/CD + Hybrid Cloud*)

开源社 kaiyuanshe

Modern Apps= *DC/OS +Micro servic*

微服务要素

The Twelve Factors

1. Code Base
One codebase tracked in revision control, many deploys

2. Dependences
Explicitly declare and isolate dependences

3. Config
Store config in environment

4. Backing Services
Treat backing services as attached resources

5. Build, release, run
Strictly separate build and run stages

6. Process
Execute the app as one or more stateless processes

7. Port Binding
Export services via port binding

8. Concurrency
Scale out via process model

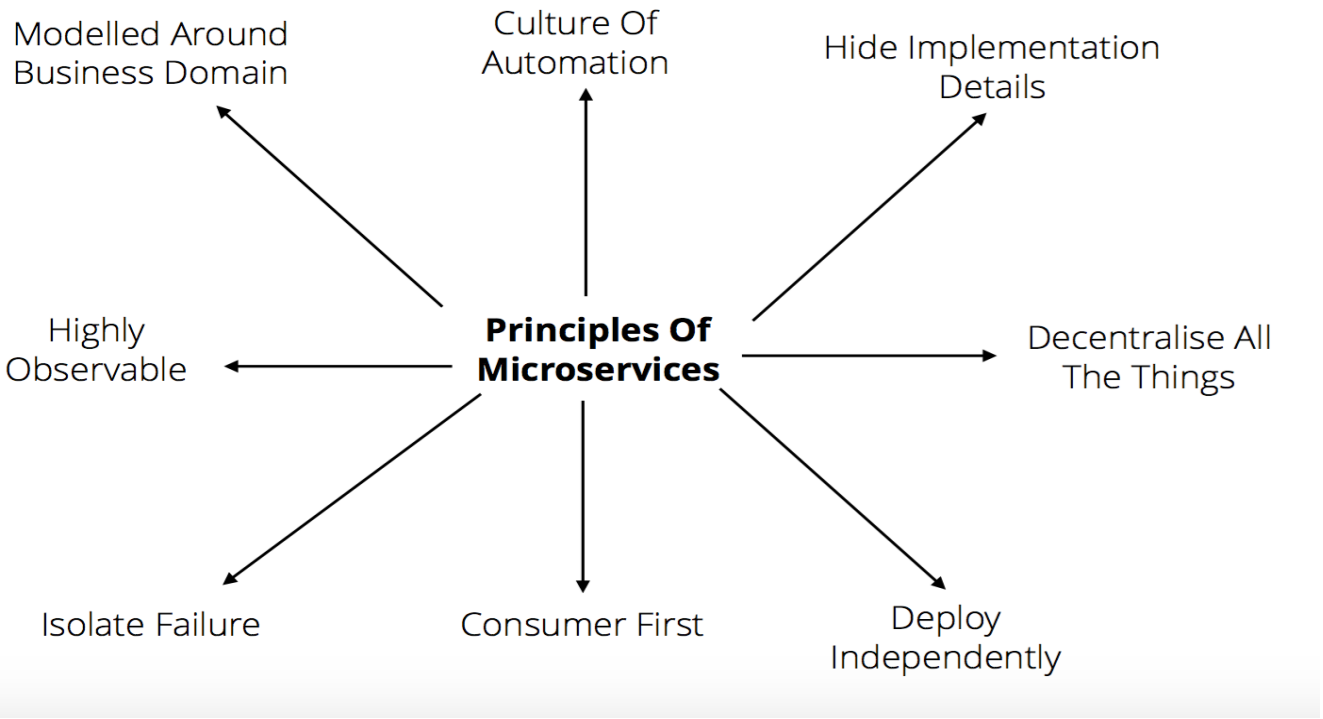
9. Disposability
Maximize robustness with fast startup and graceful shutdown

10. Dev/prod parity
Keep development, staging, and production as similar as possible

11. Logs
Treat logs as event stream

12. Admin processes
Run admin/management tasks as one-off processes

微服务准则



什么是现代应用

UNIT OF INTERACTION

NEW FORM FACTOR FOR DEVELOPING AND RUNNING APPS

PARTITION (LPAR) | SERVER | VIRTUAL MACHINE | **DATACENTER**

MAINFRAME

- DATA / TRANSACTION PROCESSING
- UNIX, IBM OS/360

PHYSICAL (x86)

- ERP, CRM, PRODUCTIVITY, MAIL & WEB SERVER
- LINUX, WINDOWS

VIRTUAL

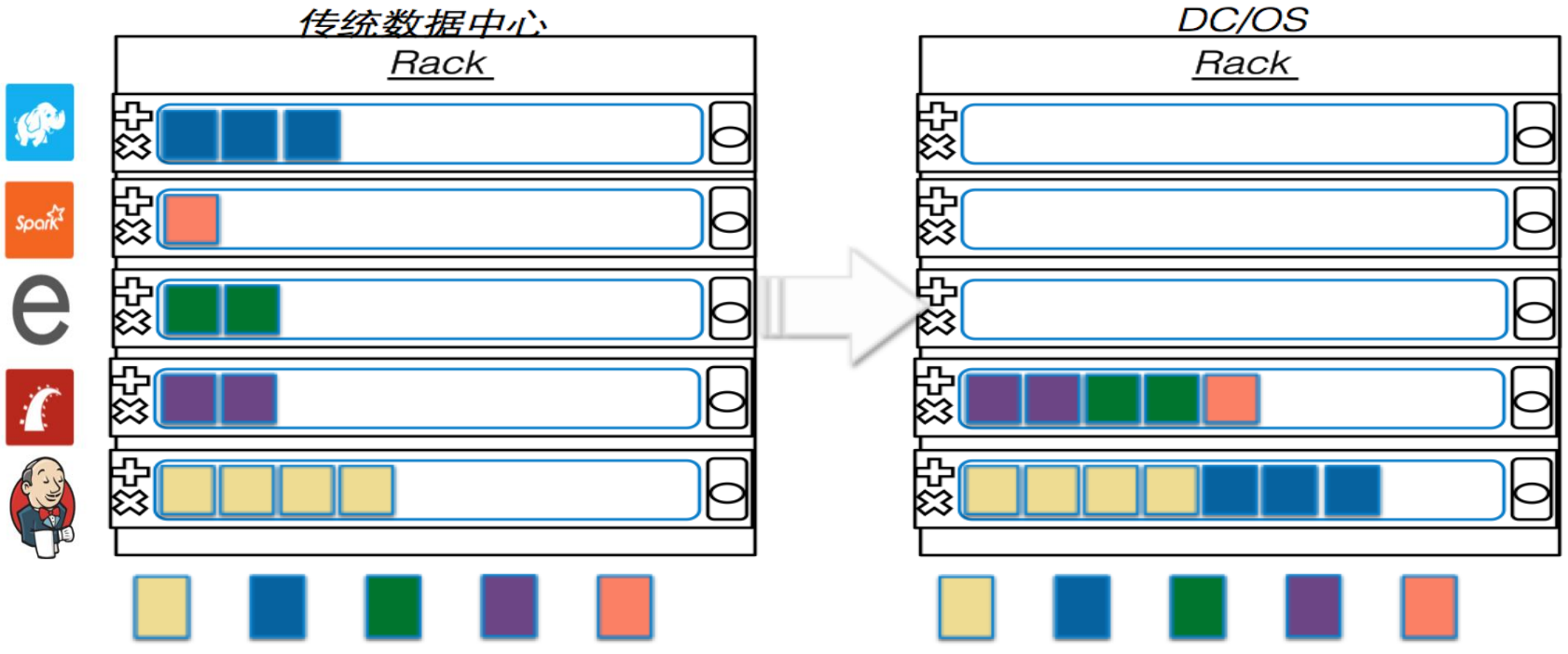
- ERP, CRM, PRODUCTIVITY, MAIL & WEB SERVER
- HYPERVISOR + GUEST OS

HYPERSCALE

- BIG DATA, INTERNET OF THINGS, MOBILE APPS
- **DATACENTER OPERATING SYSTEM (DC/OS)**

DEFINITIVE

DC/OS是什么

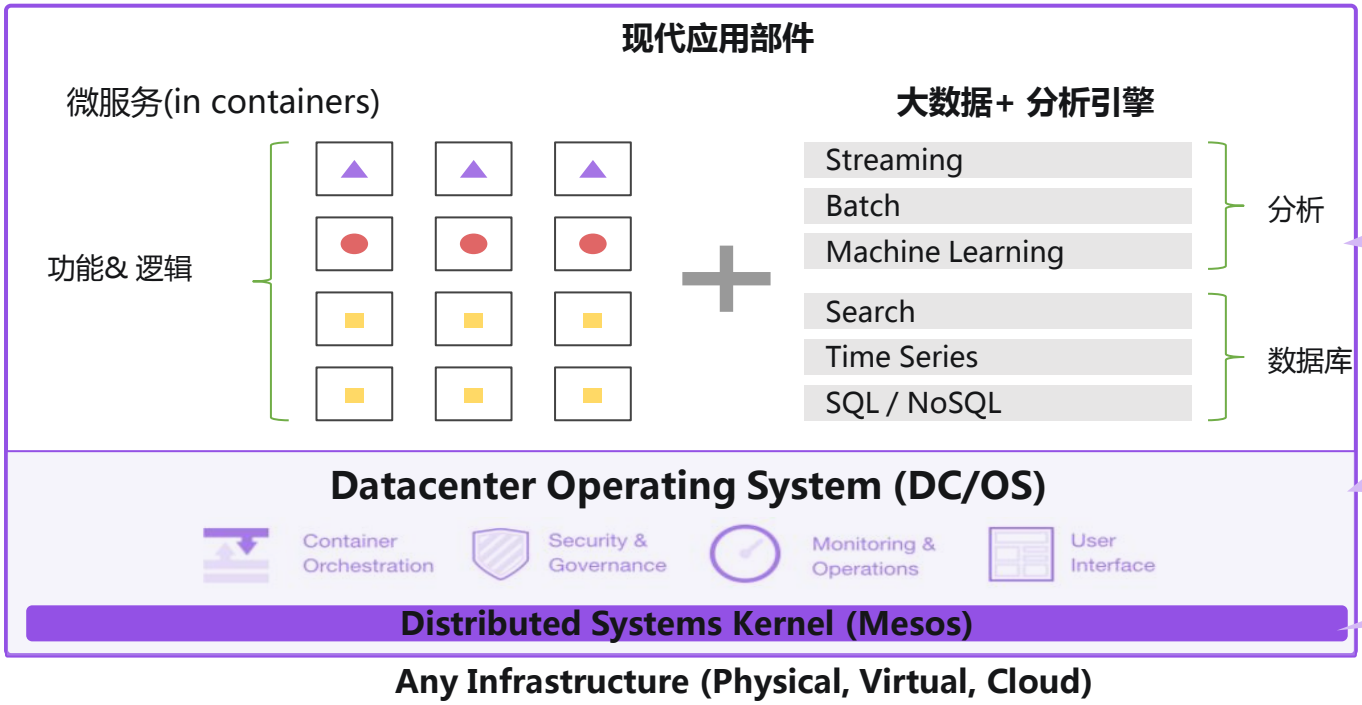


Colorful rectangles are any distributed services running on pre-allocated servers with characteristics of low resource utilization andcomplicated operation management

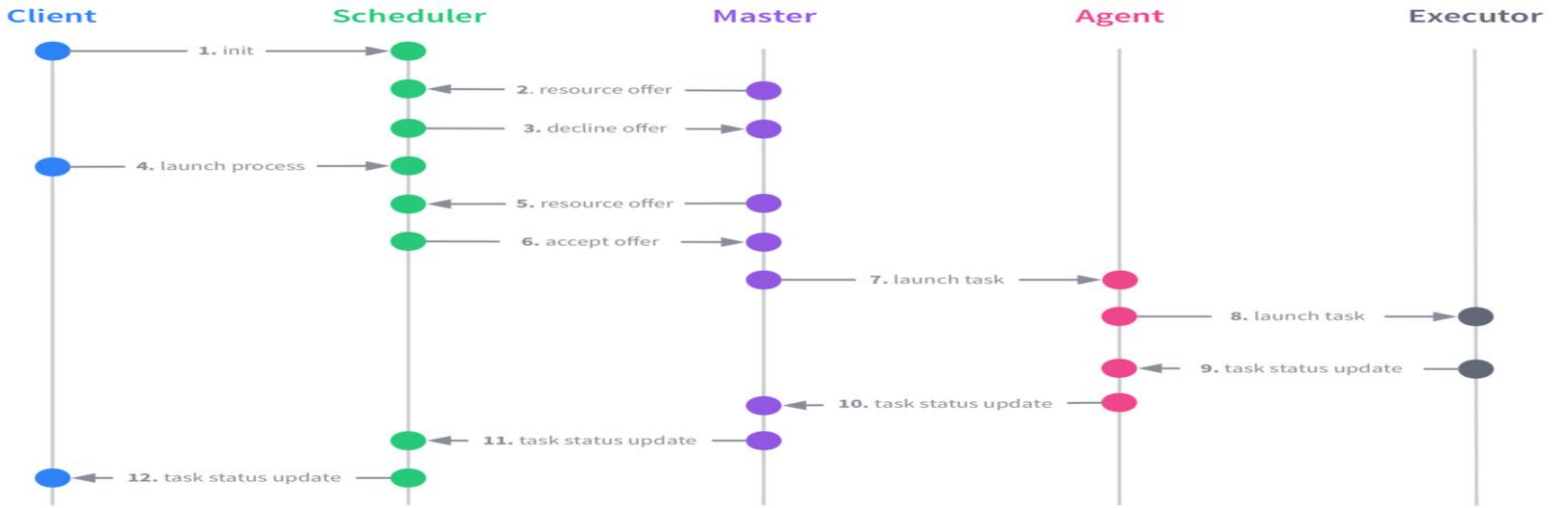
DC/OS maximizing resource utilization with higher availability, elasticity and robust in fabric and aggregated datacenter

DC/OS与现代应用

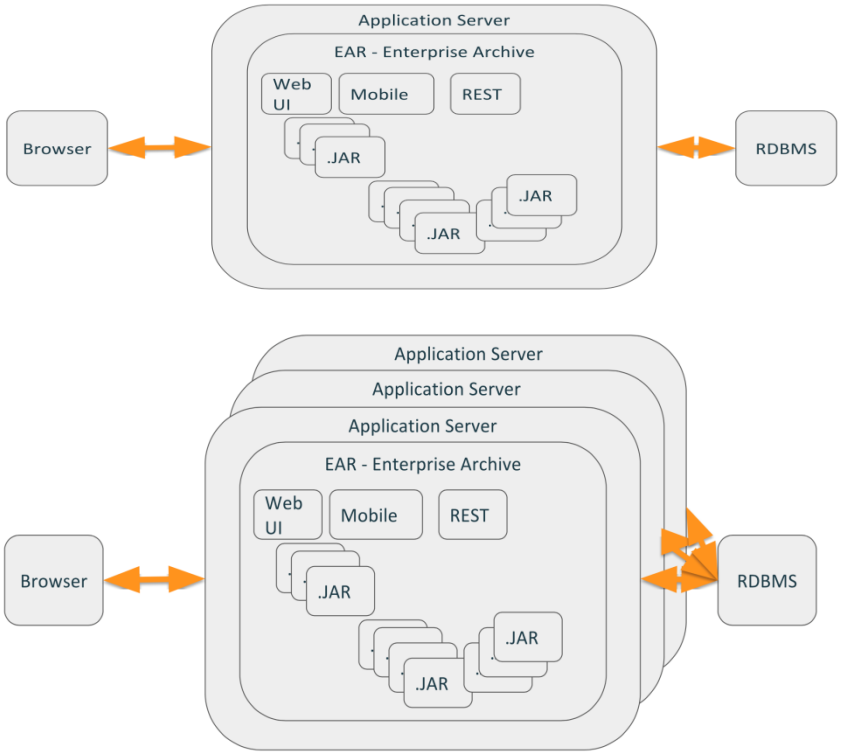
- 100% 开源(ASL2.0) + 一个生态 + 一个多元化的社区
- 在此之下有 ~30 OSS 项目 + 路线图和设计 + The build tool chain + Docs and tutorials + Not limited in a



DC/OS在现代应用中的优势



传统应用现状

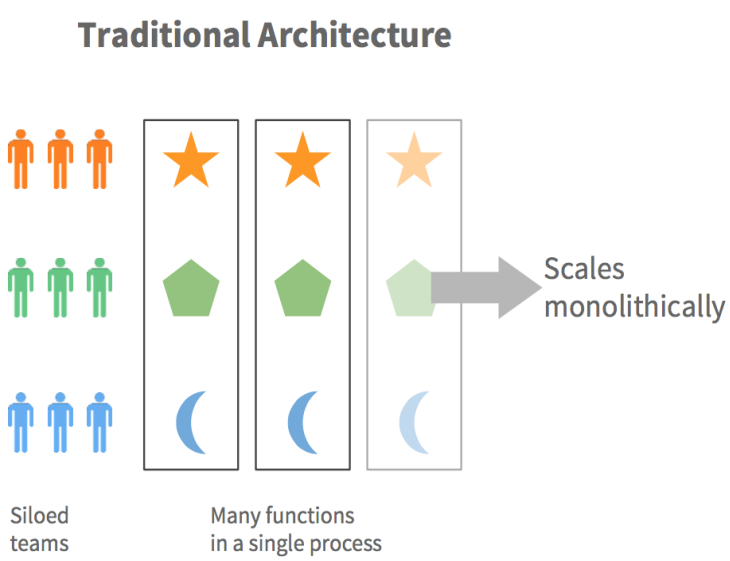


- Monolithic application – everything is package into a single .ear
- Reuse primarily by sharing .jars
- A “big” push to production once or twice a year
- Single database schema for the entire application
- >= Heavyweight Infrastructure
- Thousands of Testcases
- Barely New Testcases
- >= 20 Team Member
- The single .ear requiring a multi-month test cycle /
- Huge bug and feature databases
- User Acceptance Undefined
- Technical Design Approach
- Barely Business Components or Domains
- Requiring multiple team involvement & significant oversight
- Technical Dept
- Grown applications
- Outdated Runtimes (Licenses, Complex Updates）

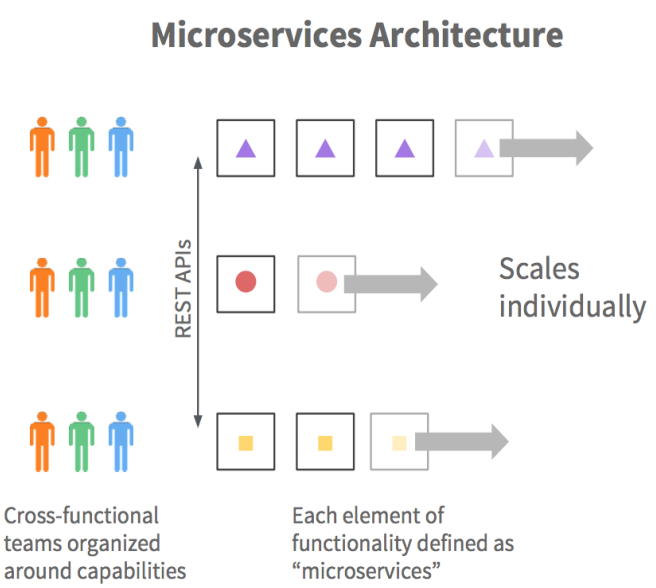
传统应用新需求

- 研发部门期待缩短从开发到产品的生命周期，更好服务公司业务降低风险；
- 激烈竞争的商业市场下，GTM期待传统应用快速演进，能够满足市场需求；
- 前端部门对终端用户的无间断，高可靠性的服务要求不断加强，提供竞争力；
- 在IT和数据中心上的费用不断攀升，期待降低运维和运营成本；
- 传统企业期待业务系统能够根据Data的分析，从而持续修正自己；

传统应用>>>现代应用



Small number of large processes with strong inter-dependencies



Cross-functional teams creating new microservices without interdependencies

生产环境验证过的伸缩性

DC/OS FEATURE

- DC/OS is built on the production-proven Apache Mesos and Marathon
- Built and refined with best practices by some of the world's top experts in distributed systems
- Tested to 10s of thousands of nodes in production for more than 6+ years in production
- Designed for resiliency and scale from day 1

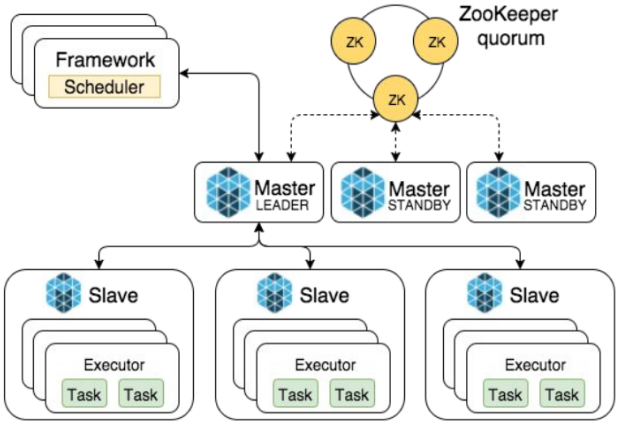
BENEFITS

- Reliable, mature technology with years of bug fixing, reduce risk of downtime for mission critical workloads
- Easily expand from small to very large clusters with confidence
- Open source core prevents vendor lock-in

现代应用分布式高可用性

DC/OS FEATURE

- No SPOF, N+1 Architecture
- Integrated KV Store, DB & load balancing for seamless failover and maintenance
- Turn-key solution, no external dependencies for HA
- Agent crash, restart & upgrade does not affect workloads
- Erlang based distributed networking stack for on the fly upgrade
- Non disruptive failover and upgrade for masters and agents



BENEFITS

- Highest uptime for scalable masters, agents and workloads
- No manual intervention in case of master/agent failover
- Easily move to the latest DC/OS release
- Resilience to network partitioning & split brain scenarios

容器和数据服务

DC/OS FEATURE

Supports Stateless and Stateful Containers

- Docker Containers
- Mesos Containers (i.e Java Binaries, Go Binaries)

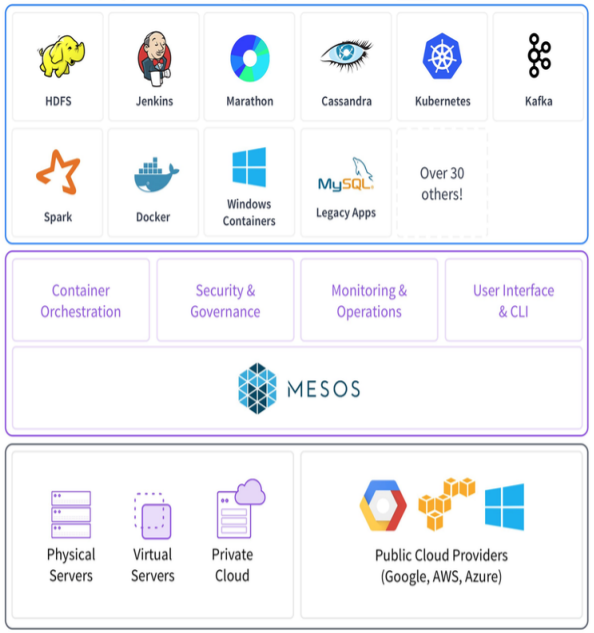
Supports Stateful Data and Analytics Services

- Analytics & Big Data (i.e Spark, HDFS), No-SQL Databases (i.e Cassandra)
- Search (i.e Elasticsearch), Message Queueing (i.e Kafka)

Linux and (soon) Windows

BENEFITS

- Increased Utilization across datacenter
- Easily scale applications up and down
- Single platform for all applications, simplifying operations



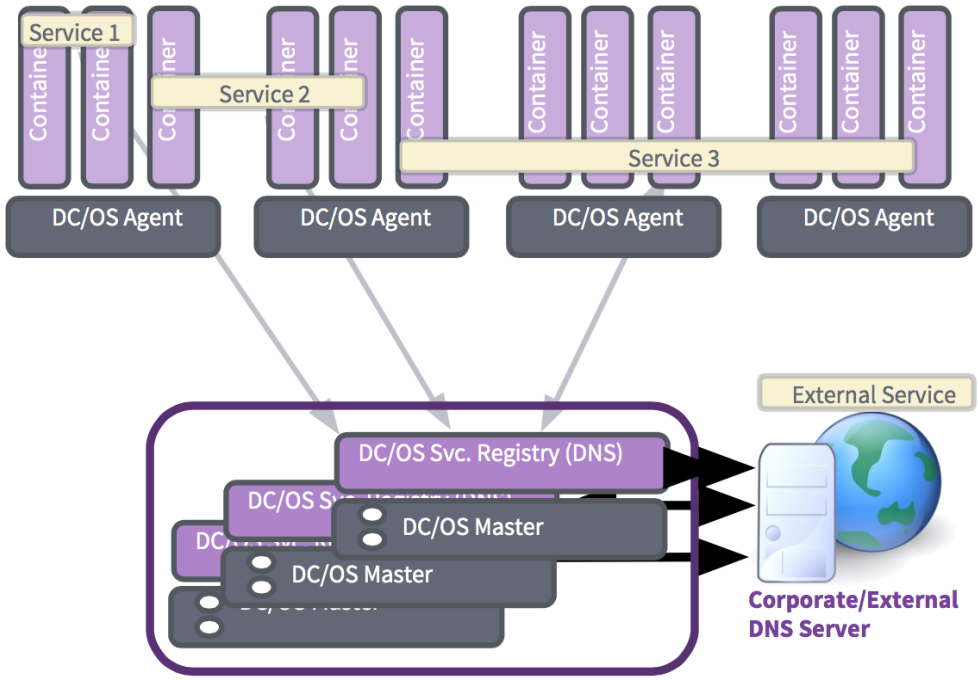
服务发现

DC/OS FEATURE

- DNS-based Service Registry & Distr. DNS Proxy
- Integrated DC/OS Service Discovery
- Integrated External Service Discovery

BENEFITS

- Highly available & scalable service registry
- Turn-key solution to deploy micro-services
- Integration with existing non-DC/OS services to enable brownfield deployments.



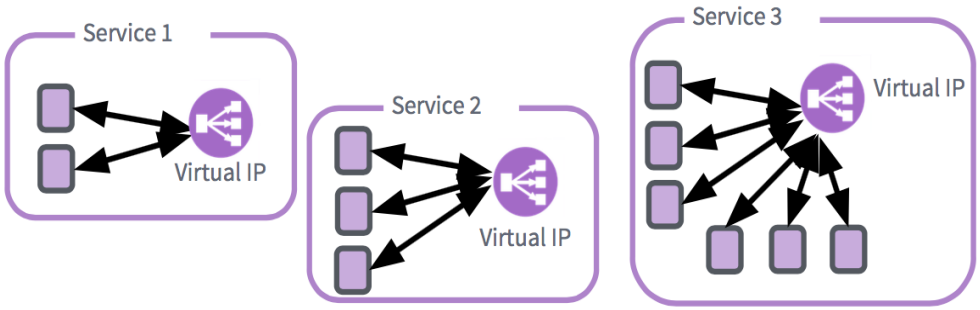
可靠性 - L4 LB

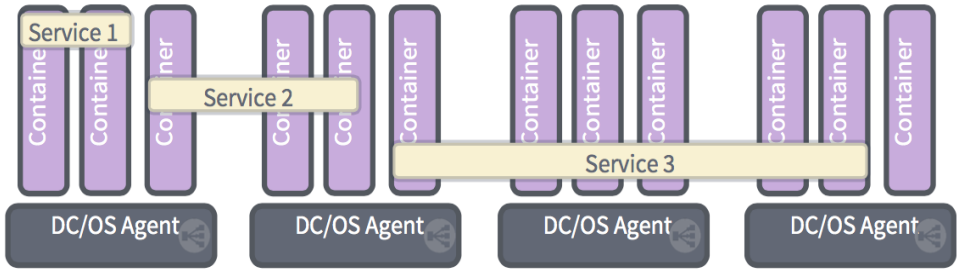
DC/OS FEATURE

- Fast Converging Distributed Load Balancer
- Integrated with DC/OS Service Discovery
- Integrated Blue-Green deployment
- Support for variety of Layer 4 LB algorithms

BENEFITS

- Highly available LB with no single choke point
- Enable non-disruptive service upgrades with blue-green deployments.
- Highly scalable and tolerant to large # of host failures.

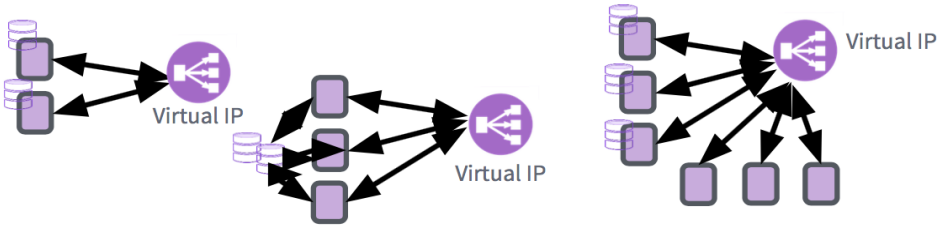




有状态容器和卷

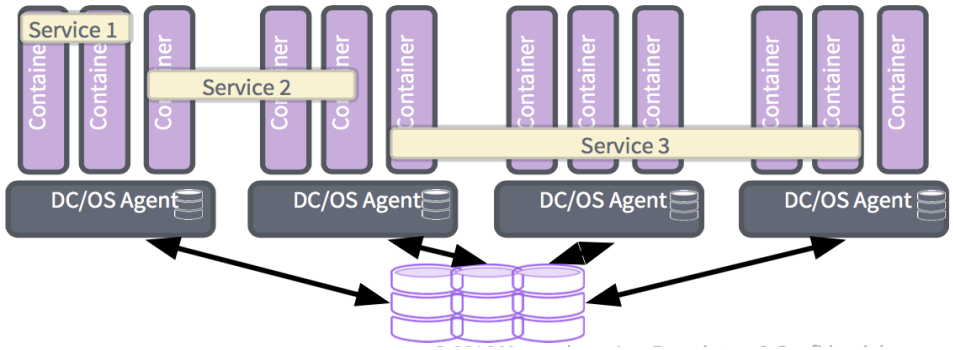
DC/OS FEATURE

- Persistent Local Volumes for Containers
- External Volumes using Shared Storage
- DC/OS Volume Management UI

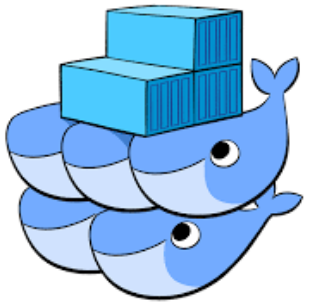


BENEFITS

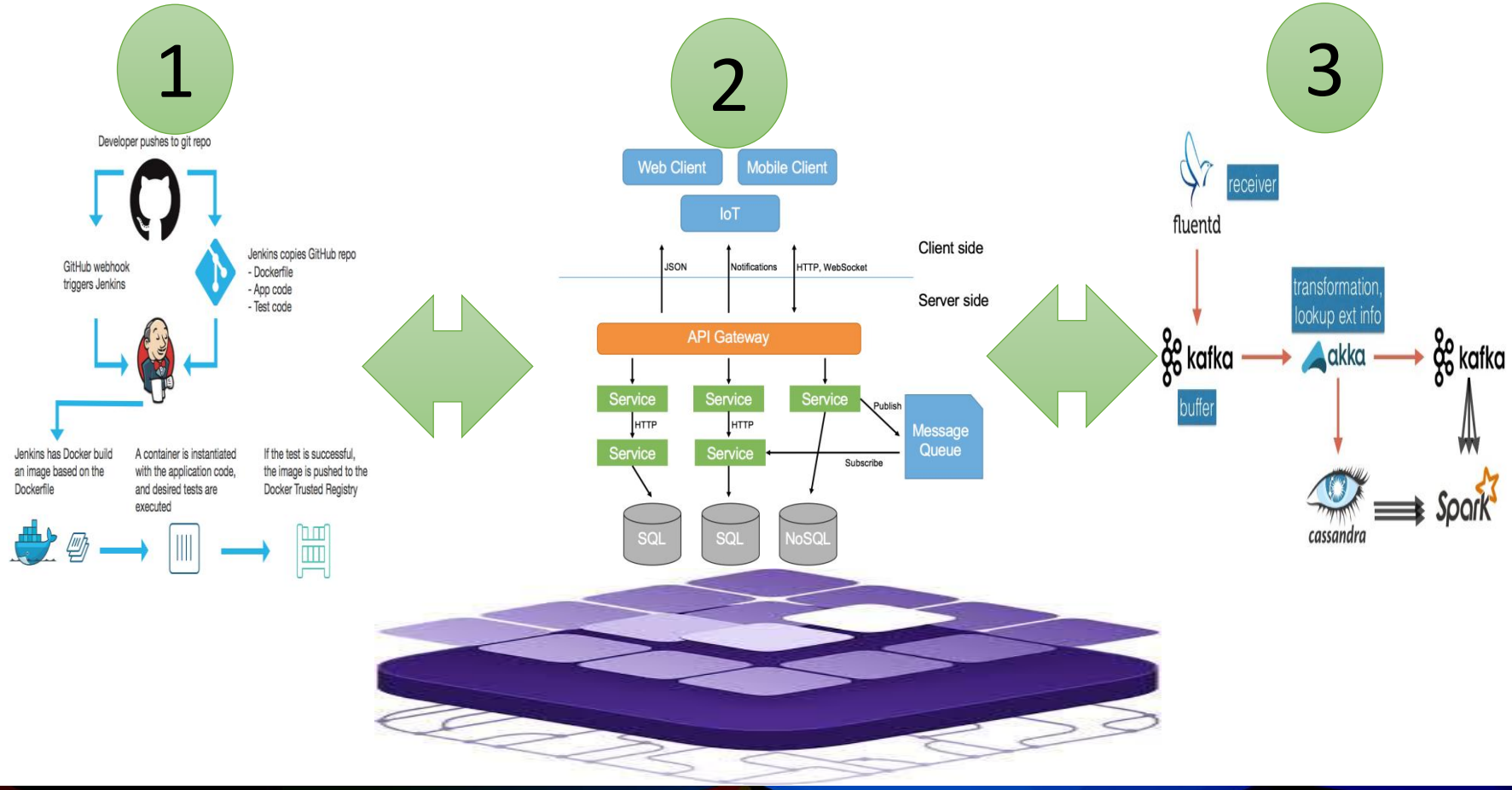
- Run Stateful applications like MySQL in Containers.
- Integrate with existing shared storage
- Manage & Troubleshoot with integrated volume manager



“三国杀” + ？



现代架构最佳实践



案例分享一

Challenges

- 400 developers submitting jobs to Jenkins for CI/CD builds experienced sizable delays in task completion due to Jenkins job queueing

Mesosphere Solution

- Customer wanted to replicate the eBay use case[1] of running CI builds for eBay's applications in Mesos with Docker containers
- Mesosphere DC/OS allowed Customer to move from an enterprise Continuous Integration solution to open source as Marathon provides equivalent HA functionality

- One of North America's leading diversified financial services companies
- Provides banking, wealth management, insurance and capital markets services on a global basis

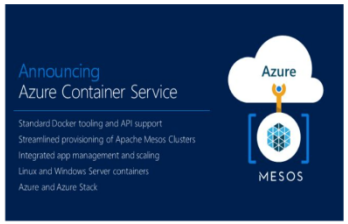
Challenges

- Needed a production grade native container service that would work on premises and on azure, at massive scale
- Must easily integrate with Azure CI/CD, app management and auto scaling infrastructure
- Microsoft and Linux friendly technology

Mesosphere Solution

- After independent evaluation, MS team determined Mesos/Mesosphere was the right fit
- Currently integrating Mesosphere DC/OS as a core technology for Azure Container Service

Microsoft



Challenges

- Two Sigma OPS struggled with developer demands for agile real-time analysis
- Already explored various IaaS & PaaS solutions

Mesosphere Solution

- After successful consulting services engagement determined Mesosphere was the right fit
- Compelling reason to move fast; Agility/Performance/Scalability
- 100s of servers moving to 1000s in next 6 months

2σ

- NYC-based Hedge Fund
- $25 Billion AUM
- Uses a variety of technological methods for its trading strategies:
  - Artificial Intelligence
  - Machine Learning
  - Distributed Systems

Challenges

- Verizon needed infrastructure that could handle the volume and speed of data that its users generate across video services and mobile apps
- Verizon was seeking to improve automation, scalability and efficiency when deploying applications, services and big data infrastructure

Mesosphere Solution

- Mesosphere DC/OS allowed Verizon to quickly launch new products and services while reducing the IT requirements in their datacenters
- Chose Mesosphere DC/OS for hybrid cloud capabilities, to move from AWS to Verizon's private datacenter

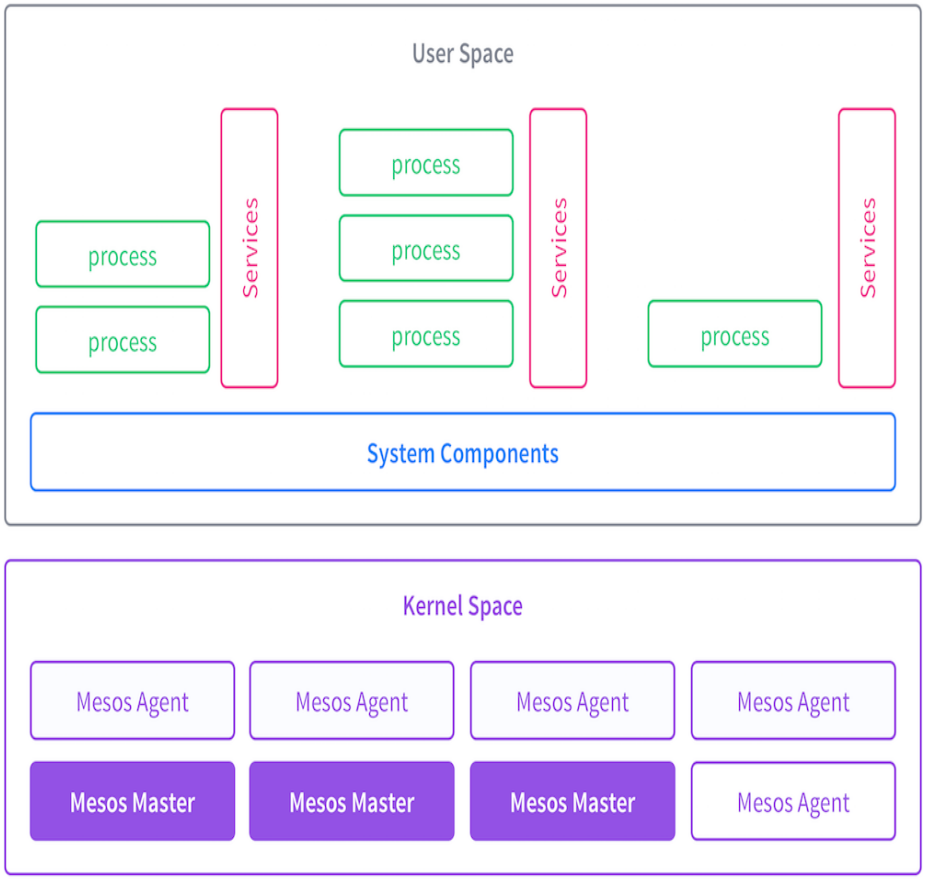
verizon

Larry Rau from @Verizon with @flo Launching 50,000 containers in seconds with @mesosphere #DC/OS

案例分享二

- 电信运营商
- 银行
- 互联网公司
- 保险
- 车联网

DC/OS服务集



System Components are installed and are running by default in the DC/OS cluster and include the following:

The **Admin Router** is an open source NGINX configuration that provides central authentication and proxy to DC/OS services.

Exhibitor automatically configures ZooKeeper during installation and provides a usable Web UI to ZooKeeper.

Mesos-DNS provides service discovery, allowing apps and services to find each other by using the domain name system (DNS).

Minuteman is the internal layer 4 load balancer.

Distributed DNS Proxy is the internal DNS dispatcher.

DC/OS Marathon, the native Marathon instance that is the 'init system' for DC/OS, starts and monitors DC/OS services.

ZooKeeper, a high-performance coordination service that manages the DC/OS services.

The **Cosmos service** is our internal packaging API service.

The diagnostics service (also known as **3DT** or dcos-ddt.service, no relationship to the pesticide!) is our diagnostics utility for DC/OS systemd components.

Logrotate service does what you think it does: ensures DC/OS services don't blow up cluster hosts with too much log data on disk.

Marathon shouldn't need any introduction, it's the distributed init system for the DC/OS cluster.

The DC/OS **signal service** queries the diagnostics service /system/health/v1/report endpoint on the leading master and sends this data to SegmentIO for use in tracking metrics and customer support.

The **history service** provides a simple service for storing stateful information about your DC/OS cluster.

Join the DC/OS Community

Connect with our community of users and browse the latest DC/OS news.

GitHub

Are you interested in helping us make DC/OS even better? Let's work together! Check out our source code on GitHub.

View repositories →

Slack

Have any questions? Our Slack channel is the best place to get help. Just send us a request to automatically receive your invitation.

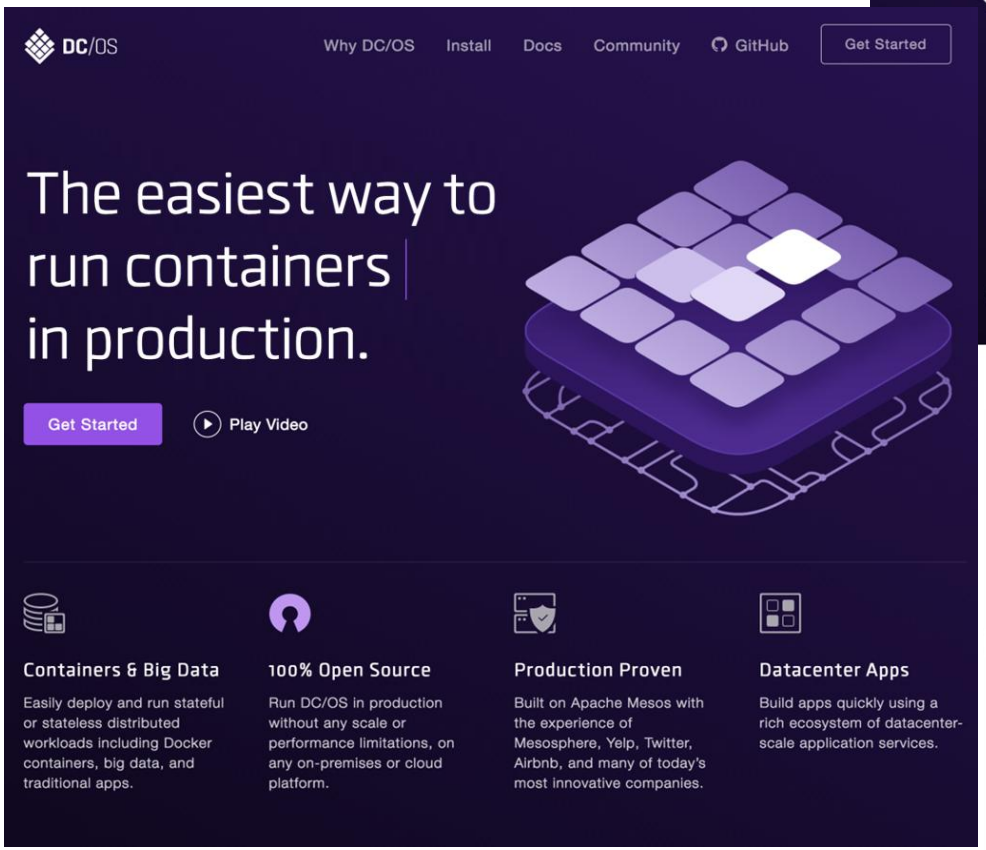
Join chat →

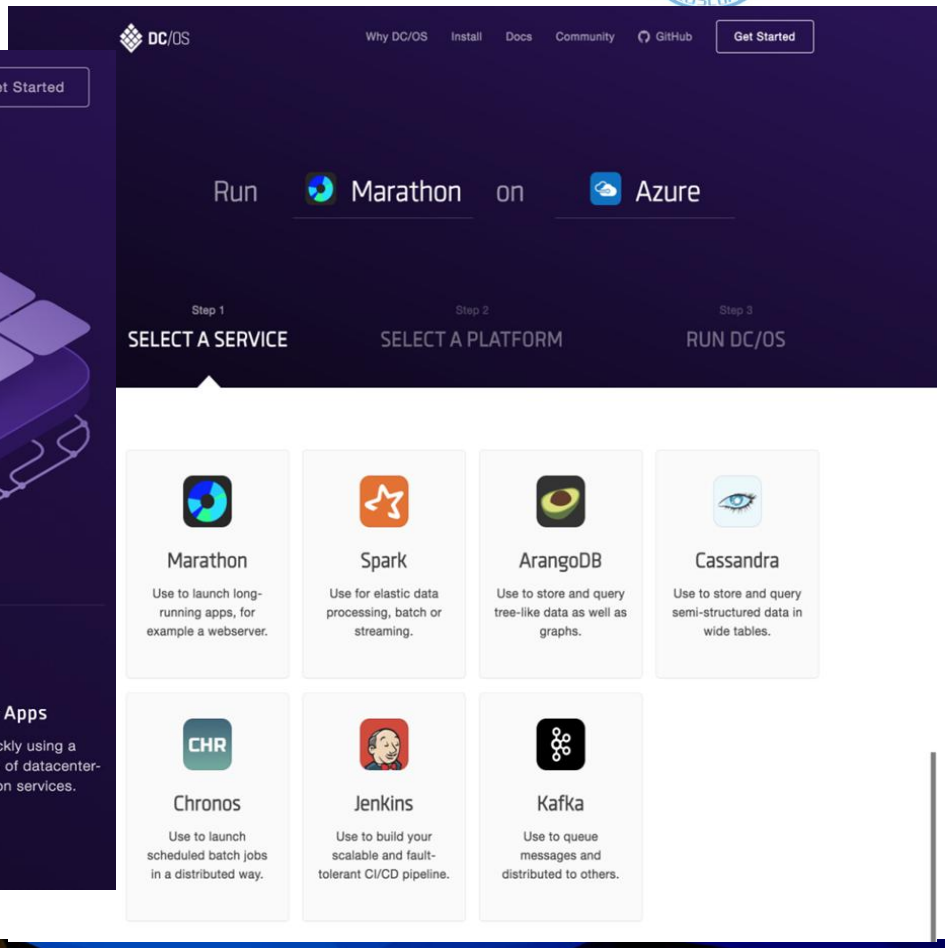
Mailing List

Want to stay in the loop and connect with other community members? Our public mailing list has all the latest updates. Join the discussion.

Join users@dcos.io →

DCOS.io





开源社
kaiyuanshe

谢谢

Q&A

开源社
kaiyuanshe