

从Kubernetes到Cloud Native 云原生应用之路

宋净超(jimmysong)

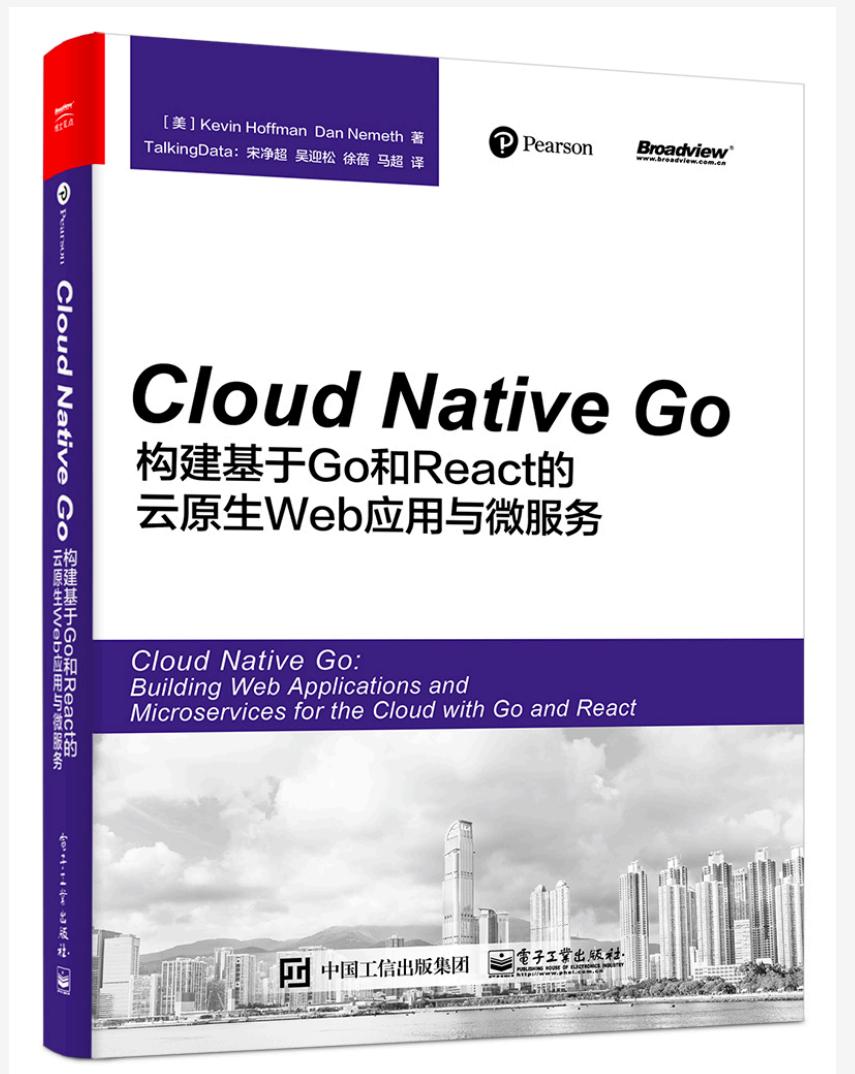
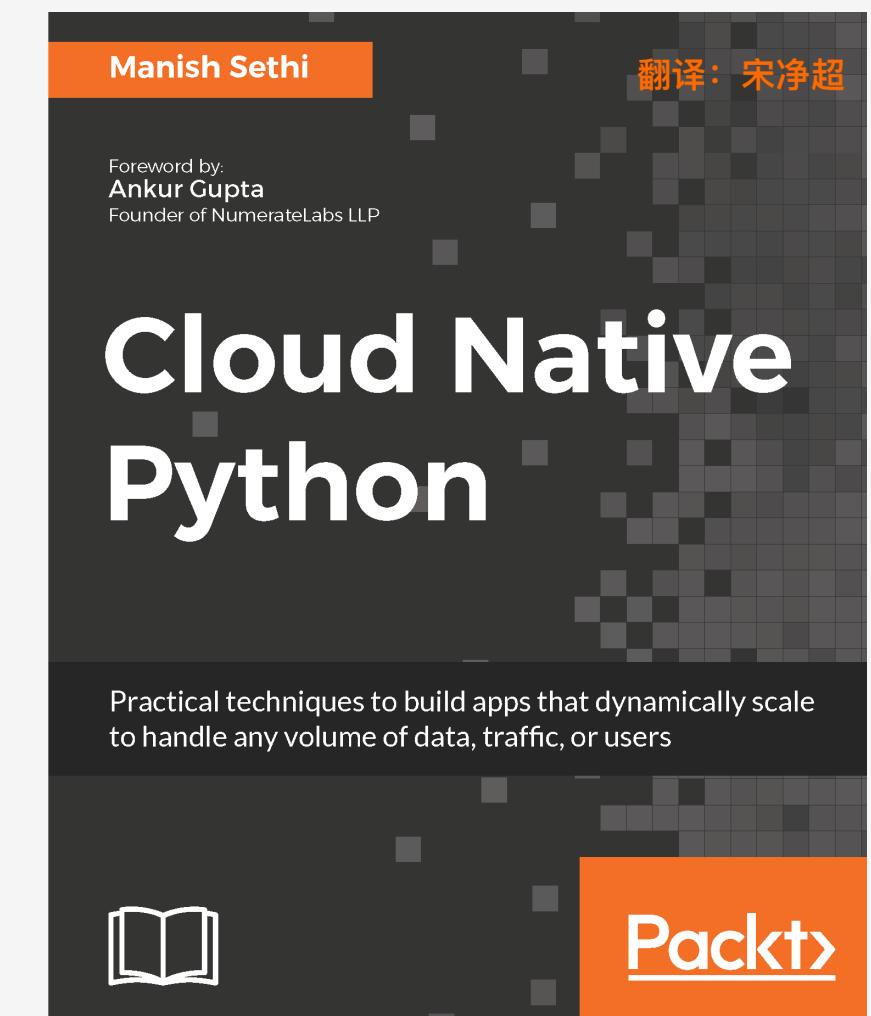
TalkingData



关于我

TalkingData容器平台负责人

- 关注大数据与容器平台、微服务、Kubernetes与云原生应用
- 《Cloud Native Go - 构建基于React和Go的云原生应用和微服务》译者
- 《Cloud Native Python - 使用Python和React构建云原生应用》译者
- Kubernetes中文社区/k8smeetup
- 个人博客 <https://jimmysong.io>



Outline

- **Docker in TalkingData**
- **Containers**
- **Why Kubernetes?**
- **Microservices**
- **Cloud Native**
- **Service Mesh**
- **Use Cases**
- **Open Source**

Docker in TalkingData

2015-2016

Yarn on Docker CLI tool



<https://github.com/rootsongjc/magpie>

Web UI



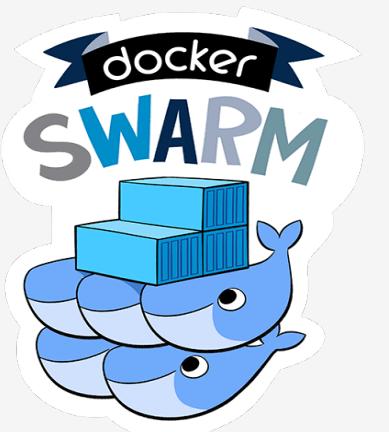
<https://github.com/shipyard/shipyard>

Docker network plugin



<https://github.com/talkingdata/shrike>

Container orchestrating



<https://github.com/docker/swarm>

CI/CD



<https://jenkins.io/>

- Yarn on Docker
- About 1k containers
- 10TB+ memory

2017-

Client-go



<https://github.com/kubernetes/client-go>

Kubernetes dashboard



<https://github.com/kubernetes/dashboard>

Flannel



<https://github.com/coreos/flannel>

Kubernetes



<https://kubernetes.io/>

CI/CD



Jenkins

Yarn on Docker

- Hadoop Yarn
- Swarm
- Docker1.11
- 容器固定IP
- 环境变量
- 类虚拟机
- 命令行管理、UI辅助

The screenshot shows the Shipyard UI interface. The top navigation bar includes tabs for shipyard, CONTAINERS (which is selected), IMAGES, NODES, REGISTRIES, ACCOUNTS, and EVENTS. On the far right, there are ADMIN and HELP buttons. Below the navigation is a search bar containing the text "yarn1". A toolbar below the search bar features a Refresh button and a Deploy Container button. The main area is a table listing eight Docker containers. The columns are: ID, Node, Name, Image, Status, Created, and Actions. Each row contains a checkbox, a green heart icon, the container ID, the node it's running on (redacted), the container name (e.g., yarn1add-nm145), the image name (e.g., yarn1add-nm145), its status (Up 37 hours), the creation date (2016-08-12 11:39:56 +0800), and two small action icons (search and edit).

ID	Node	Name	Image	Status	Created	Actions
374c48312654	[REDACTED]	yarn1add-nm145	[REDACTED]	Up 37 hours	2016-08-12 11:39:56 +0800	
7533fcb3c39d	[REDACTED]	yarn1add-nm146	[REDACTED]	Up 37 hours	2016-08-12 11:39:55 +0800	
c51af21fe7da	[REDACTED]	yarn1add-nm143	[REDACTED]	Up 2 days	2016-08-11 16:14:30 +0800	
015ea594c60f	[REDACTED]	yarn1add-nm144	[REDACTED]	Up 2 days	2016-08-11 16:14:30 +0800	
ba312670b622	[REDACTED]	yarn1add-nm141	[REDACTED]	Up 2 days	2016-08-11 14:09:42 +0800	
2790d5494bb6	[REDACTED]	yarn1add-nm142	[REDACTED]	Up 2 days	2016-08-11 14:09:41 +0800	
1b1e2bdecd38	[REDACTED]	yarn1add-nm137	[REDACTED]	Up 3 days	2016-08-10 17:08:10 +0800	

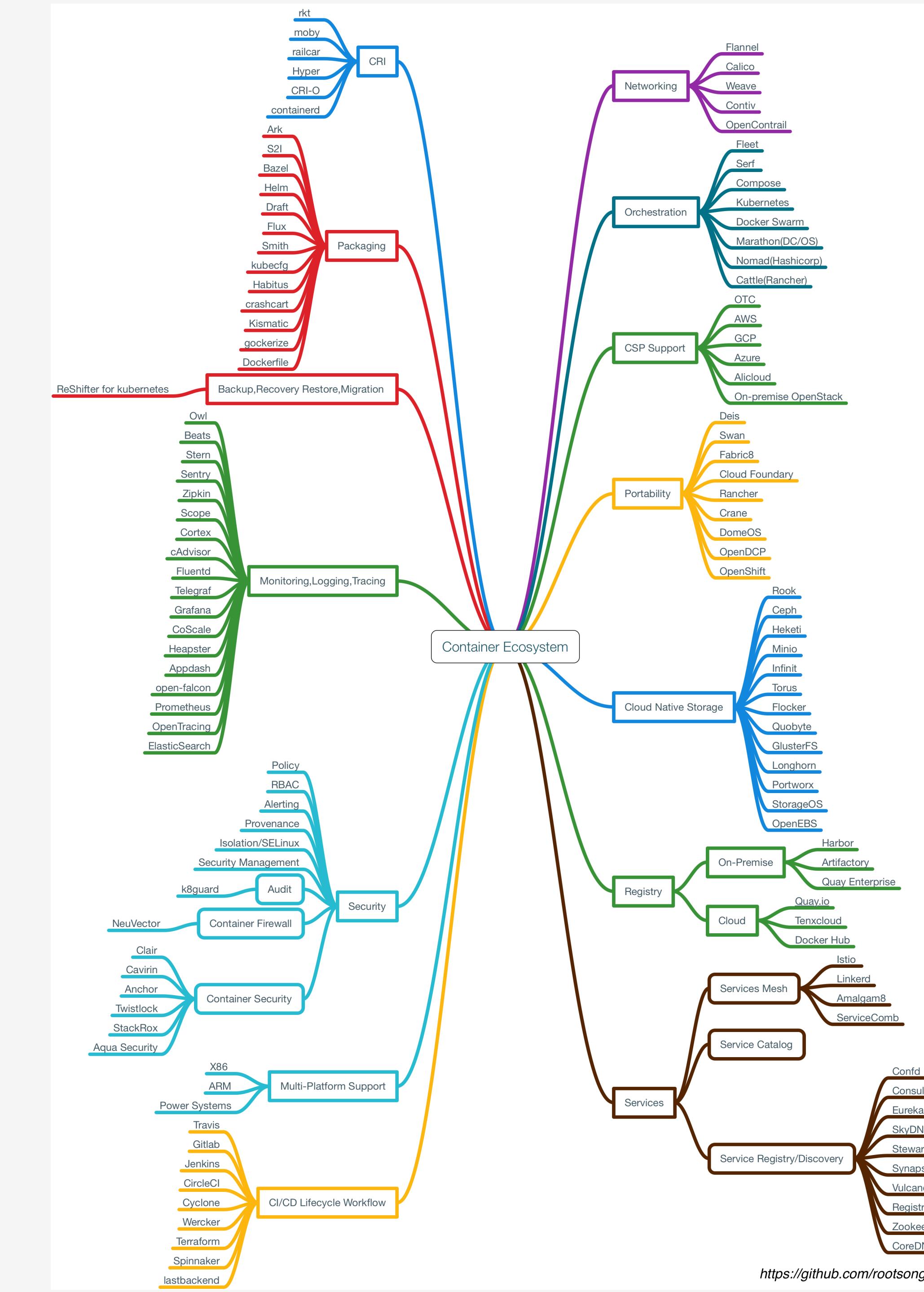
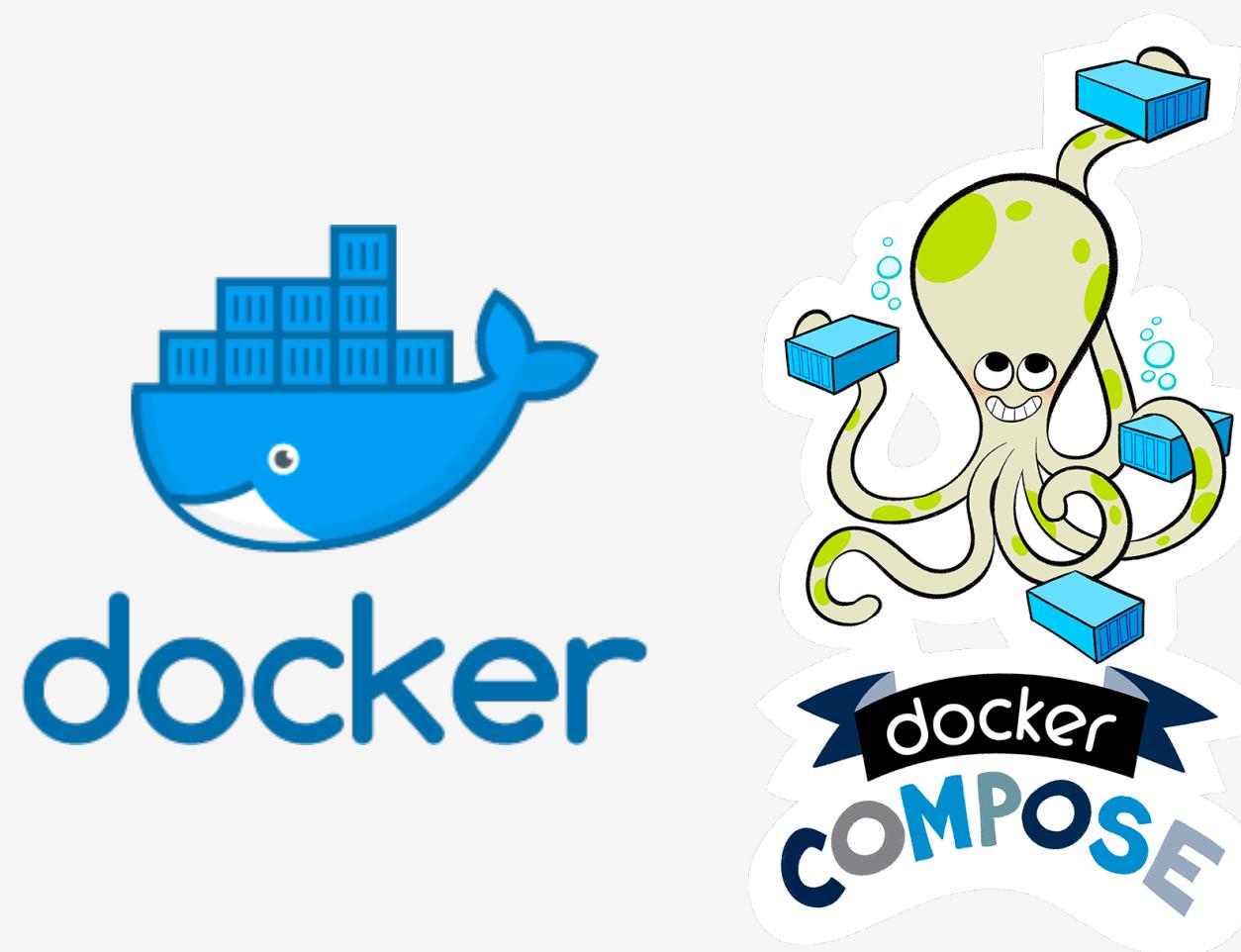
Outline

- Docker in TalkingData
- Containers
- Why Kubernetes?
- Microservices
- Cloud Native
- Service Mesh
- Use Cases
- Open Source

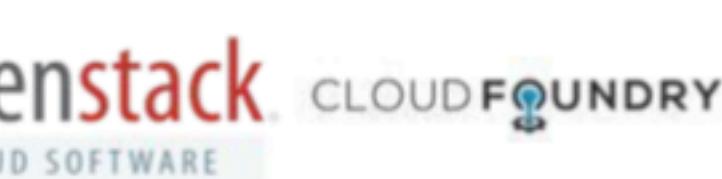
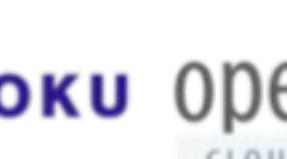
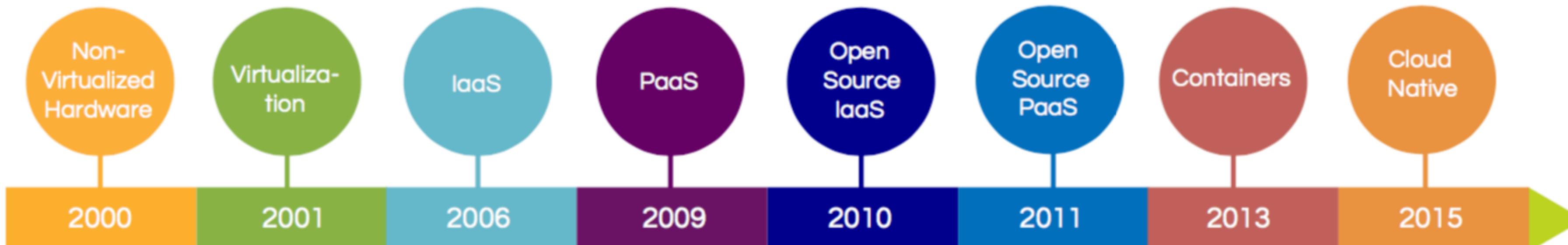


Containers

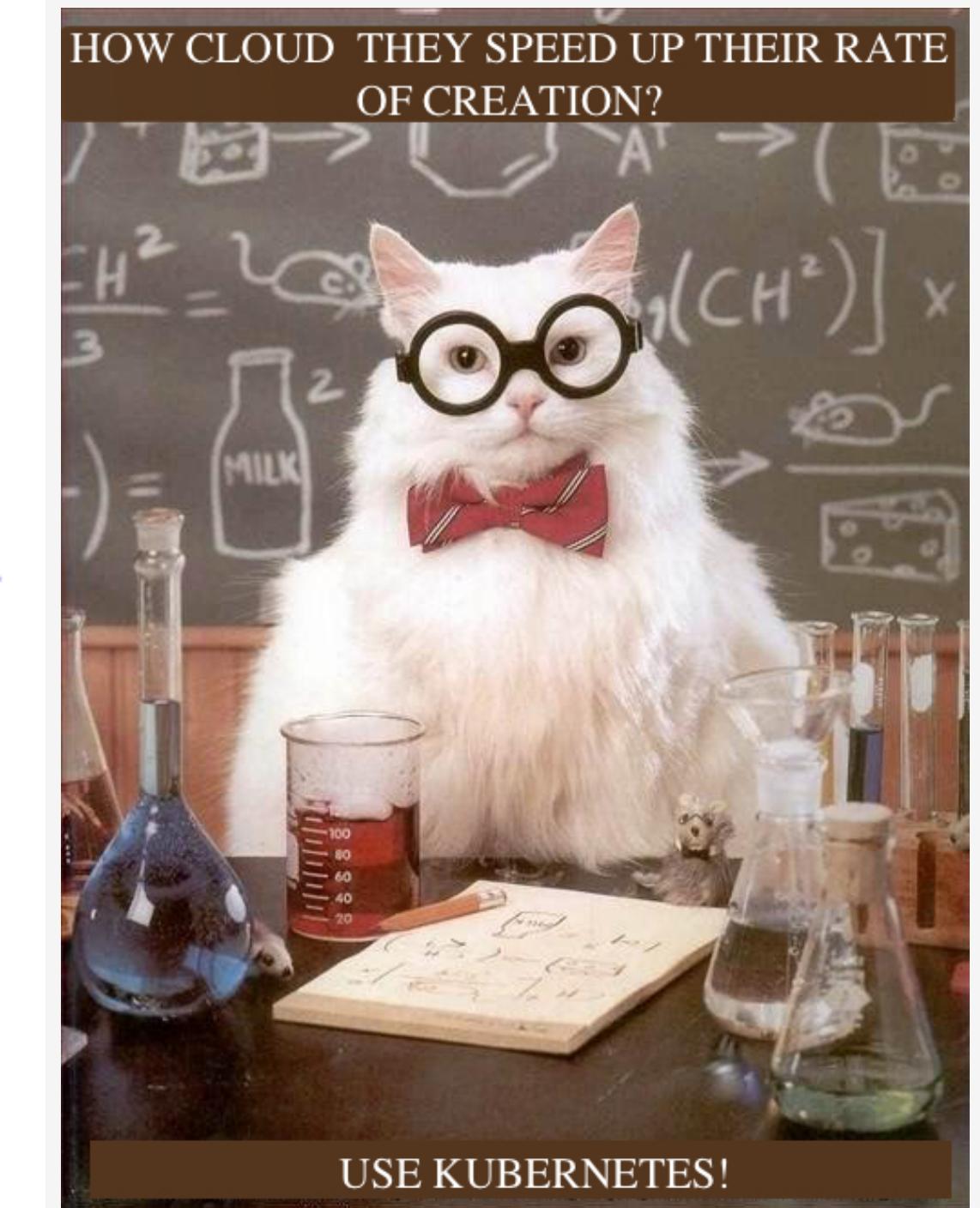
- Developer tools
- Containers 生态系统
- Productivity platform



Cloud Computing Stages



- Cloud native computing uses an open source software stack to:
 - deploy applications as *microservices*,
 - packaging each part into its own *container*
 - and dynamically *orchestrating* those containers to optimize resource utilization
- Standardization: <https://www.opencontainers.org/>



Outline

- Docker in TalkingData
- Containers
- Why Kubernetes?
- Microservices
- Cloud Native
- Service Mesh
- Use Cases
- Open Source



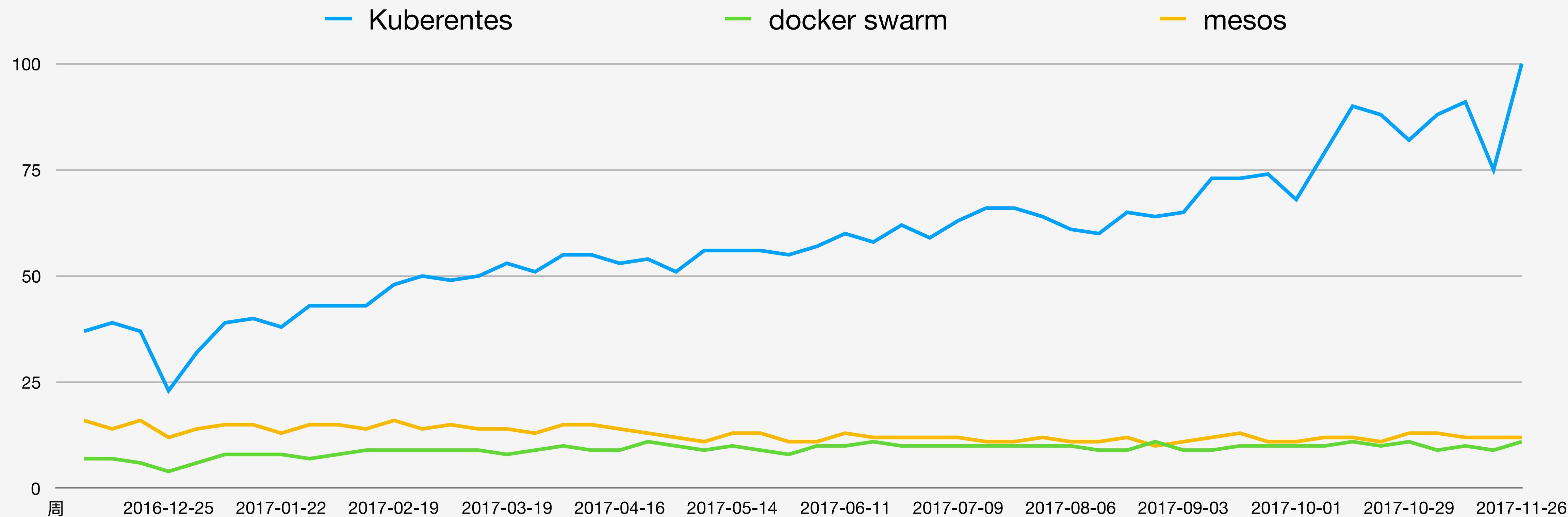
Why kubernetes?

- 优秀的容器副本控制
- 原来虚拟机式的使用方式不够灵活
- 抽象程度高，编排组合强大
- 社区活跃
- 微服务支持

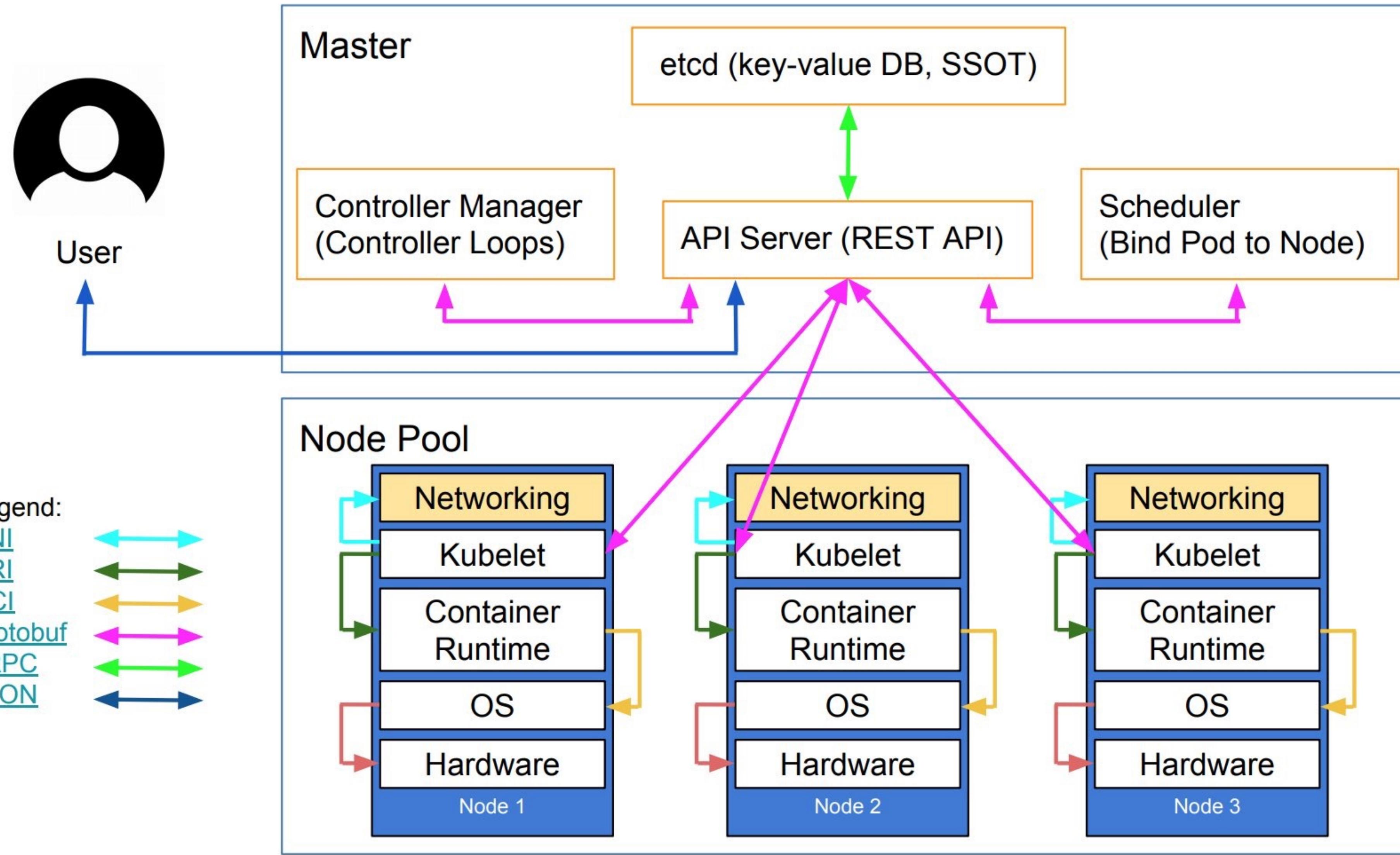


Kubernetes

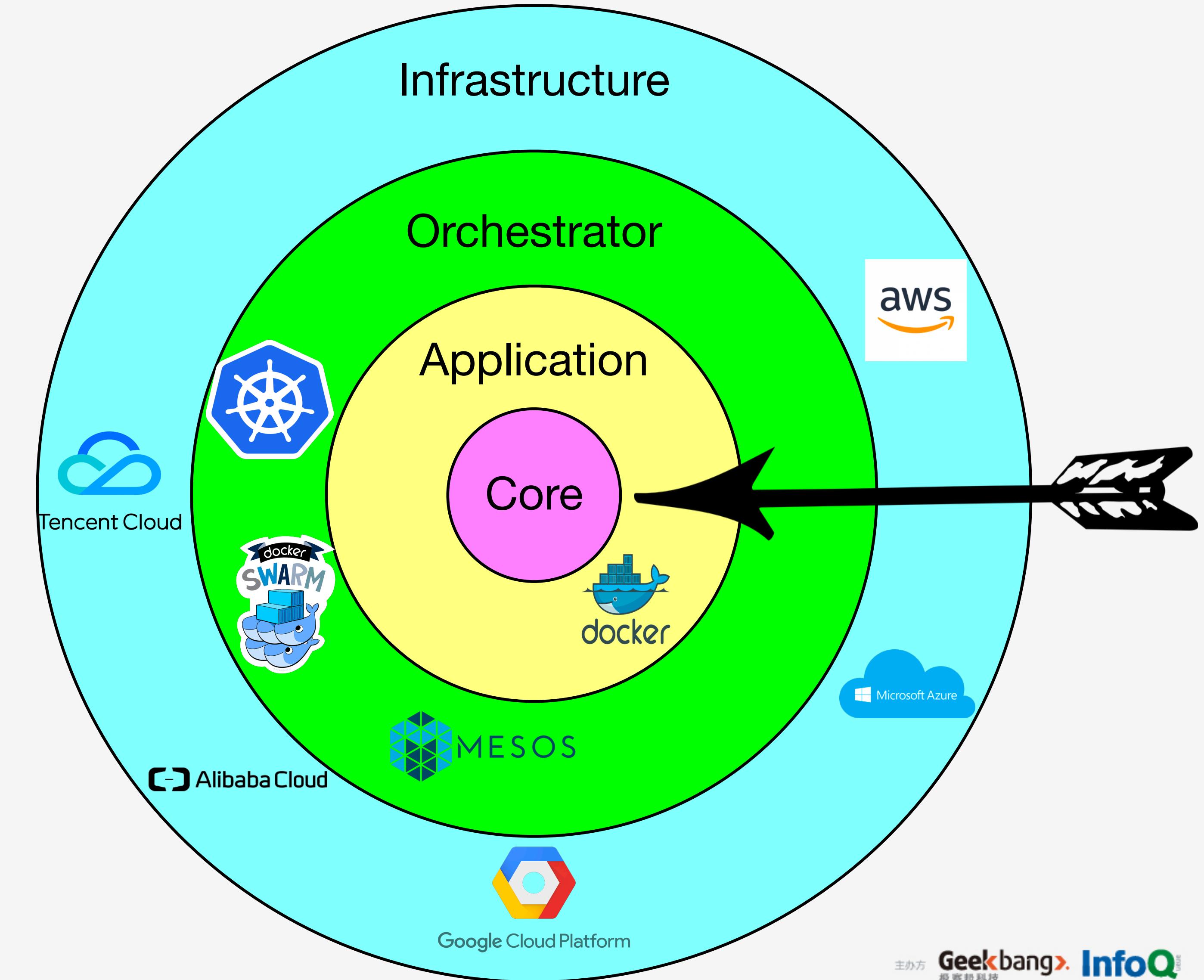
- Kubernetes the de facto container orchestrator
- more than orchestrator...



Kubernetes' high-level component architecture



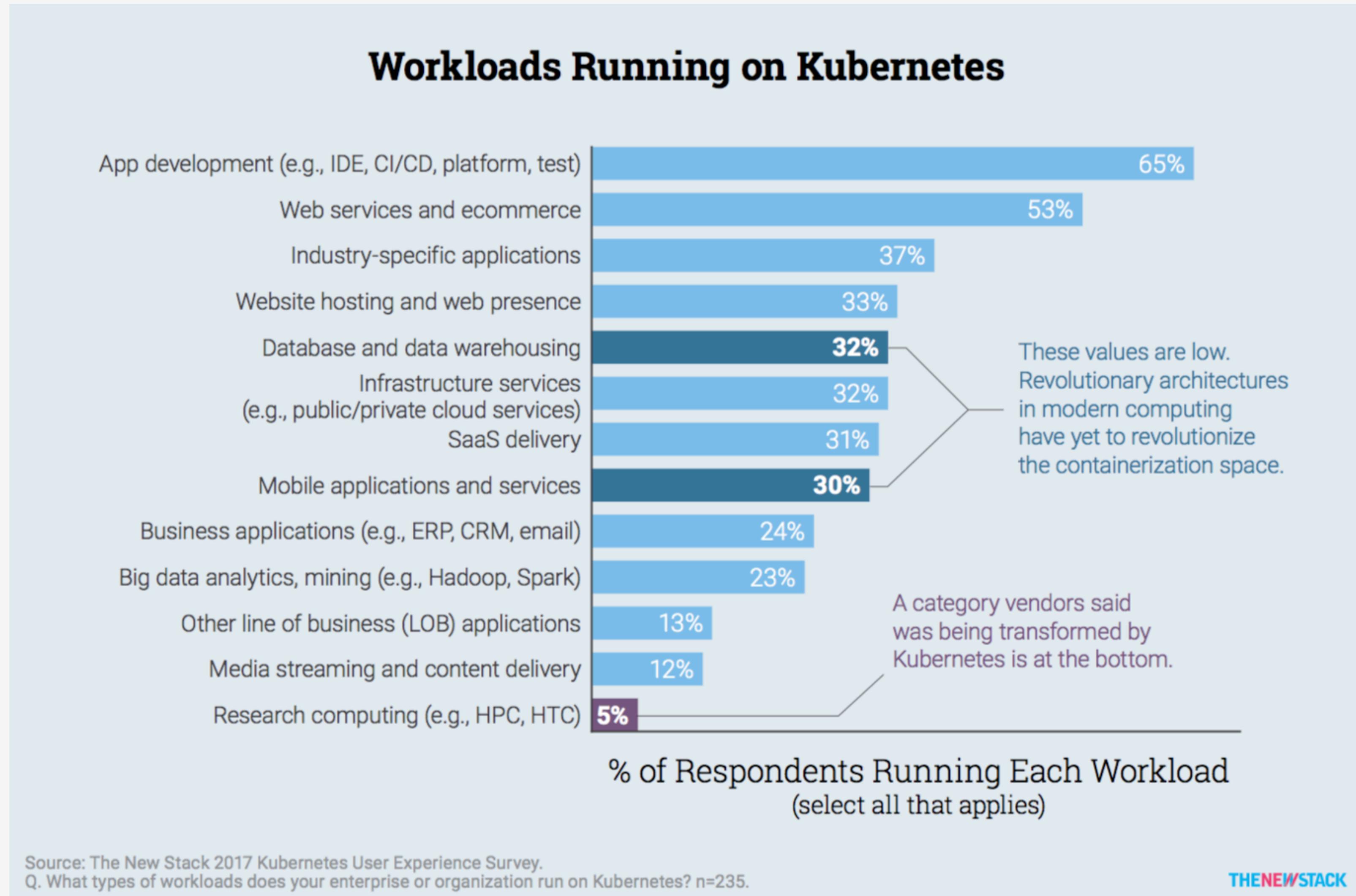
Infrastructure Portability



Serverless/FaaS



Current Use Cases

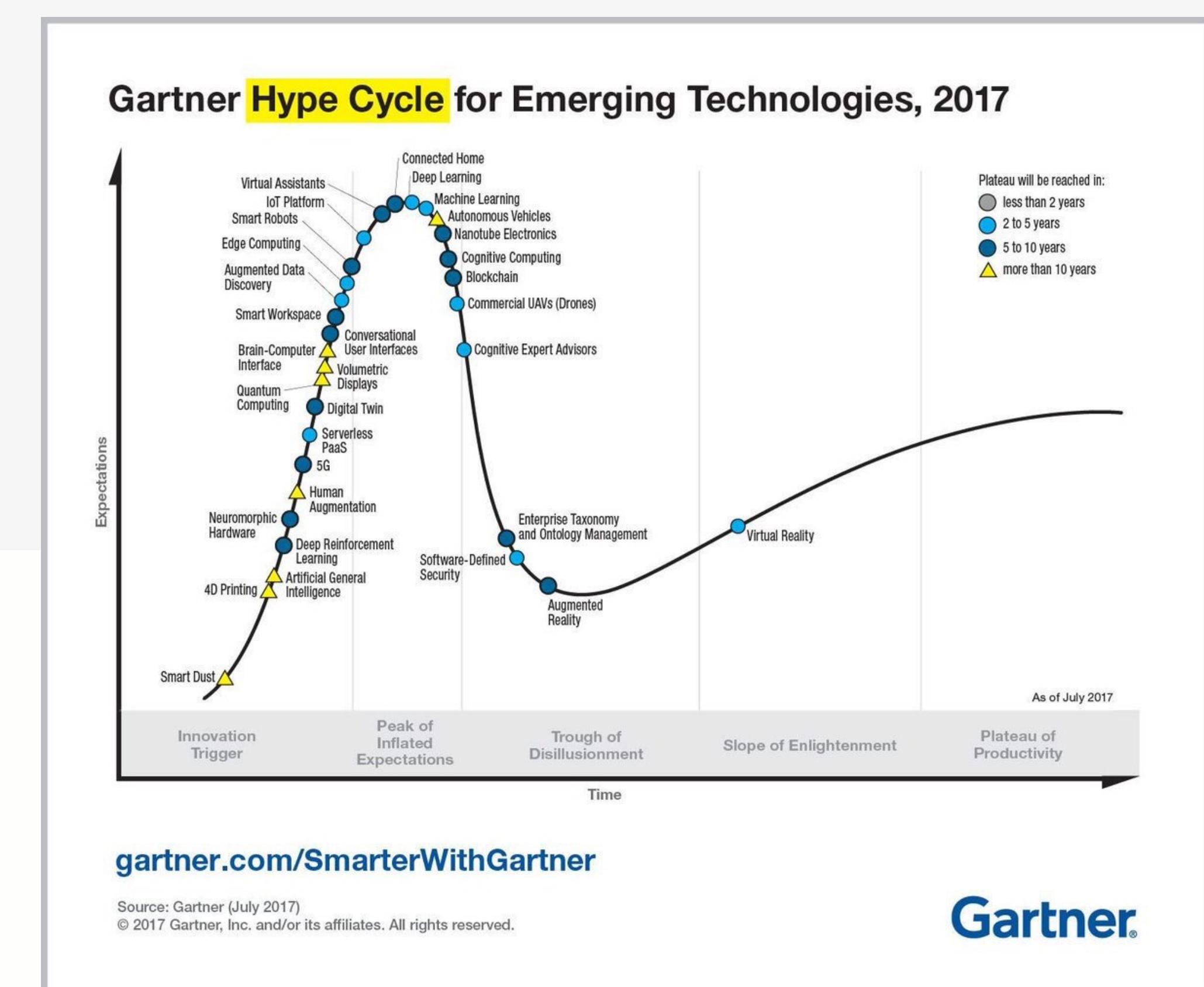
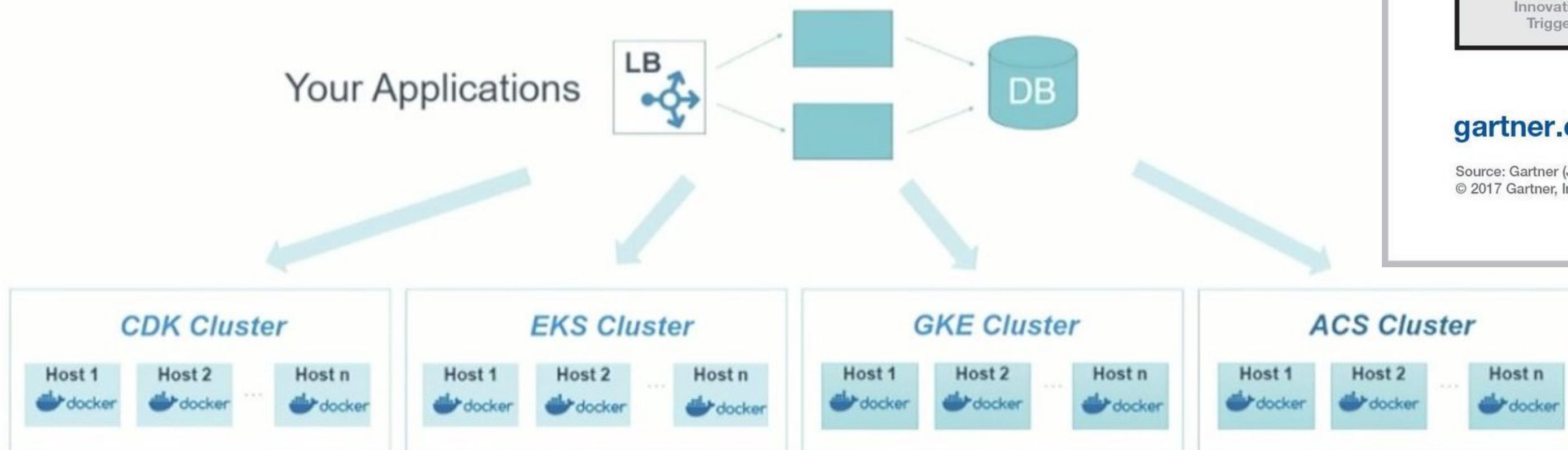


- App development
- CI/CD
- Web services

Technology disruption

Google/Microsoft/Amazon/Rancher
Canonical...

Today, Kubernetes is everywhere



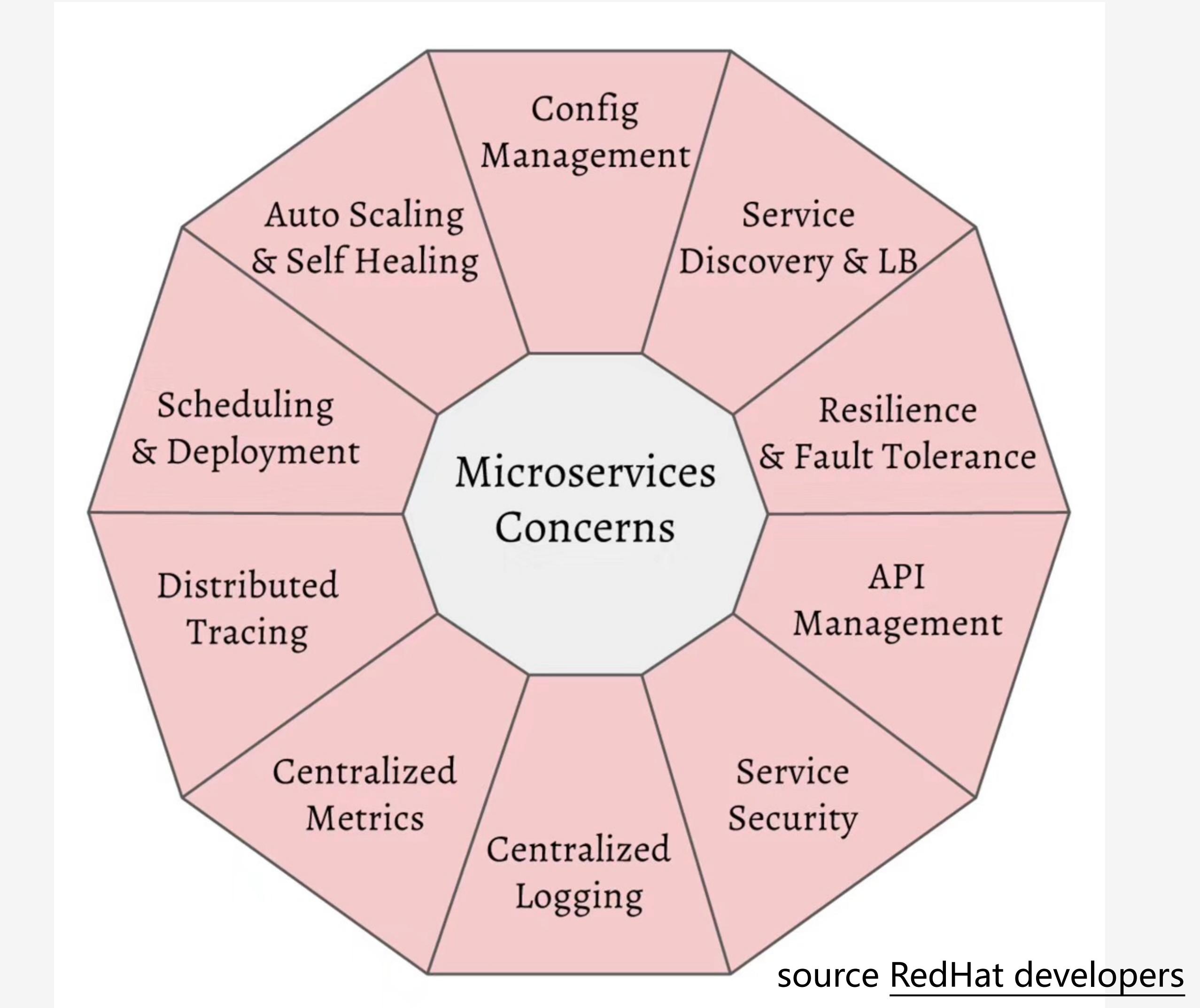
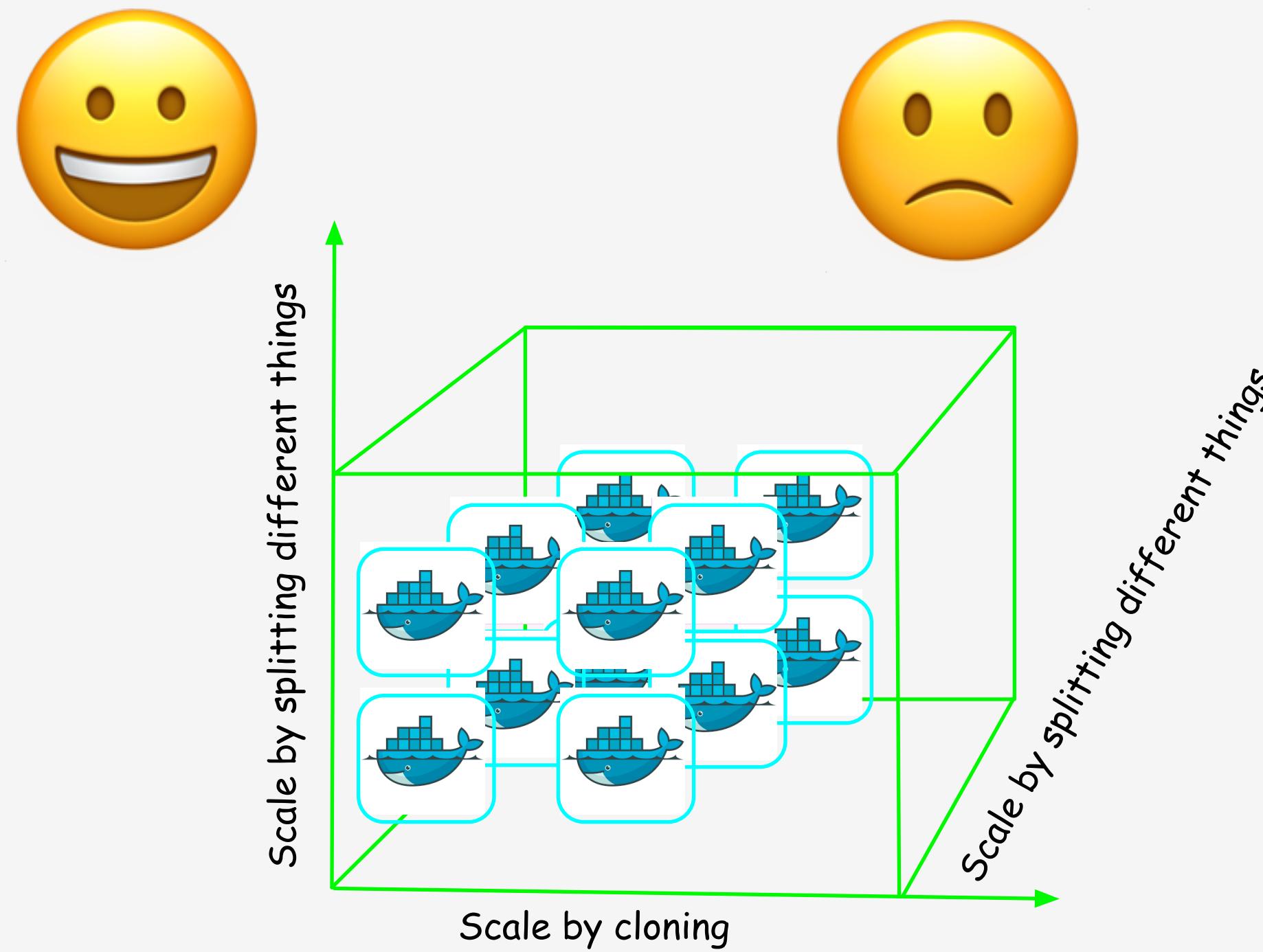
Outline

- Docker in TalkingData
- Containers
- Why Kubernetes?
- Microservices
- Cloud Native
- Service Mesh
- Use Cases
- Open Source

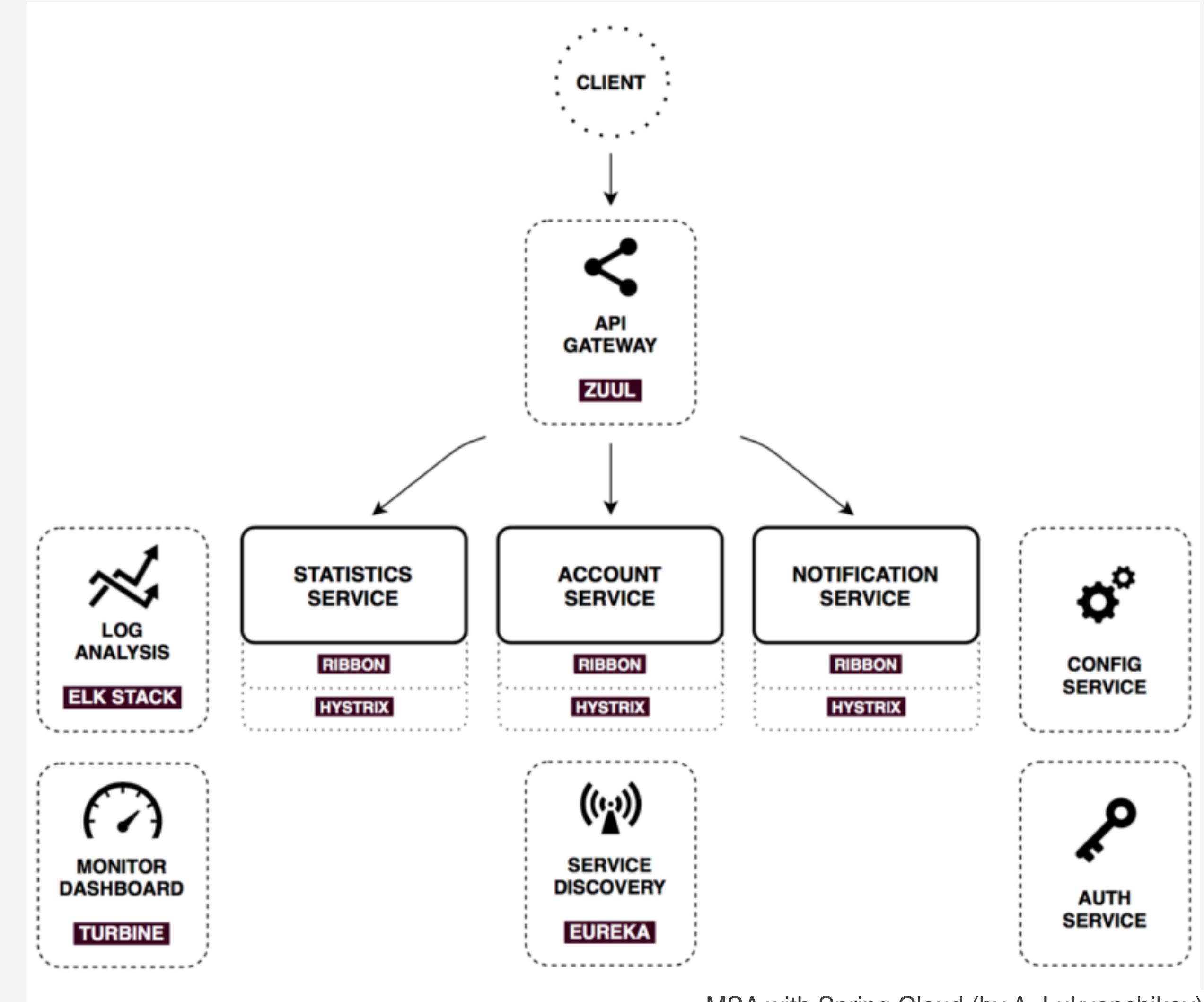
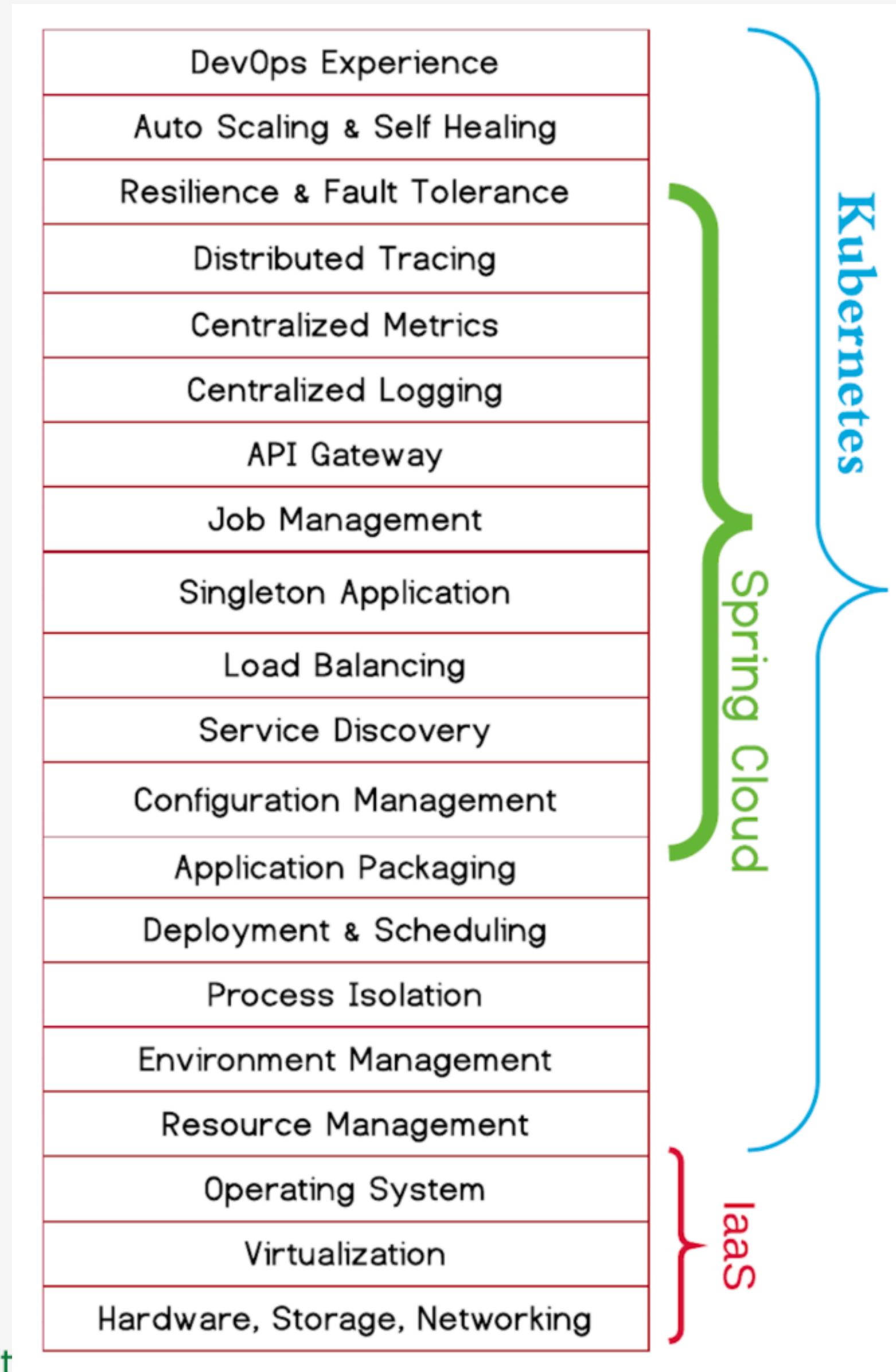
Microservices with Kubernetes

Pros & Cons

- 小团队快速开发与维护
- 技术多样性
- 独立演进
- 灵活部署与扩展
- 增加调用开销
- 分布式系统
- 调试困难
- 服务治理



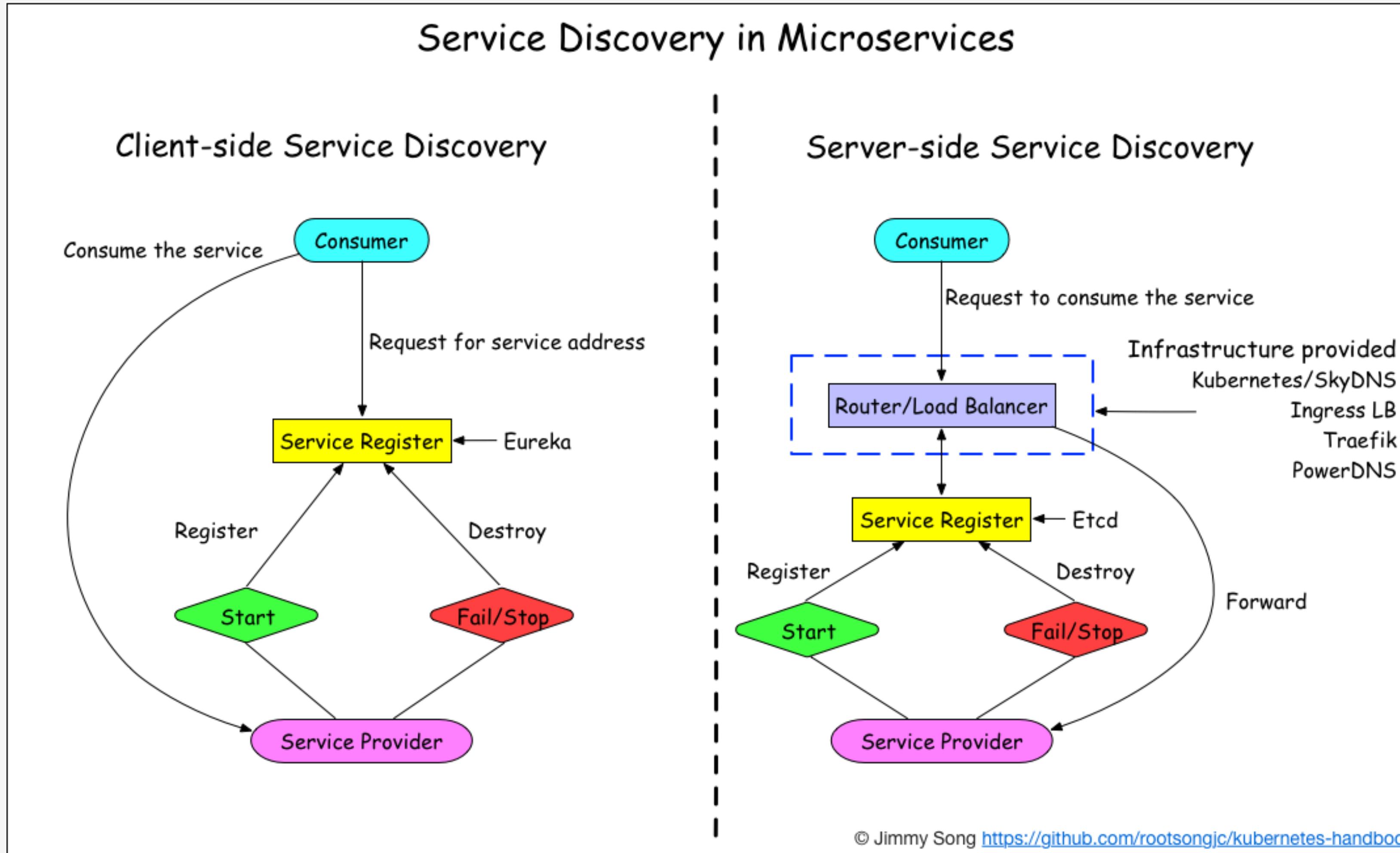
Kubernetes+Service Mesh vs Spring Cloud



MSA with Spring Cloud (by A. Lukyanchikov)

source [RedHat developers](#)

Service Discovery in Microservices



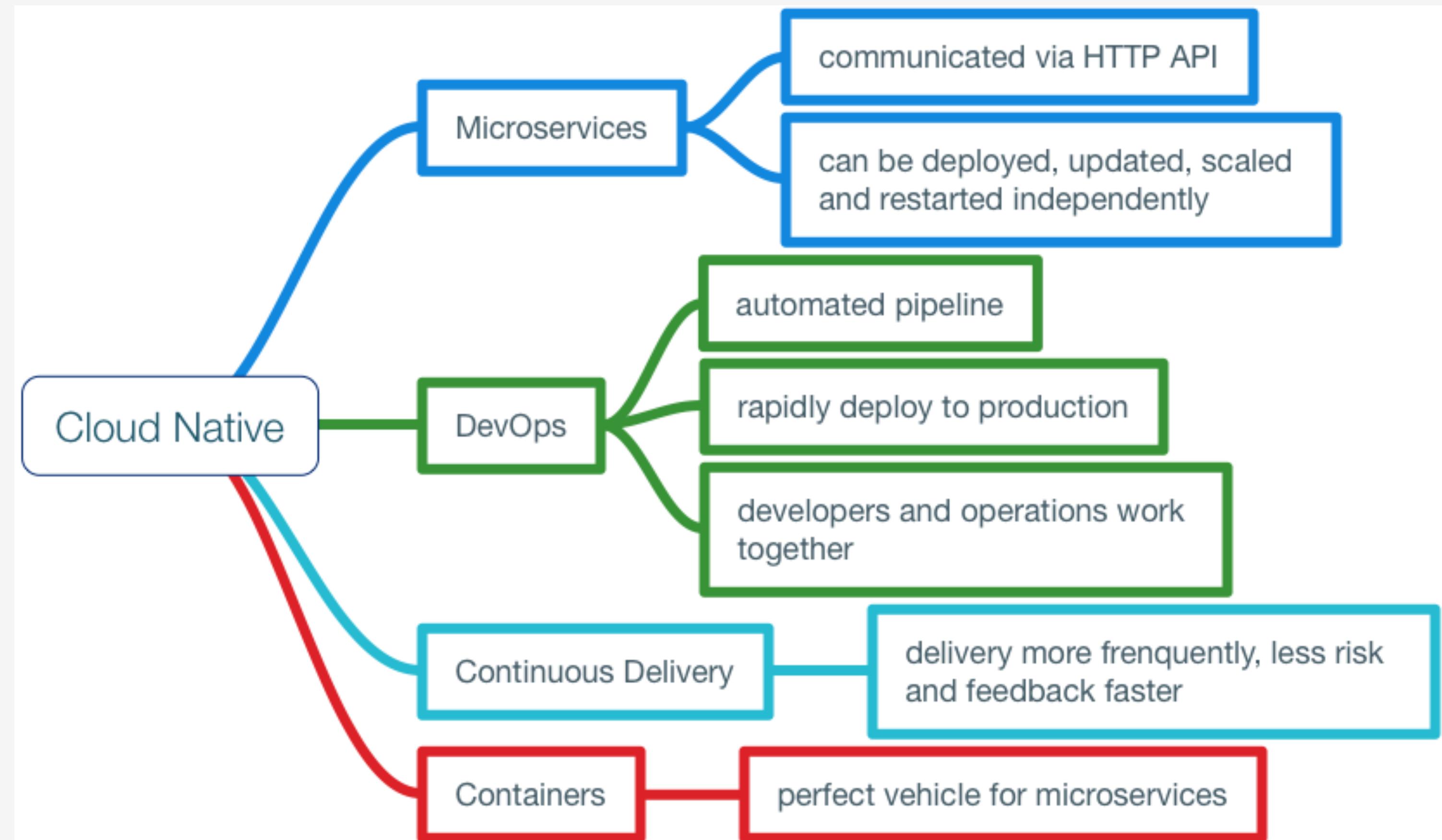
Outline

- Docker in TalkingData
- Containers
- Why Kubernetes?
- Microservices
- Cloud Native
- Service Mesh
- Use Cases
- Open Source



Cloud Native

- 容器生态
- DevOps
- 持续交付
- 微服务



Cloud Native Features

Container packaged

Running applications and processes in software containers as an isolated unit of application deployment, and as a mechanism to achieve high levels of resource isolation.

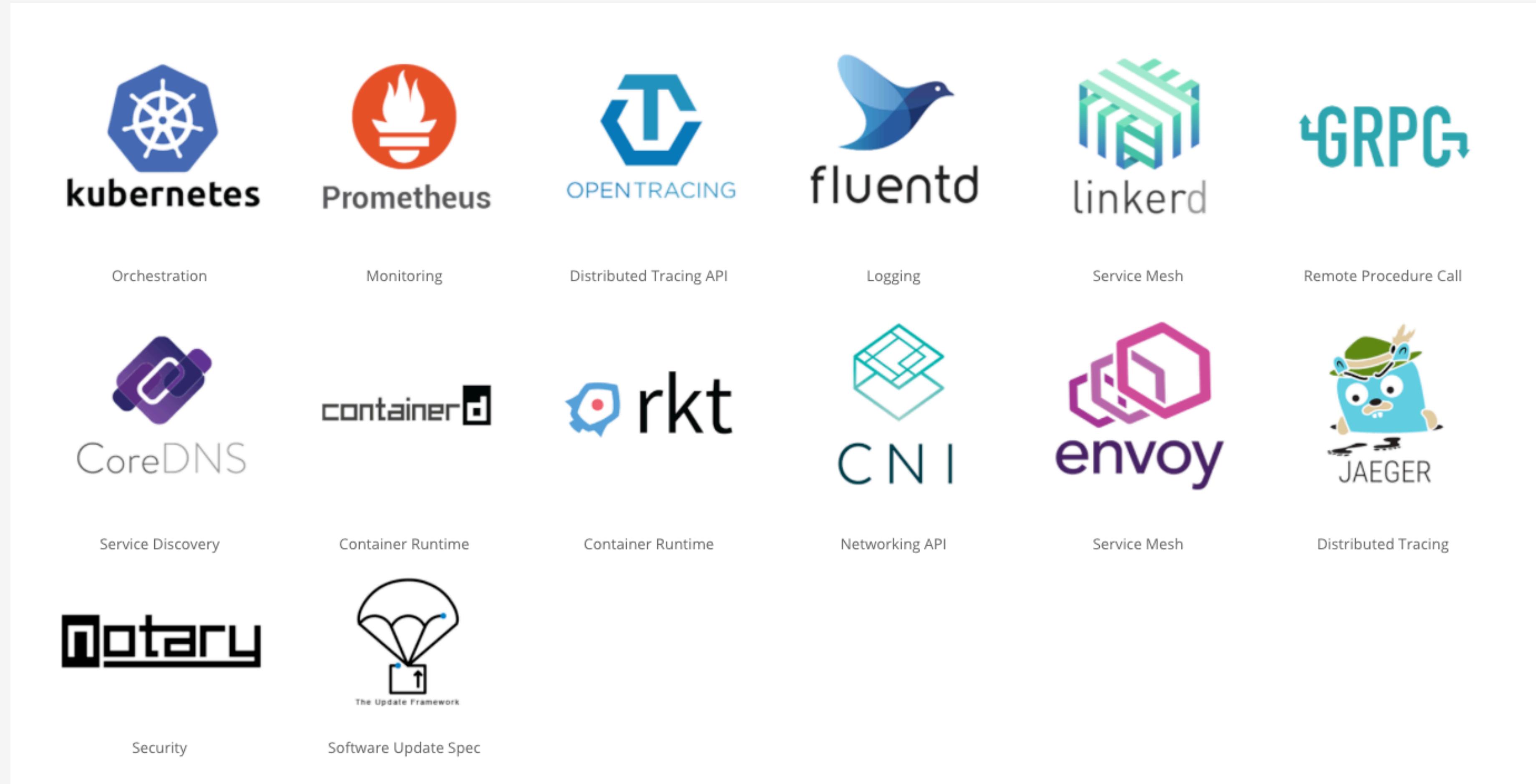
Dynamically managed

Actively scheduled and actively managed by a central orchestrating process.

Microservices oriented

Loosely coupled with dependencies explicitly described (e.g., through service endpoints).

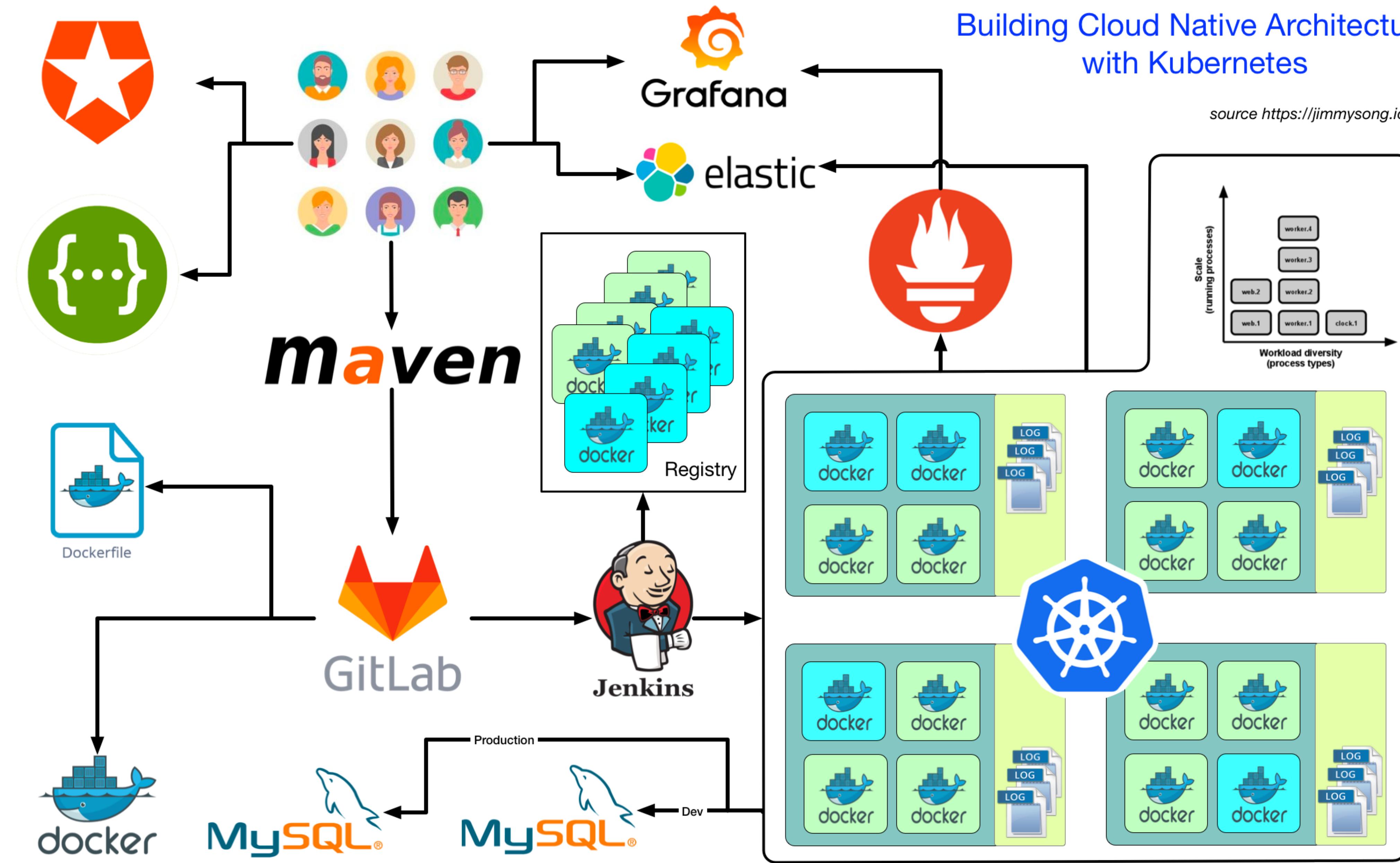
Cloud Native hosted projects



source <https://cncf.io>

Building Cloud Native Architecture with Kubernetes

source <https://jimmysong.io>



12 factors

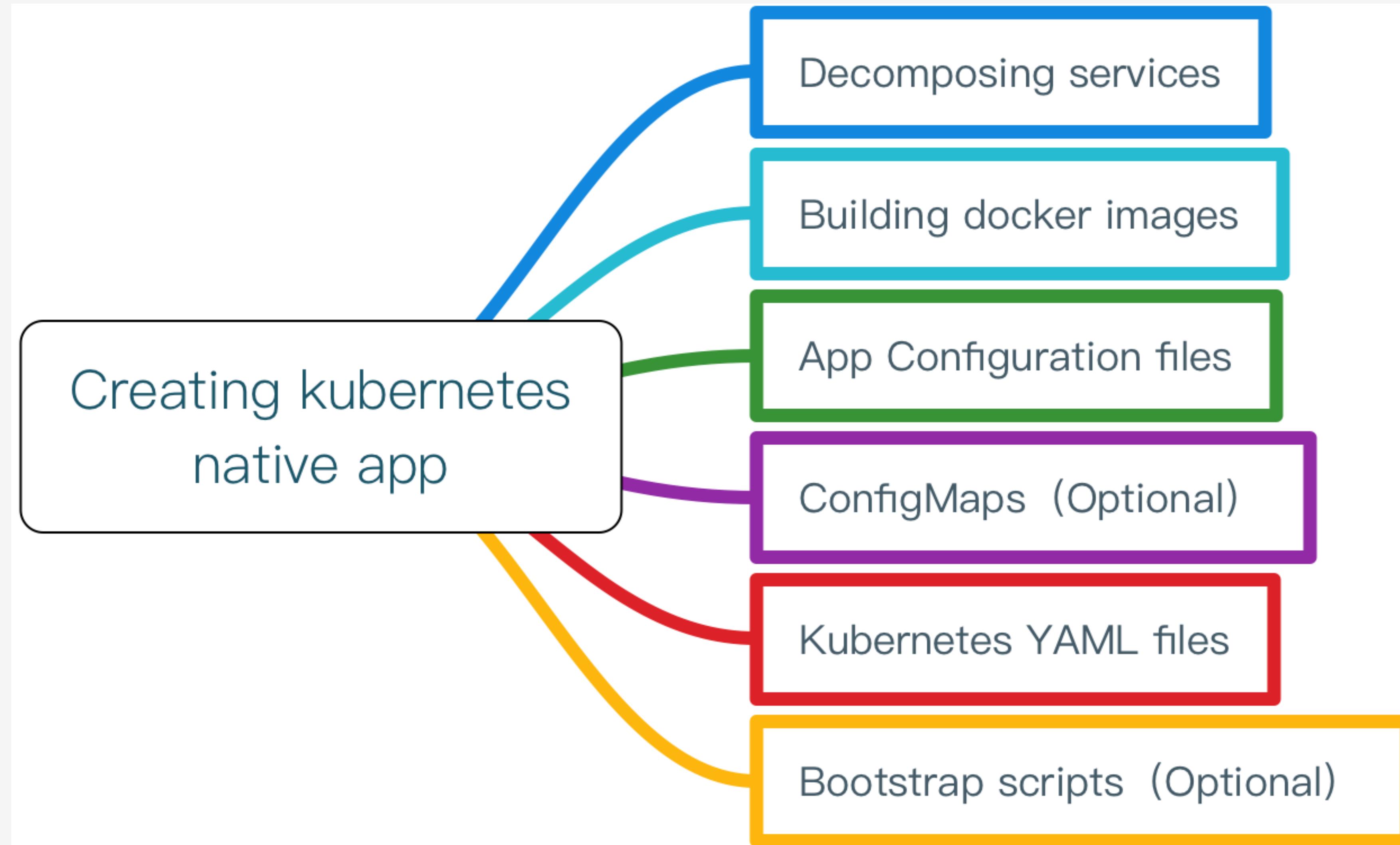
1. 基准代码
2. 依赖管理
3. 配置
4. 后端服务
5. 构建，发布，运行
6. 无状态进程
7. 端口绑定
8. 并发
9. 易处理
10. 开发环境与线上环境等价
11. 日志作为事件流
12. 管理进程

Additional

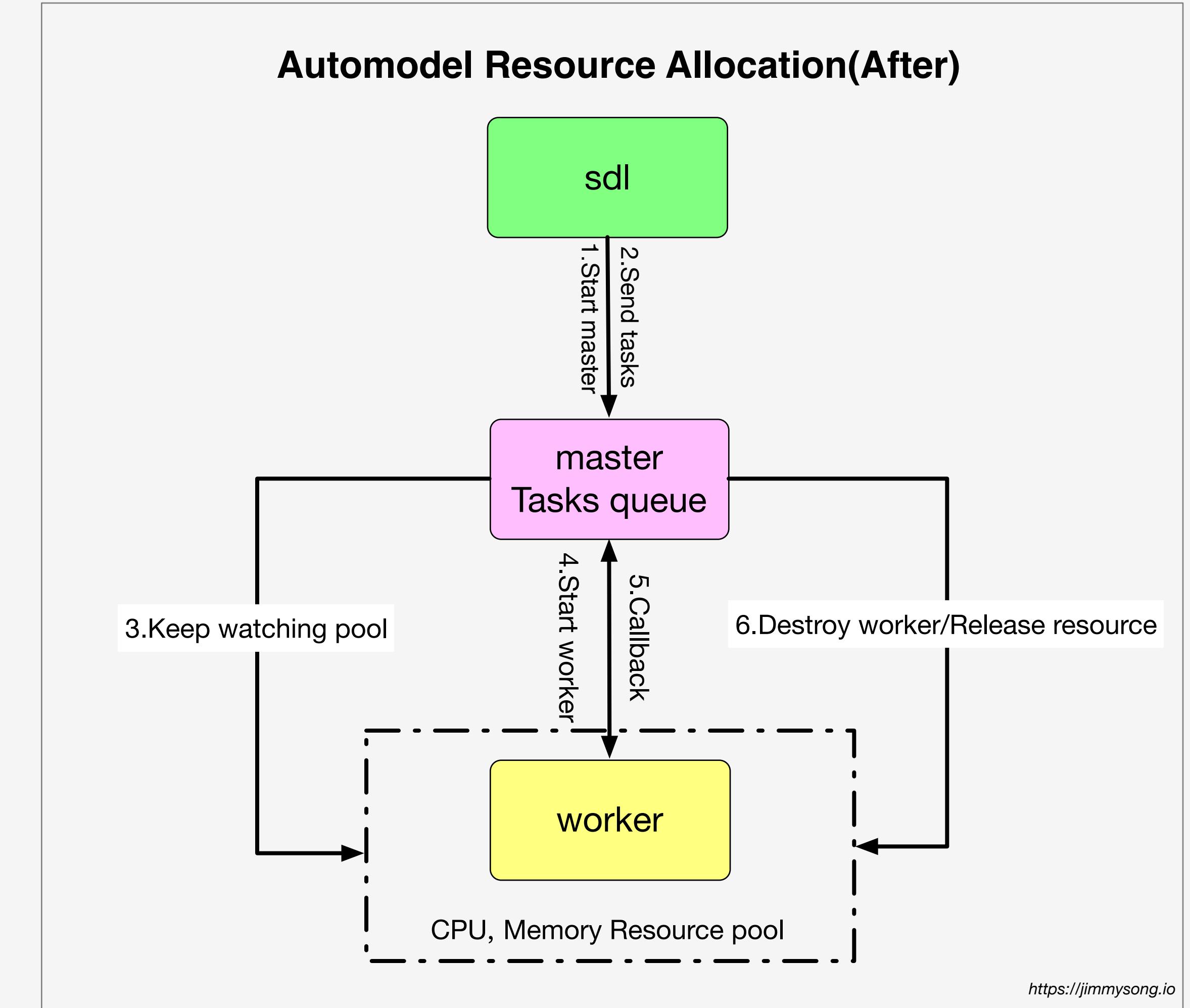
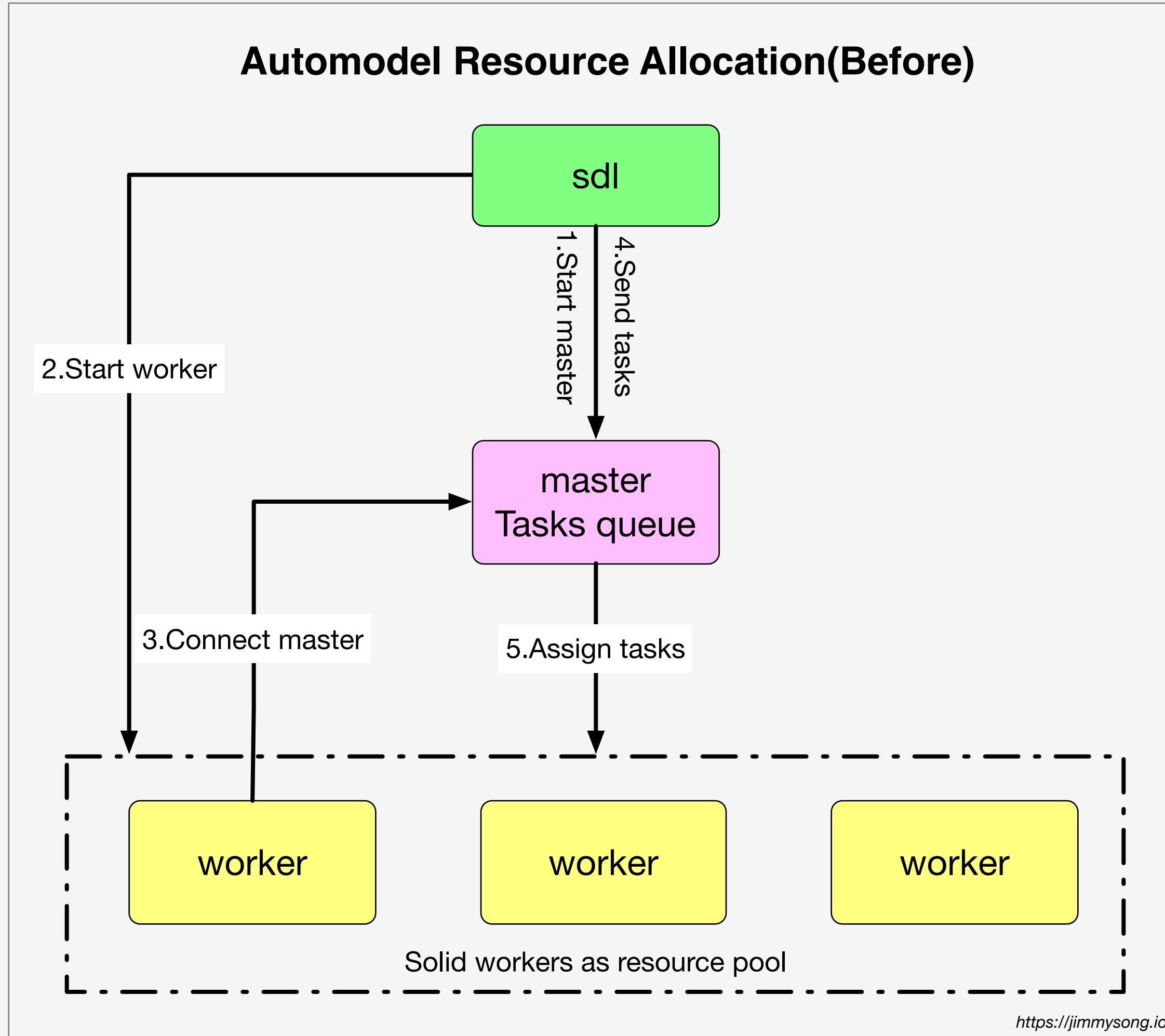
- API声明管理
- 认证和授权
- 监控与告警

source <https://12factor.net/>

How to create a Cloud Native app

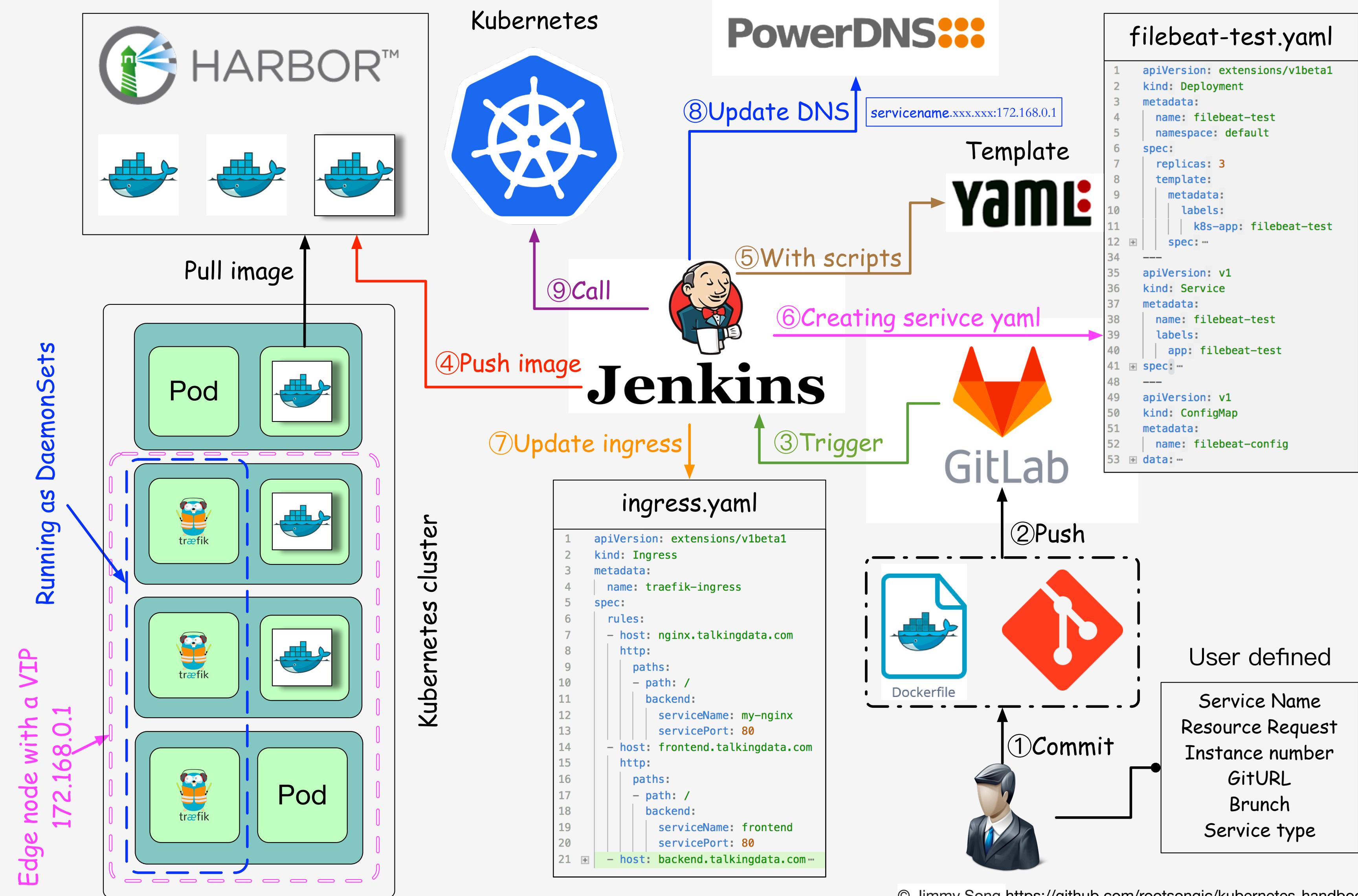


Move to Cloud



CI/CD

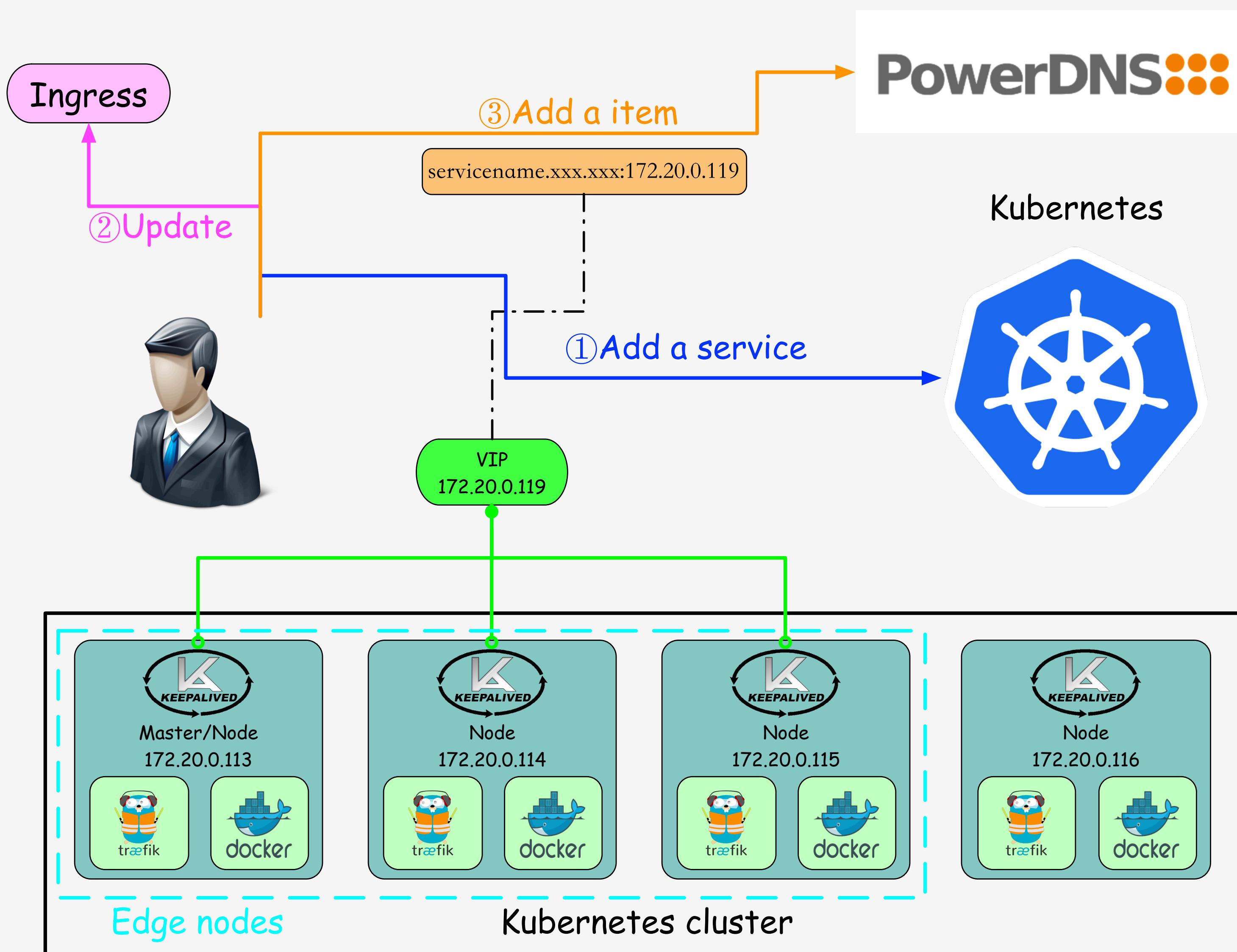
- Jenkins
- Harbor
- PowerDNS
- EFK



© Jimmy Song <https://github.com/rootsongjc/kubernetes-handbook>

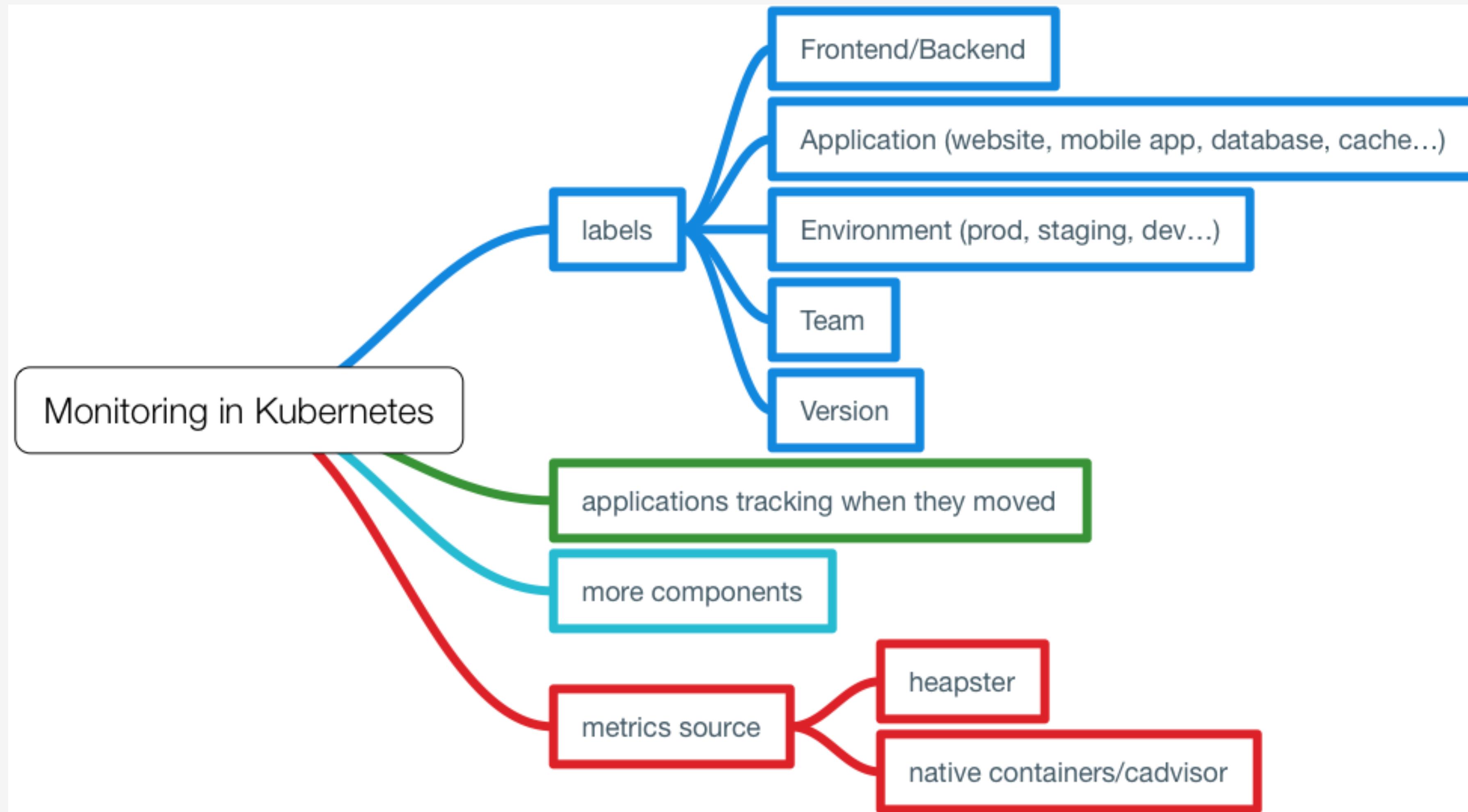
Edge node

- 边缘节点
- 流量出入口
- host + path



© Jimmy Song <https://github.com/rootsongjc/kubernetes-handbook>

Monitoring



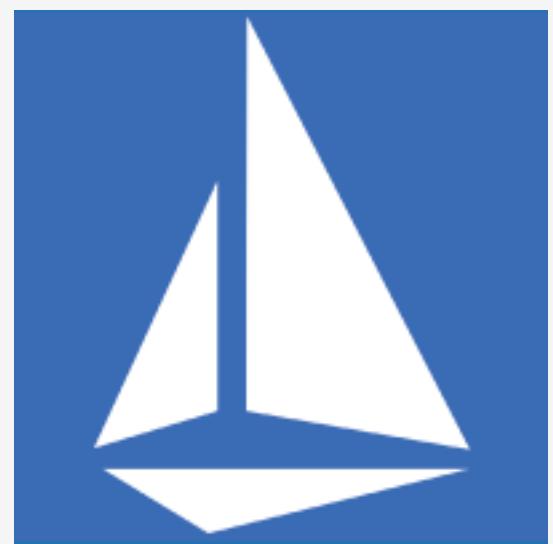
Outline

- Docker in TalkingData
- Containers
- Why Kubernetes?
- Microservices
- Cloud Native
- Service Mesh
- Use Cases
- Open Source



Service Mesh

Microservices Government on Kubernetes



Istio



Linkerd

- Traffic Management : API网关
- Observability : 服务调用和性能分析
- Policy Enforcement : 控制服务访问策略
- Service Identity and Security : 安全保护

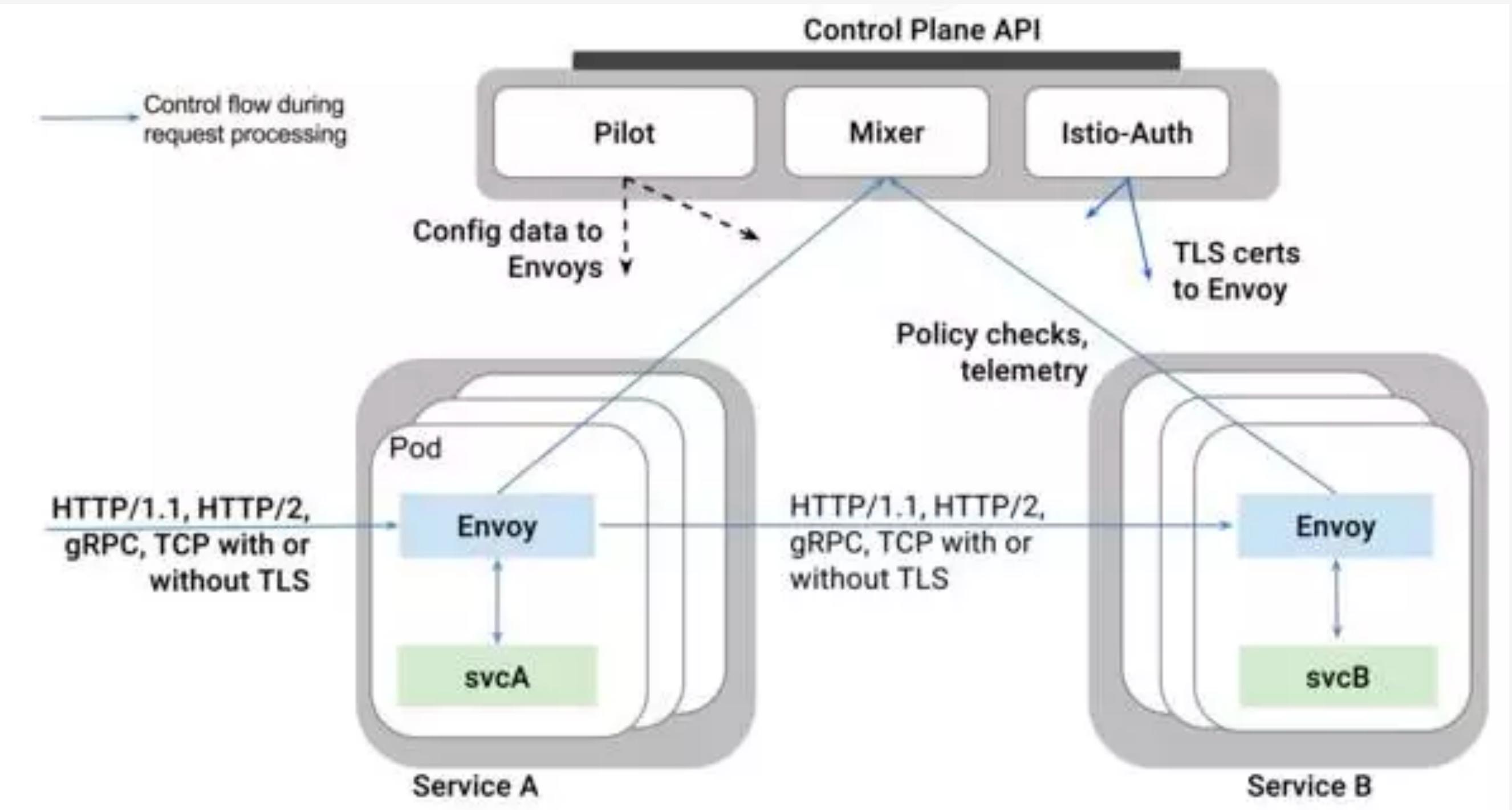
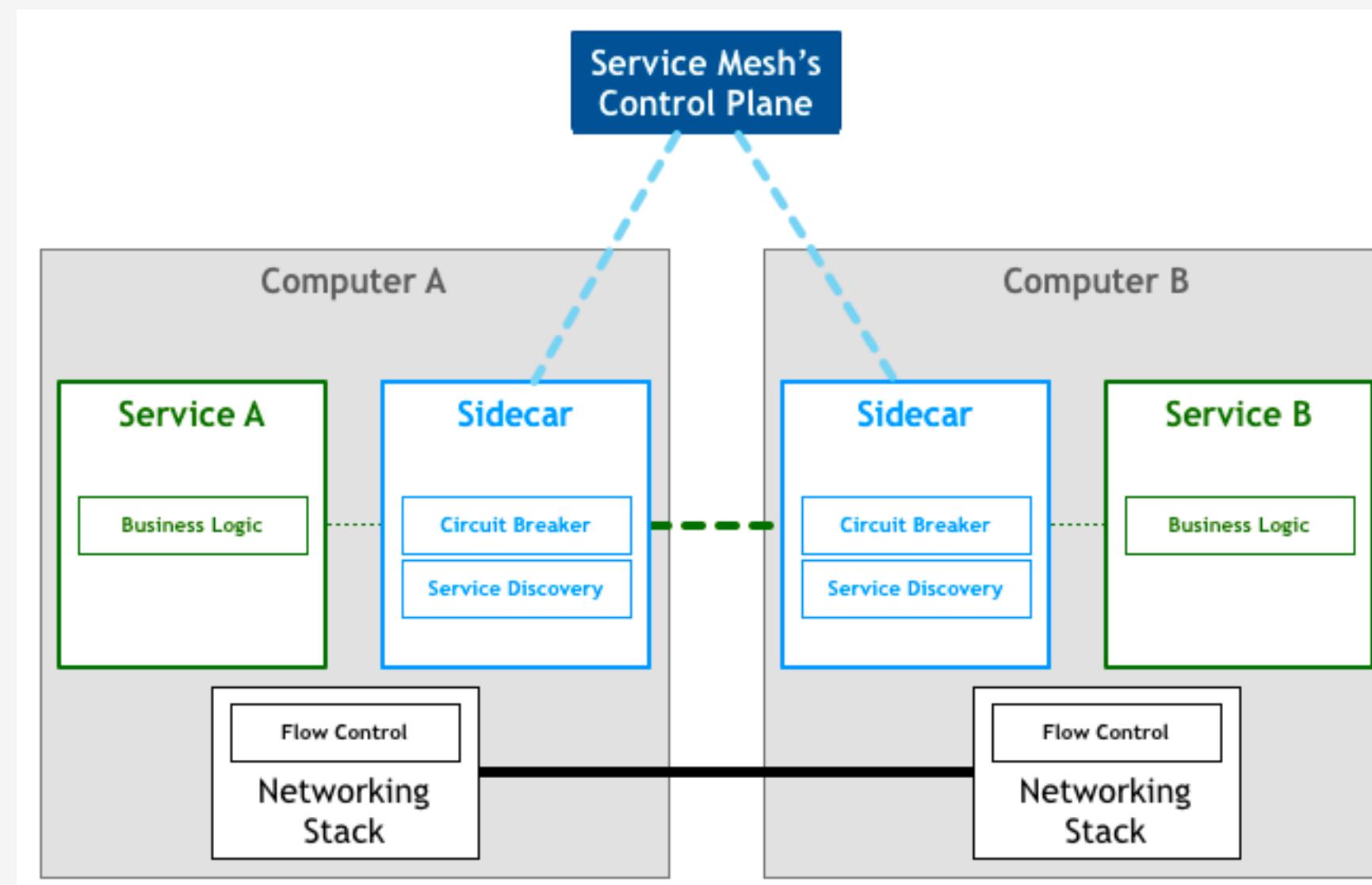
- 专用的基础设施层
- 轻量级高性能网络代理
- 提供安全的、快速的、可靠地服务间通讯
- 扩展kubernetes的应用负载均衡机制，实现灰度发布
- 完全解耦于应用，应用可以无感知，加速应用的微服务和云原生转型



<https://github.com/runconduit/conduit>

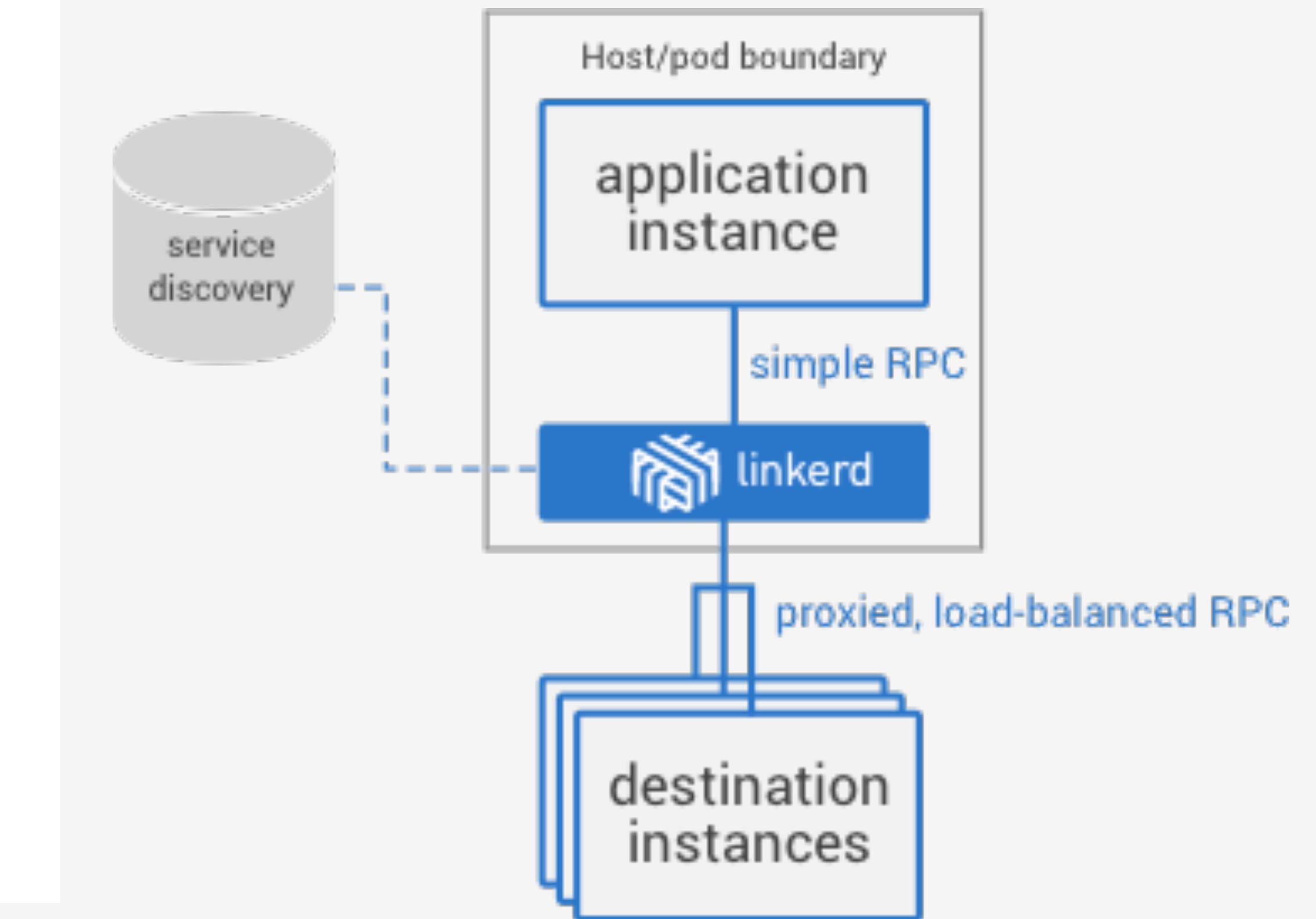
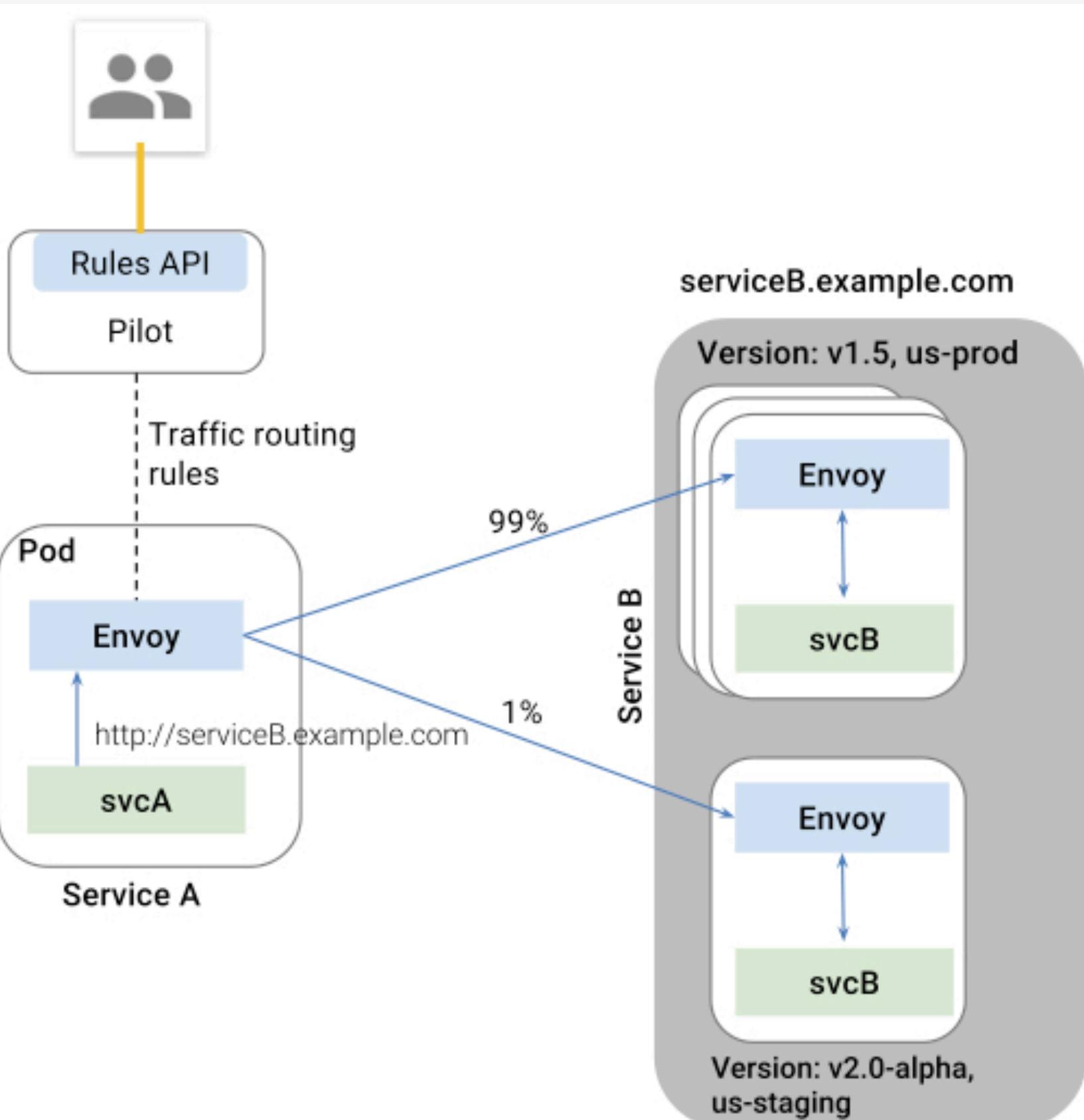
Service Mesh

Istio VS Linkerd



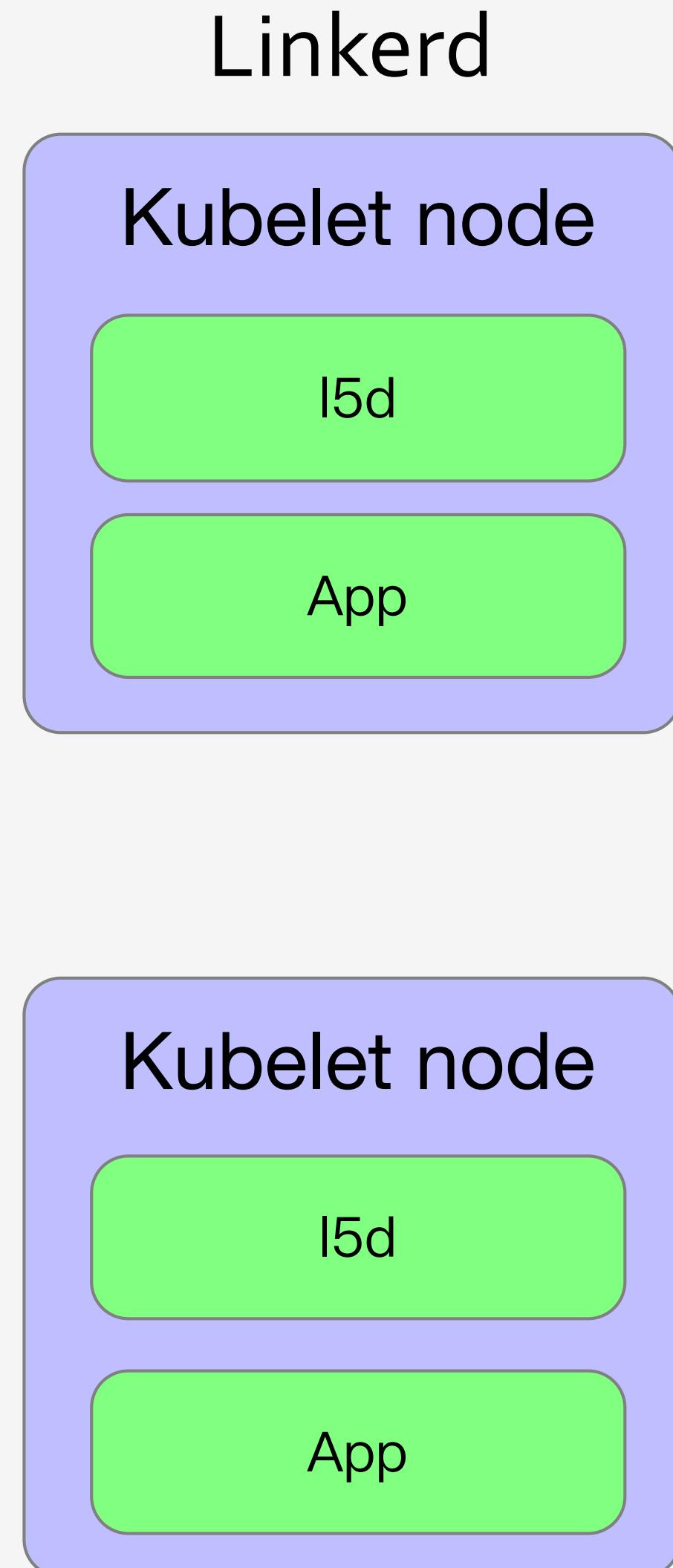
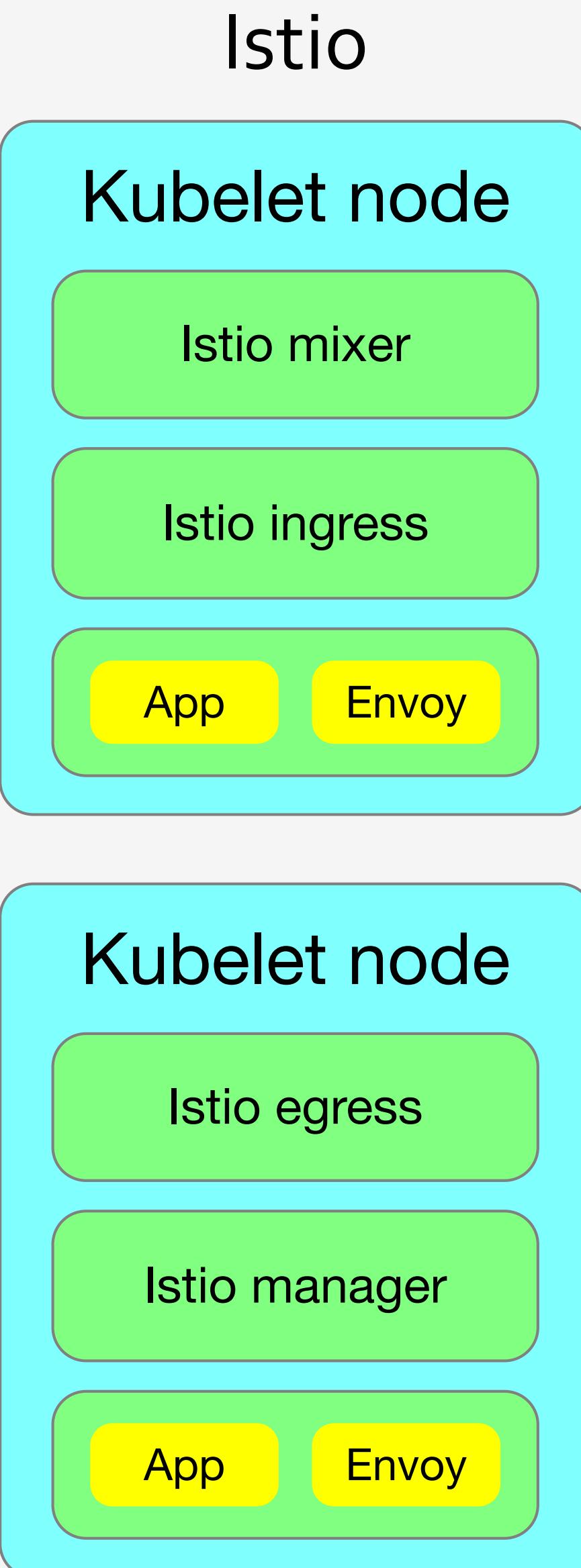
Service Mesh中文网 : <http://www.servicemesh.cn/>

Istio vs Linkerd

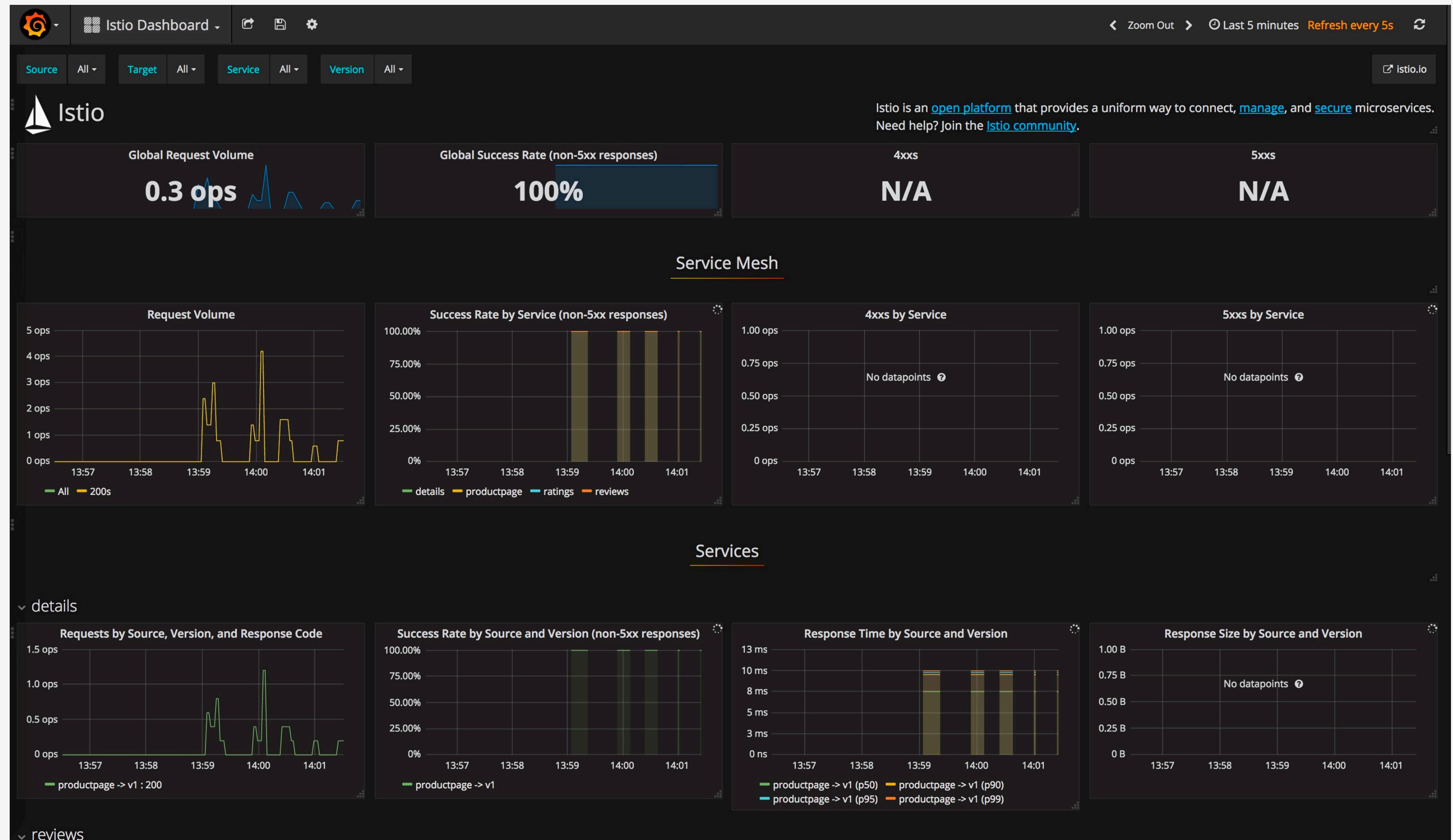


Istio vs Linkerd

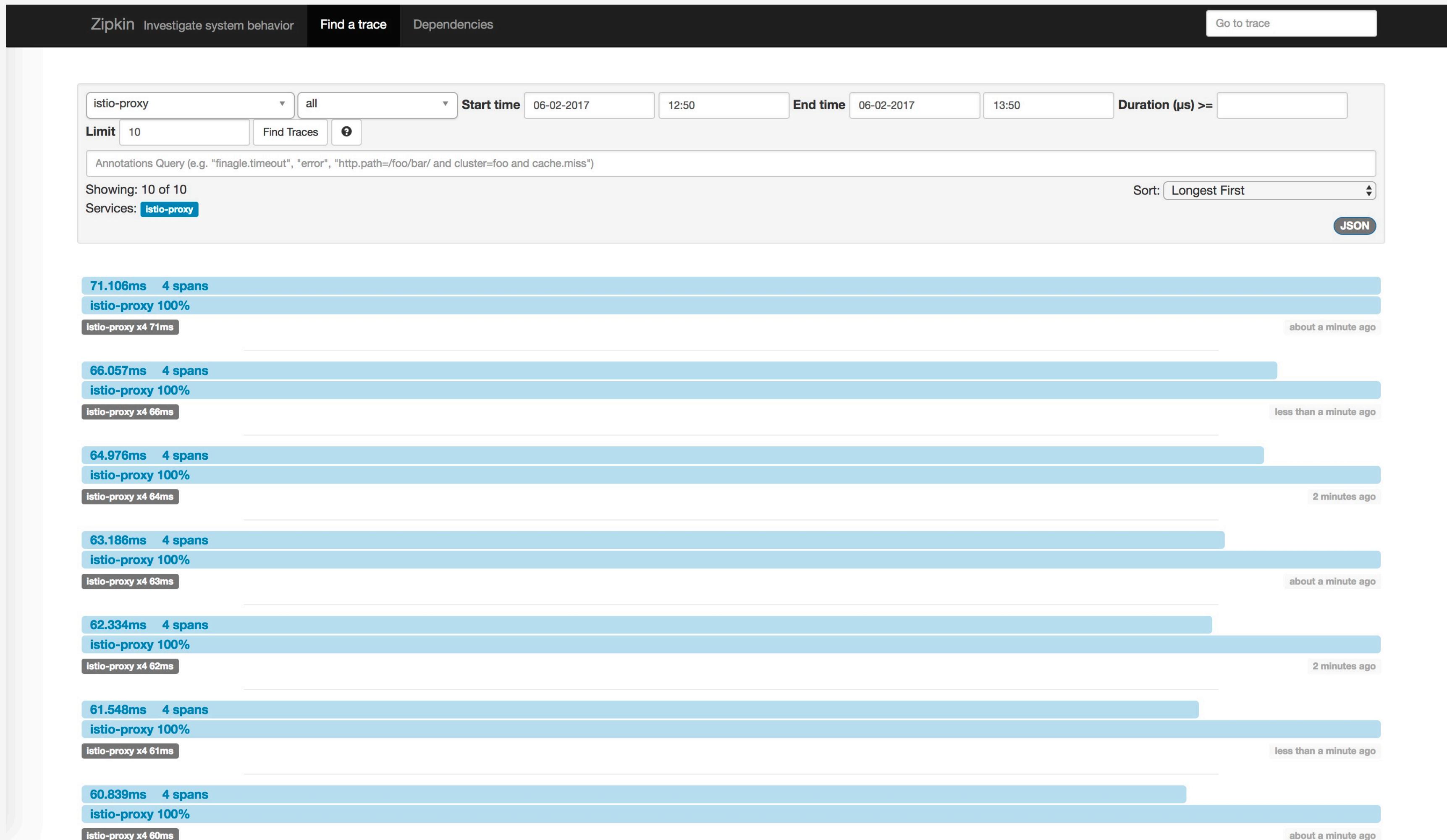
Feature	Istio	Linkerd
Deployment	Envoy/ Sidecar	DaemonSets
Easy to use	complicated	easy
Platform	kubernetes	kubernetes/ mesos/Istio/local
Version	0.3.0	1.3.3
Production	No	Yes



Grafana



Zipkin

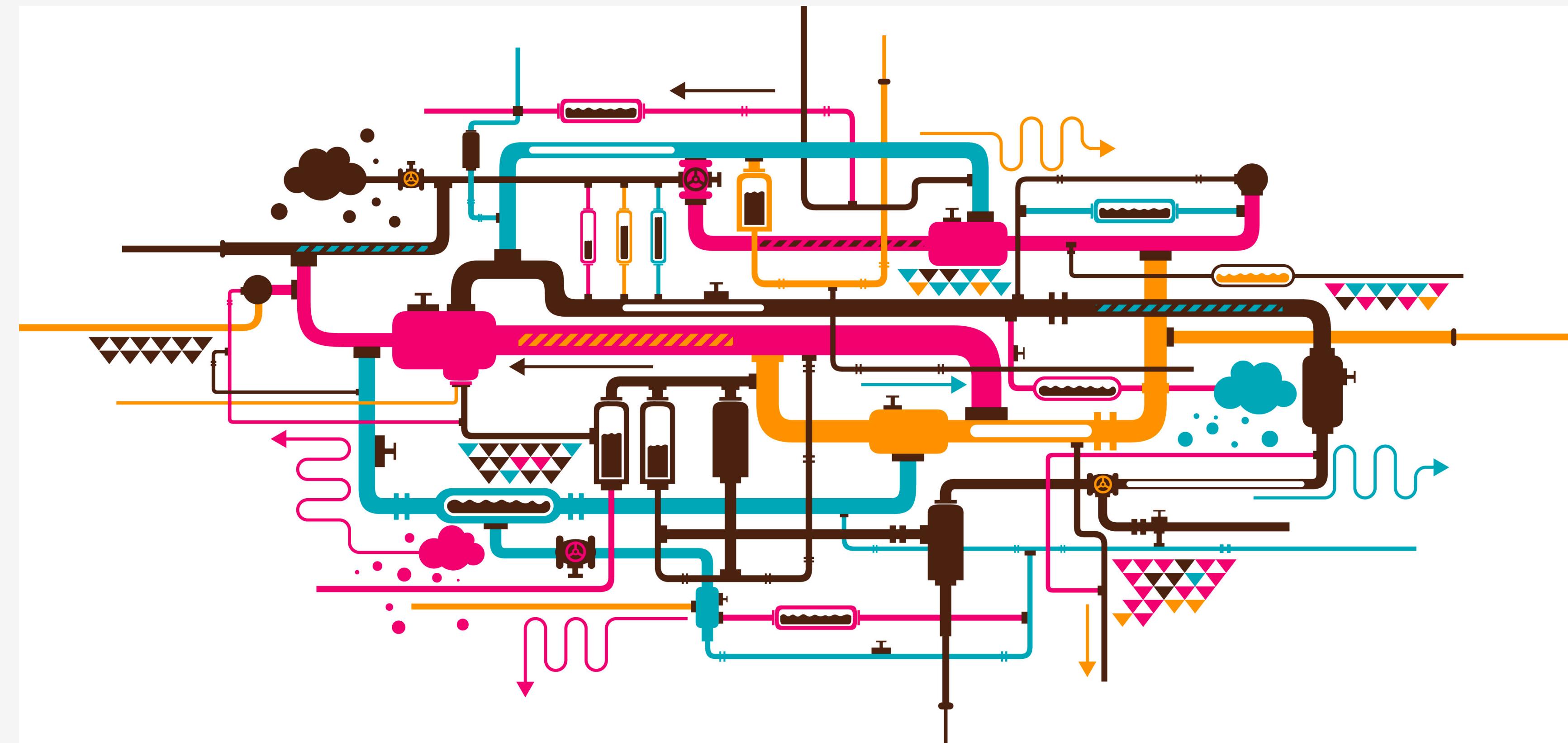


Weave scope



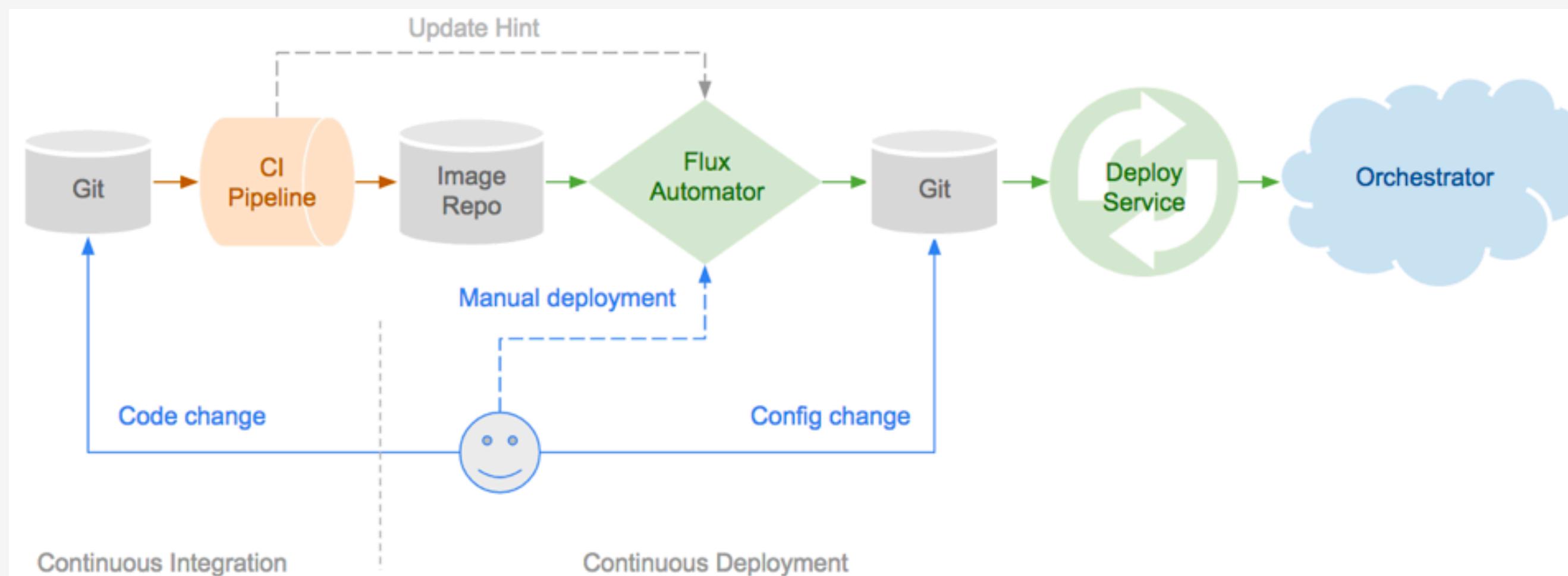
Use Cases

- Docker in TalkingData
- Containers
- Why Kubernetes?
- Microservices
- Cloud Native
- Service Mesh
- Use Cases
- Open Source

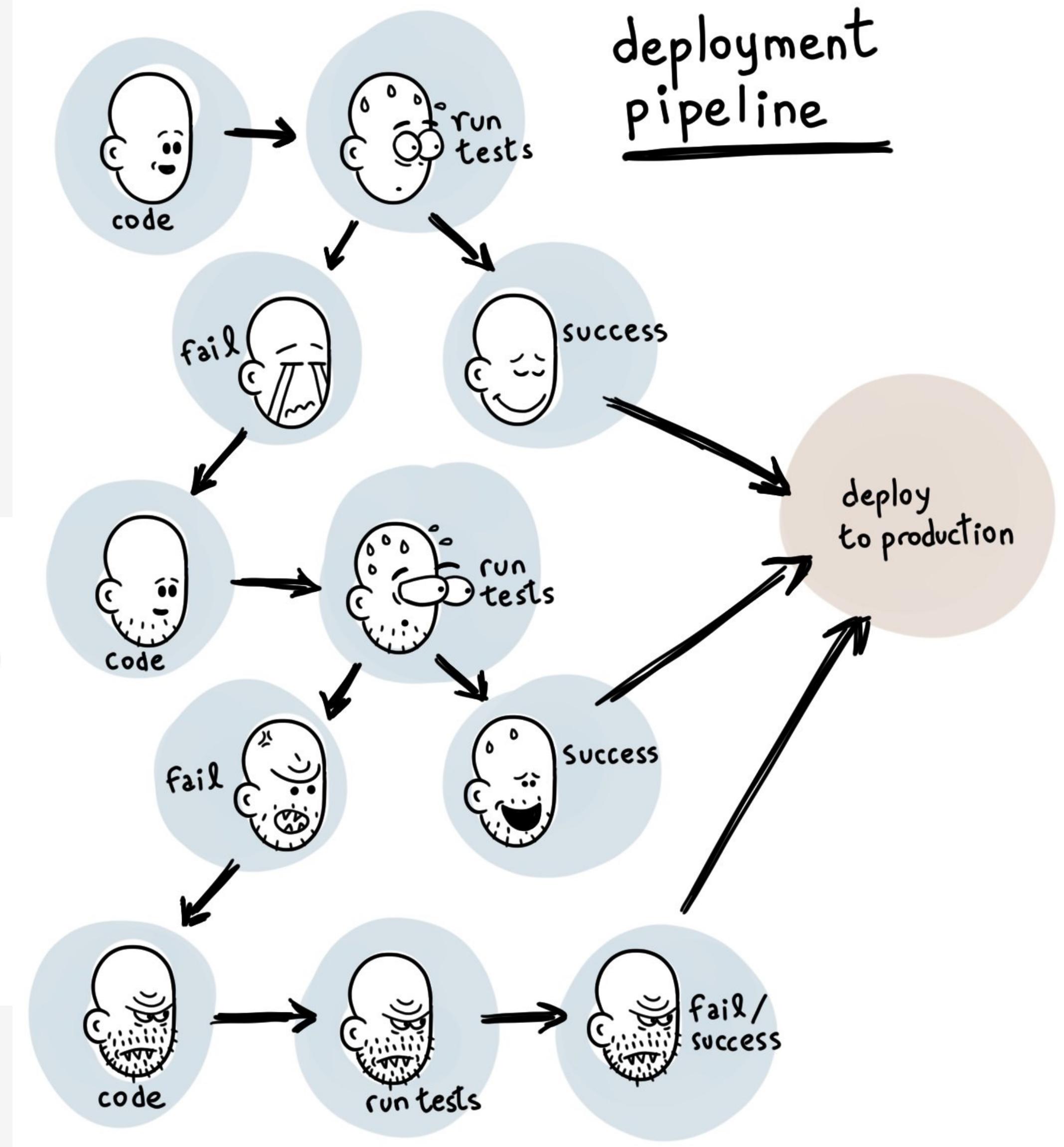


GitOps

- Infrastructure as code
- YAML! YAML! YAML!
- Git version control



source <https://www.weave.works>



GitOps

- 上线/更新/下线
- YAML! YAML! YAML!
- Helm chart

上线申请

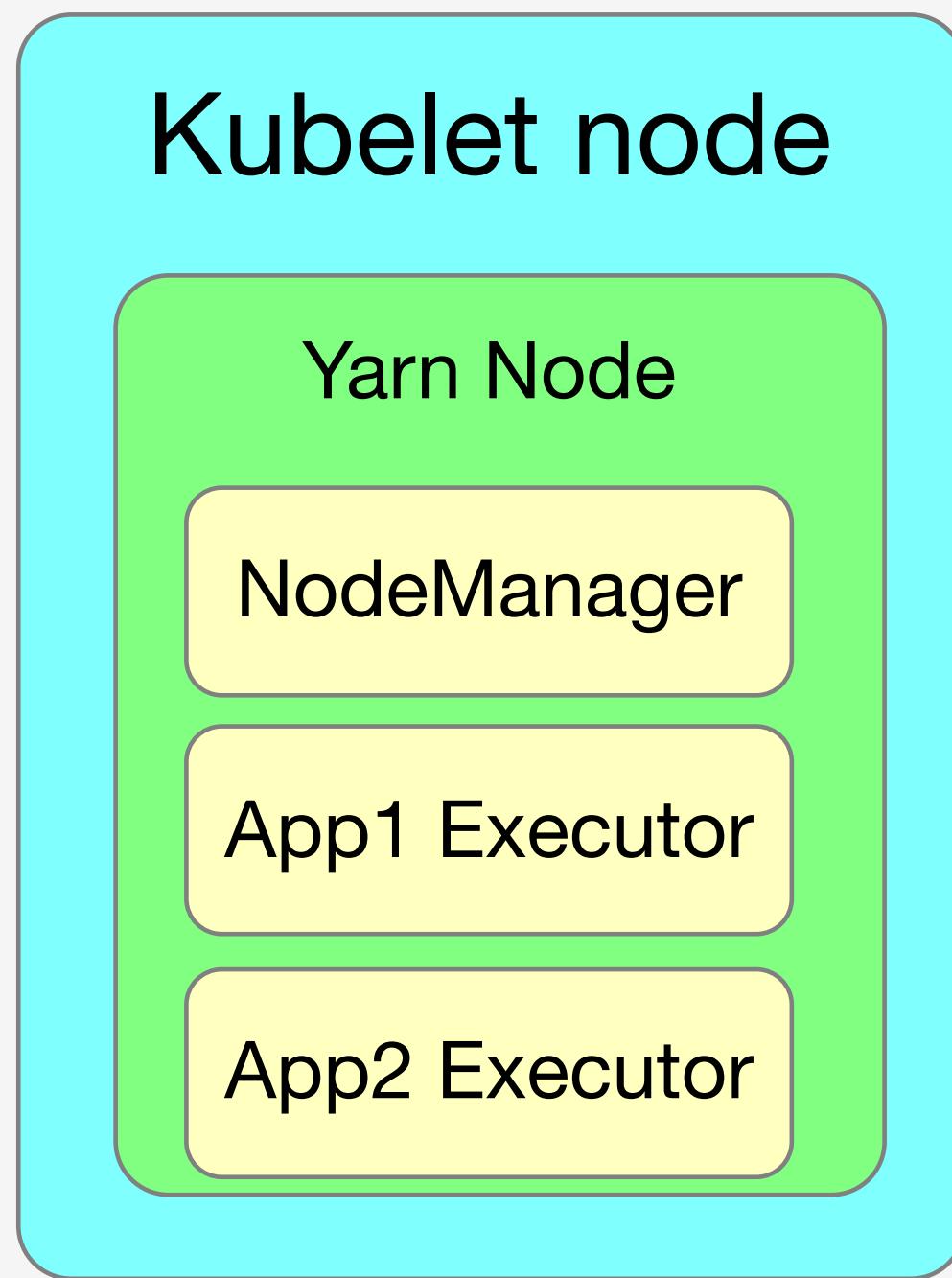
产品线	
服务类型	java
部署类型	docker
启动个数	
需要CPU	
	单位 : Core
需要内存	
	单位 : M
GitURL	必须是完整 URL
Branch	master
JDK版本	1.6.0.26

```
1 ---  
2 apiVersion: v1  
3 kind: Service  
4 metadata:  
5   name: zk-svc  
6   labels:  
7     app: zk  
8 spec:  
9   ports:  
10    - port: 2888  
11      name: server  
12    - port: 3888  
13      name: leader-election  
14    clusterIP: None  
15    selector:  
16      app: zk  
17 ---  
18 apiVersion: v1  
19 kind: ConfigMap  
20 metadata:  
21   name: zk-cm  
22 data:  
23   jvm.heap: "1G"  
24   tick: "2000"  
25   init: "10"  
26   sync: "5"  
27   client.cnxns: "60"  
28   snap.retain: "3"  
29   purge.interval: "0"  
30 ---  
31 apiVersion: policy/v1beta1  
32 kind: PodDisruptionBudget  
33 spec:  
34   metadata:  
35     name: sentinel-pdb  
36   spec:  
37     minAvailable: 2  
38 ---  
39 apiVersion: apps/v1beta1  
40 kind: StatefulSet  
41 spec:  
42   metadata:  
43     name: red-sentinel  
44   spec:  
45     replicas: 3  
46     selector:  
47       matchLabels:  
48         app: sentinel  
49     template:  
50       metadata:  
51         labels:  
52           app: sentinel  
53       spec:  
54         containers:  
55           - name: sentinel  
56             image: sz-pg-oam-docker-hub-001.tendcloud.com/library/redis:2.8.20  
57             ports:  
58               - port: 6379  
59             command:  
60               - /bin/bash  
61               - -c  
62               - "tail -f /log/test.log"  
63             volumeMounts:  
64               - name: app-logs  
65                 mountPath: /log  
66             affinity:  
67               podAntiAffinity:  
68                 requiredDuringSchedulingIgnoredDuringExecution:  
69                   - labelSelector:  
70                     matchExpressions:  
71                       - key: "app"
```

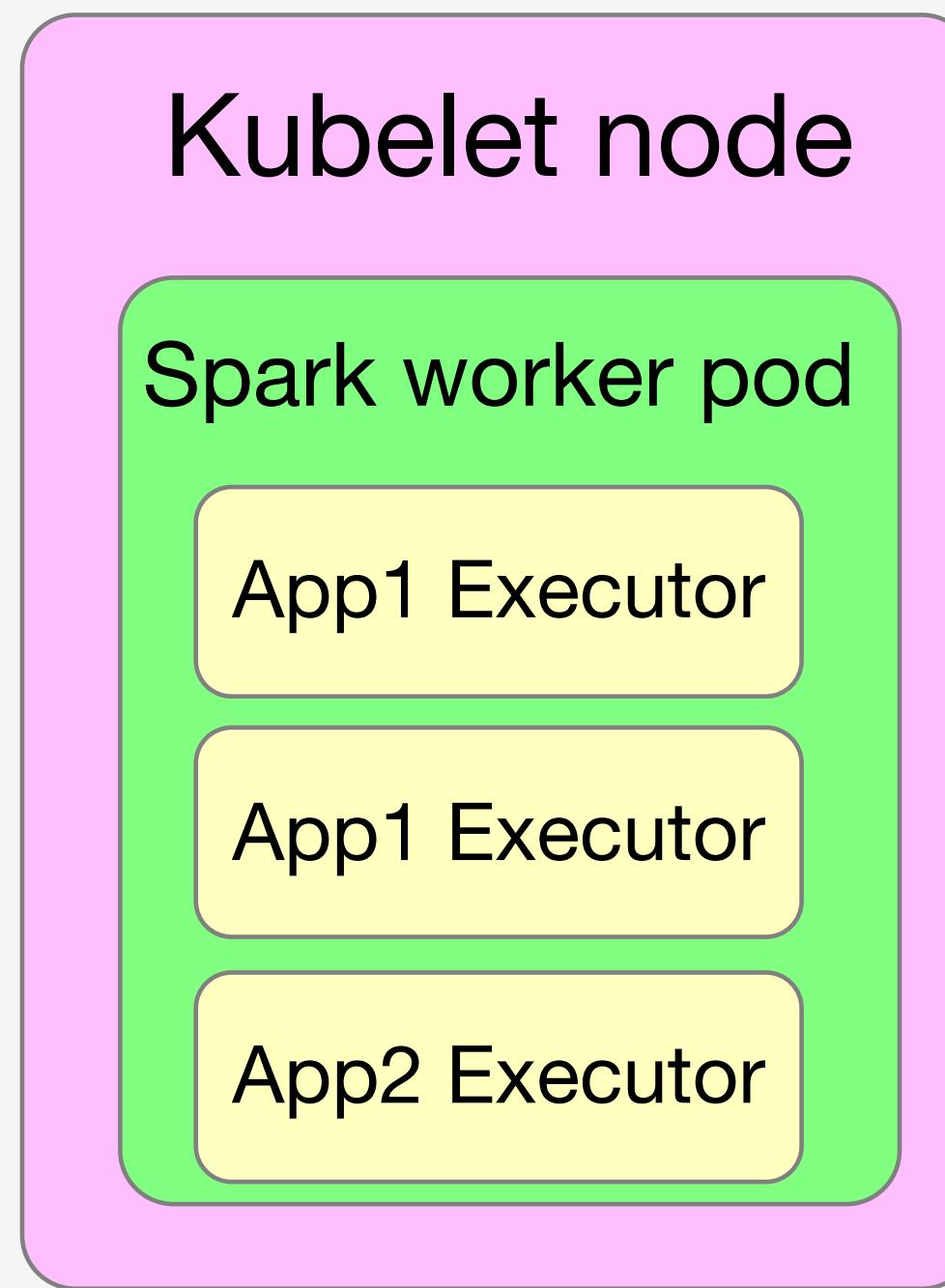
Spark on Kubernetes

Spark on Kubernetes with different schedulers

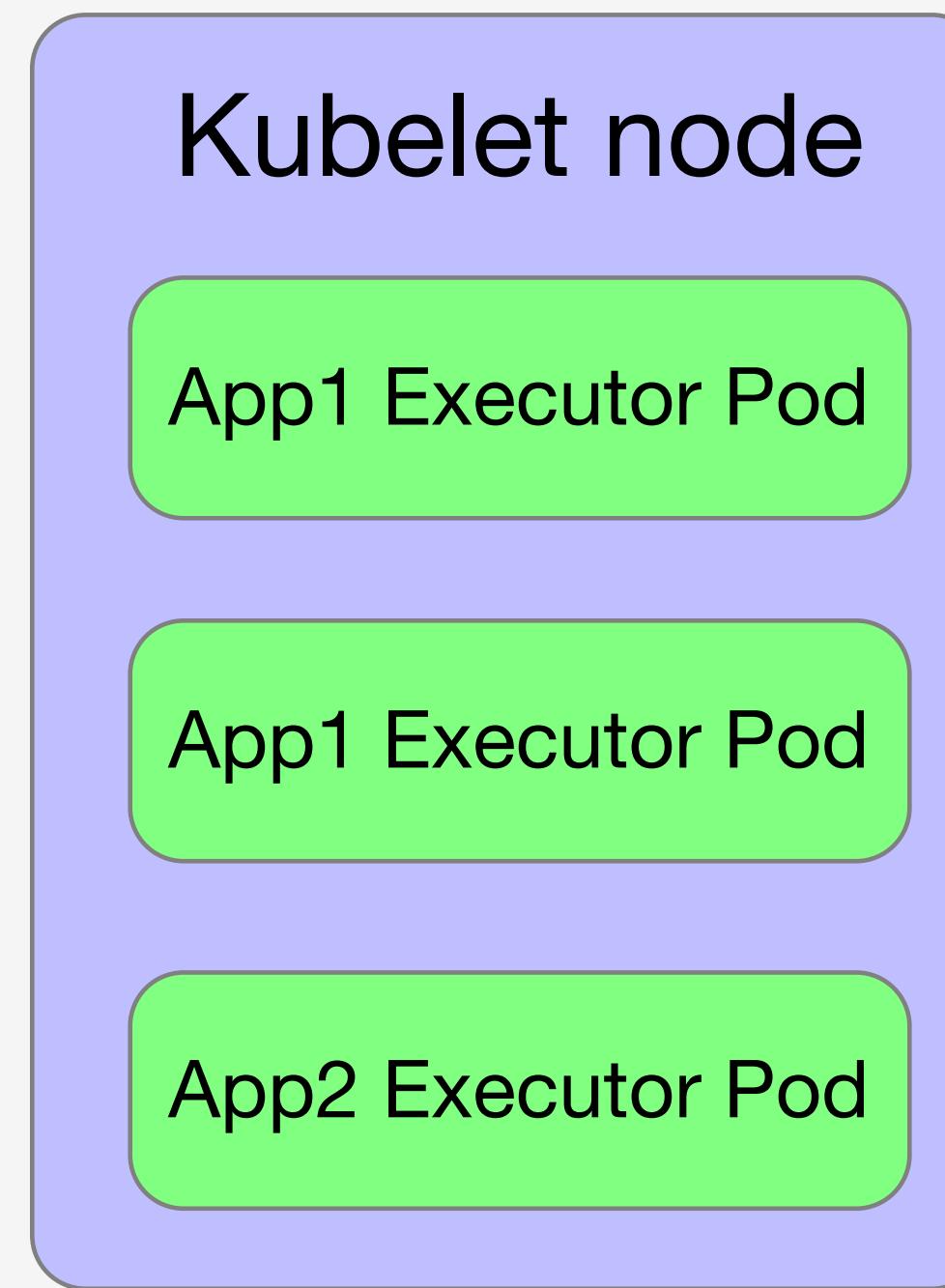
Yarn



Standalone

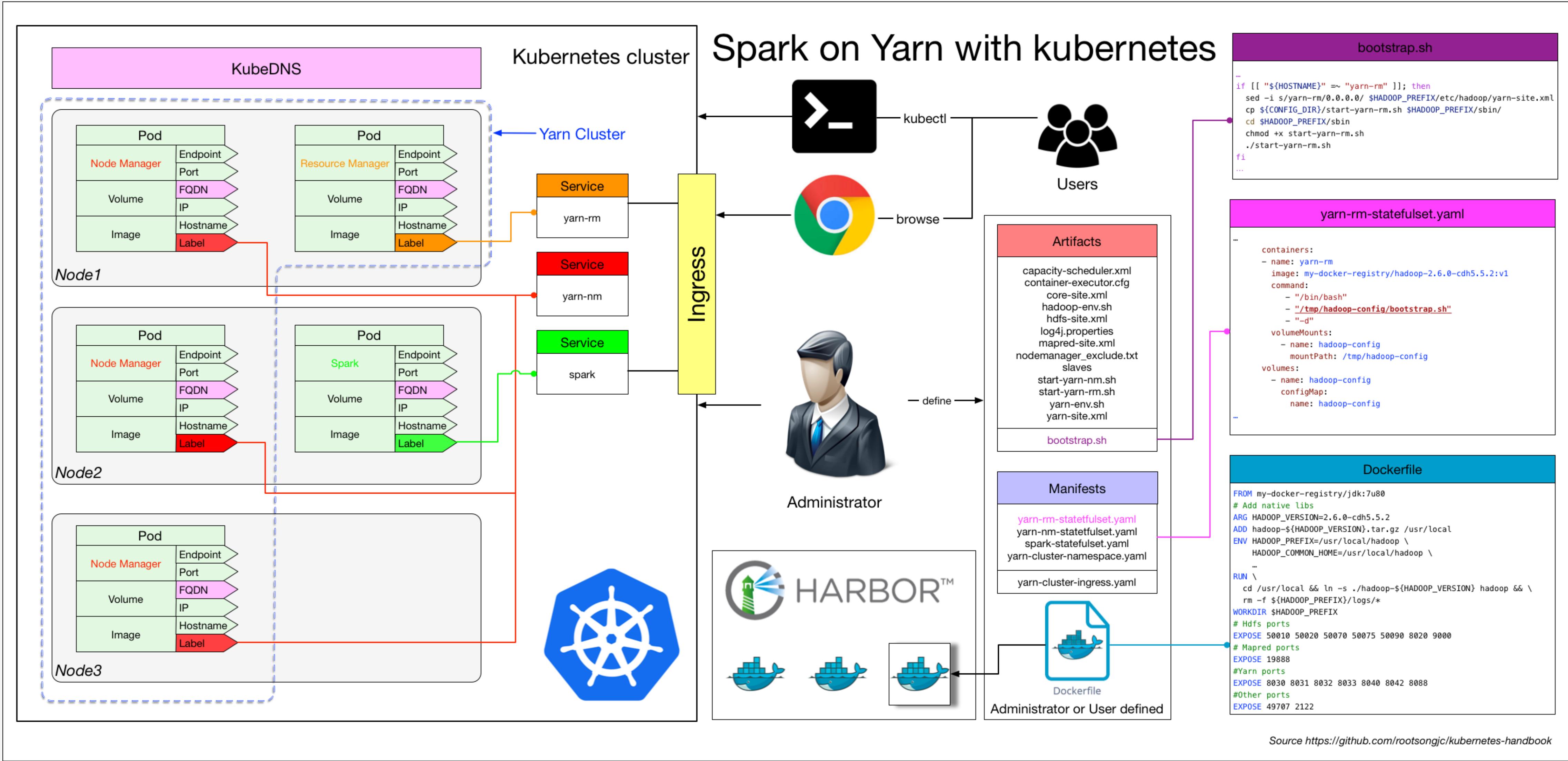


Native



<https://jimmysong.io>

Spark on Kubernetes with yarn scheduler



Spark on Kuberentes with native scheduler

- Kubernetes原生调度：与yarn、mesos同级
- 资源隔离，粒度更细：以namespace来划分用户
- 监控的变革：单次任务资源计量
- 日志的变革：pod的日志收集

```
./spark-submit \
--deploy-mode cluster \
--class com.talkingdata.alluxio.hadooptest \
--master k8s://https://172.20.0.113:6443 \
--kubernetes-namespace spark-cluster \
--conf spark.kubernetes.driverEnv.SPARK_USER=hadoop \
--conf spark.kubernetes.driverEnv.HADOOP_USER_NAME=hadoop \
--conf spark.executorEnv.HADOOP_USER_NAME=hadoop \
--conf spark.executorEnv.SPARK_USER=hadoop \
--conf spark.kubernetes.authenticate.driver.serviceAccountName=spark \
--conf spark.driver.memory=100G \
--conf spark.executor.memory=10G \
--conf spark.driver.cores=30 \
--conf spark.executor.cores=2 \
--conf spark.driver.maxResultSize=10240m \
--conf spark.kubernetes.driver.limit.cores=32 \
--conf spark.kubernetes.executor.limit.cores=3 \
--conf spark.kubernetes.executor.memoryOverhead=2g \
--conf spark.executor.instances=5 \
--conf spark.app.name=spark-pi \
--conf spark.kubernetes.driver.docker.image=spark-driver:v2.1.0-kubernetes-0.3.1-1 \
--conf spark.kubernetes.executor.docker.image=spark-executor:v2.1.0-kubernetes-0.3.1-1 \
--conf spark.kubernetes.initcontainer.docker.image=spark-init:v2.1.0-kubernetes-0.3.1-1 \
--conf spark.kubernetes.resourceStagingServer.uri=http://172.20.0.114:31000 \
~/Downloads/tendcloud_2.10-1.0.jar
```

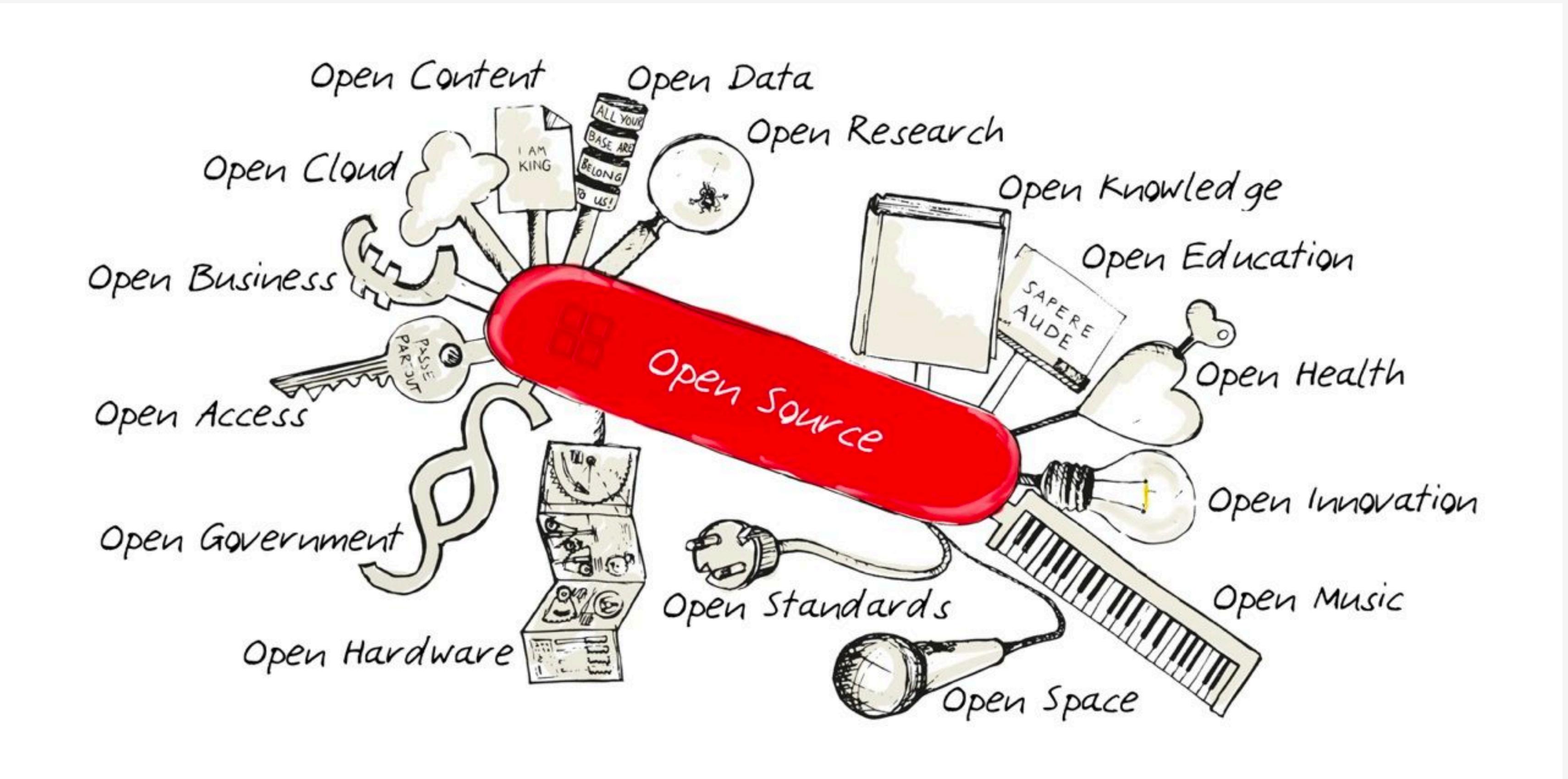
Spark on Yarn V.S Spark on Kuberentes

Feature	Yarn	Kubernetes
queue	queue	namespace
instance	ExcuterContainer	Executor Pod
network	host	plugin
heterogeneous	no	yes
security	RBAC	ACL

<https://github.com/apache-spark-on-k8s/spark>

Outline

- Docker in TalkingData
- Containers
- Why Kubernetes?
- Microservices
- Cloud Native
- Service Mesh
- Use Cases
- Open Source



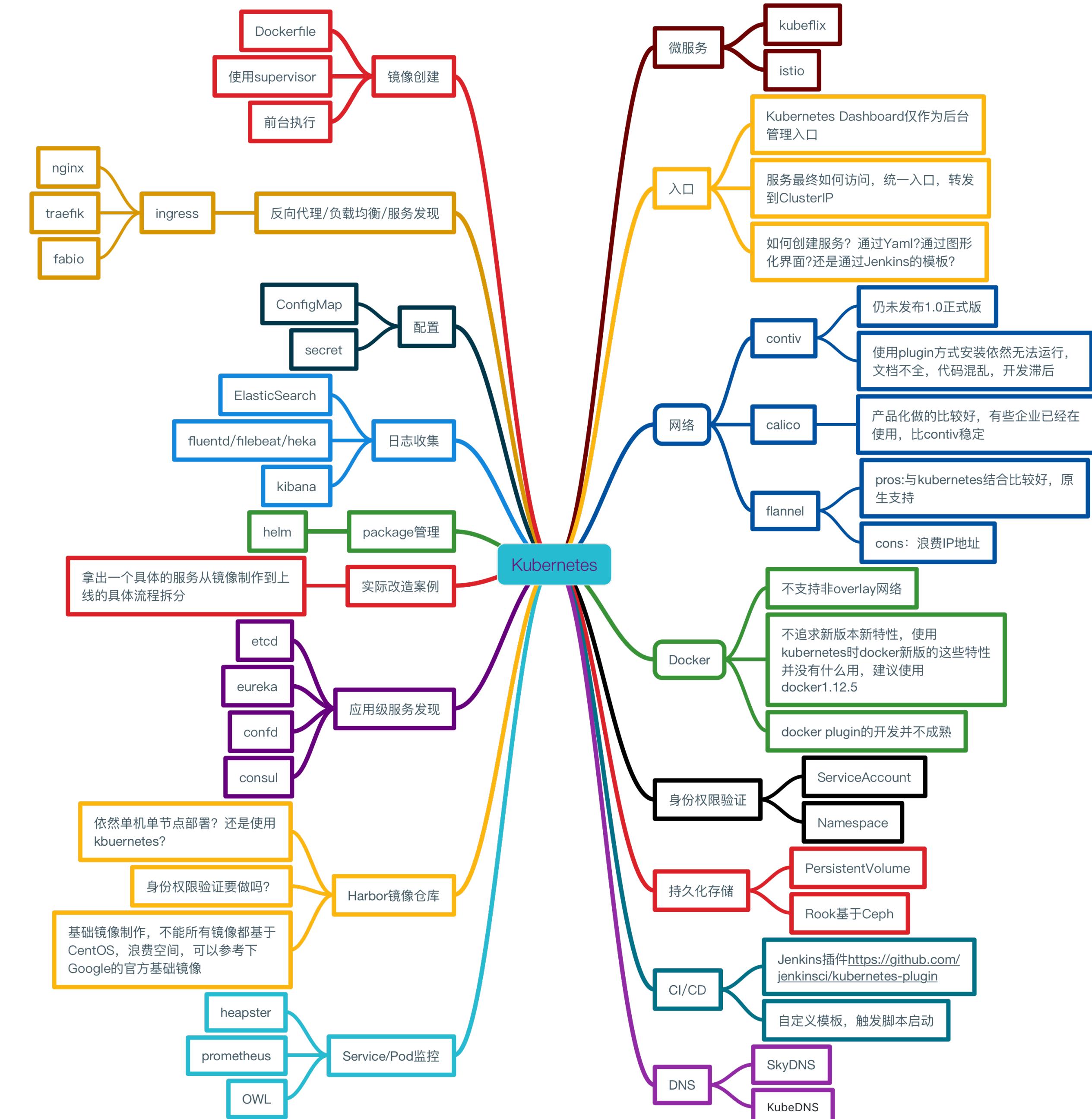
Open Source

优点

- 解决方案
- 减轻开发成本
- 社区参与
- 创新
- 人才机制

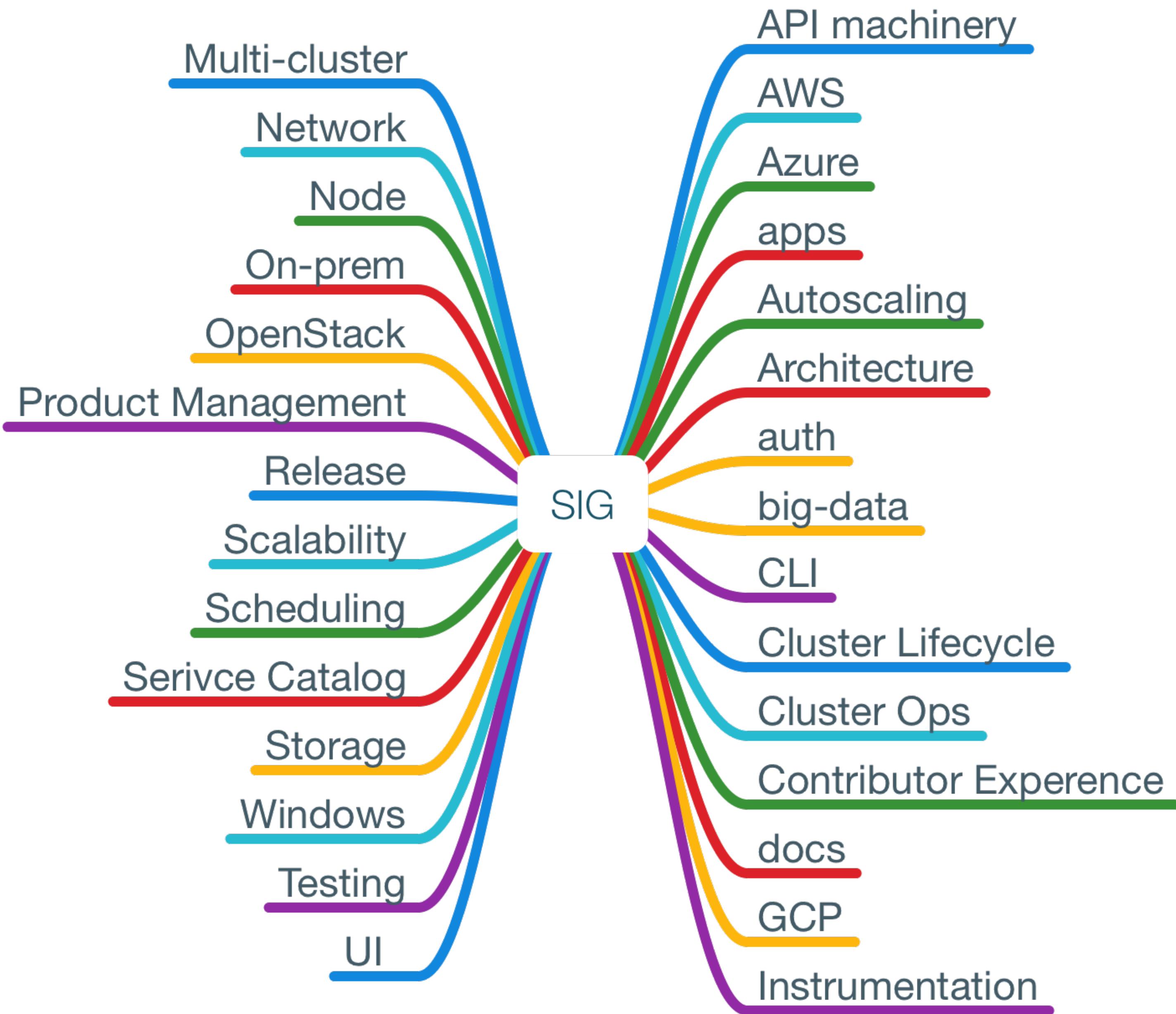
缺点

- API变动频繁
- 集成测试困难
- 持续跟进



SIGs

- 社区参与
- 对症下药
- 分享交流



Access kubernetes Cluster



```
kube-shell> kubectl top node
NAME          CPU(cores)   CPU%      MEMORY(bytes)  MEMORY%
172.20.0.114  734m        1%       34540Mi        26%
172.20.0.115  308m        0%       9334Mi         7%
172.20.0.113  894m        2%       50624Mi        39%
kube-shell> kubectl describe
persistentvolumeclaim
endpoints
horizontalpodautoscaler
job
storageclass
secret
serviceaccount
```

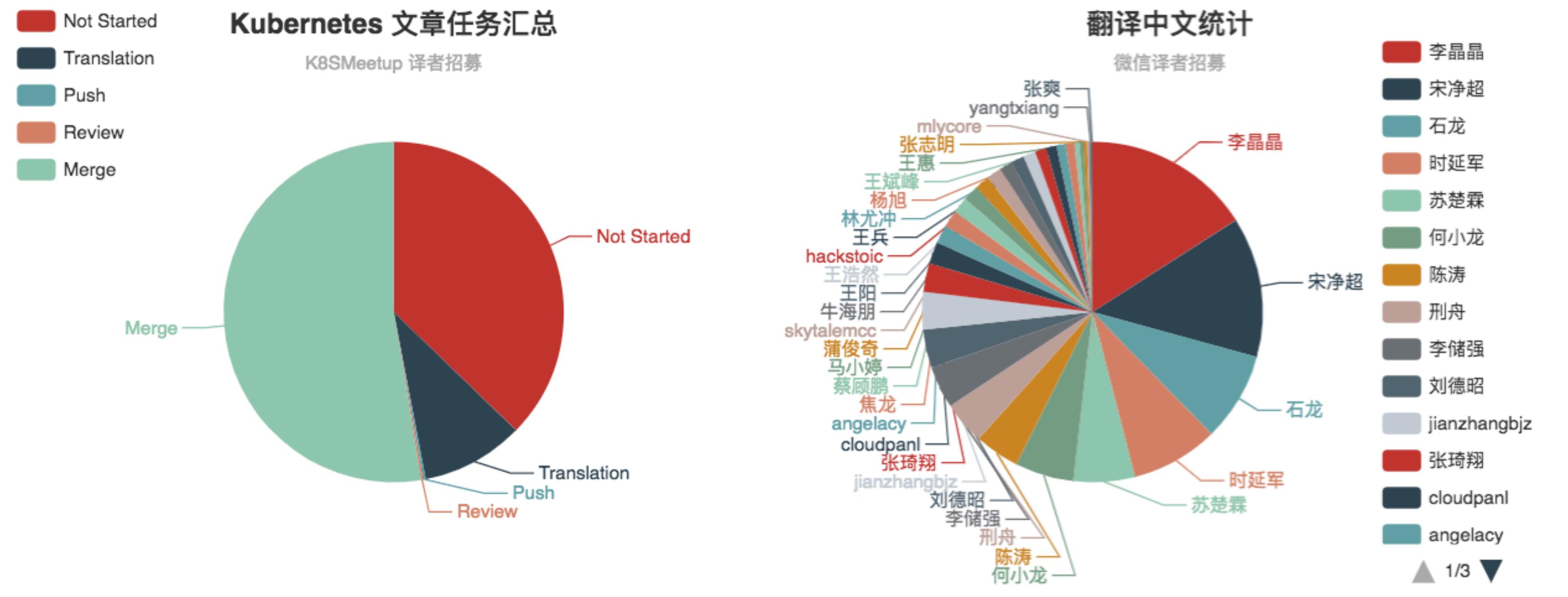
Kubernetic - The Desktop Client for Kubernetes

Deployments > auto-model-master-1510556211771

NAME	TYPE	READY	STATUS
auto-model-master-1510556211771	Deployment	2/2/2/2	test_40
auto-model-master-15105562117...	ReplicaSet	2/2	test_40
auto-model-master-15105562117...	Pod	1/1	Running
auto-model-master-15105562117...	Pod	1/1	Running

The screenshot shows the Kubernetic desktop client. At the top, it displays network and battery information. Below that is a search bar and a dropdown for 'Namespace: All namespaces'. The main area is a list of pods categorized by type: Pods, Services, Nodes, and Deployments. Each pod entry includes a preview icon, the pod name, its type, and its current status. A large red 'carbin' logo is visible at the bottom right.

kubernetes-docs-cn



- 34 contributors
- 27万字
- official Chinese docs

<https://github.com/kubernetes/kubernetes-docs-cn>

kubernetes-docs-cn

27万字

<https://k8smeetup.github.io/chart/>

Kubernetes 中文翻译汇总 (单位:字)

Kubernetes 中文站



张志明 王斌峰 蔡顾鹏 angelacy 何小龙 cloudpanl hackstoic jianzhangbjz 李储强 林尤冲 mlycore 王兵
宋净超 skytalemcc 蒲俊奇 王浩然 刑舟 yangtxiang 王惠 马小婷

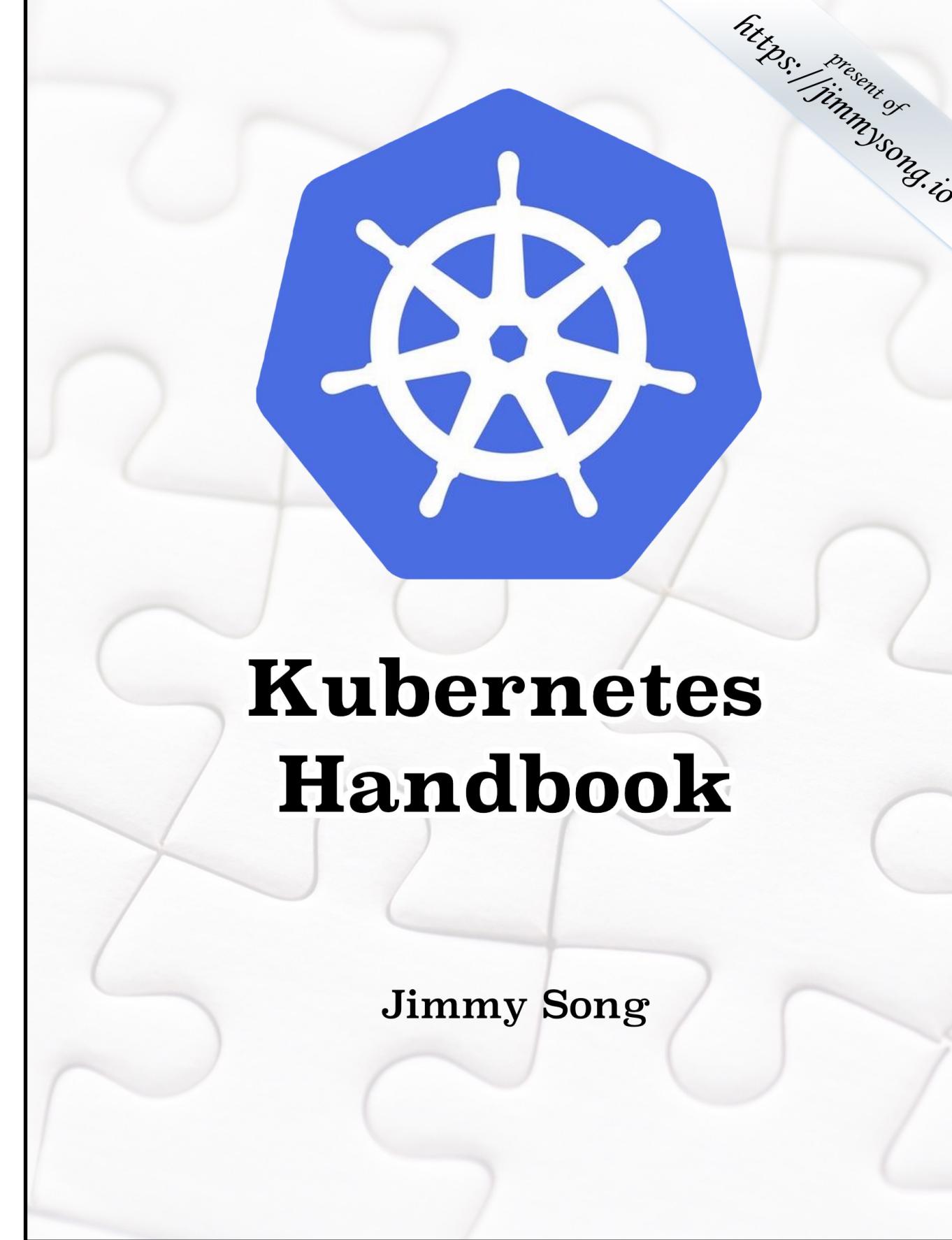
Open source

Contributing is **Not** only about code, it is about helping a community

- Blogposts: <https://jimmysong.io>
- 微信群: k8s&cloud native实战群
- Meetup: k8smeetup、DCI
- Gitbook: <https://jimmysong.io/kubernetes-handbook>
- Ecosystem: <https://jimmysong.io/awesome-cloud-native/>
- Share: <https://github.com/rootsongjc/cloud-native-slides-share>
- KubeCon + CloudNativeCon 2018年11月14-15日 上海



QA



<https://jimmysong.io>

<https://github.com/rootsongjc/kubernetes-handbook>

Thank you!