开源广进
用Service Catalog构造K8S服务能力中心

目录

- 什么是Open Service Broker API
- 什么是Service Catalog
- 什么是Service Broker
- 服务能力中心的实践

什么是Open Service Broker API

OPEN SERVICE BROKER API™

https://github.com/openservicebrokerapi/servicebroker.git

2011

- V1
- VMware开源
- MySQL
- PostgreSQL
- RabbitMQ

MongoDB

Redis

2013

- v2
- 将平台与服务提供解耦
- 定义与平台无关的 ServiceBroker API

2015

- 增加异步服务创建

2016

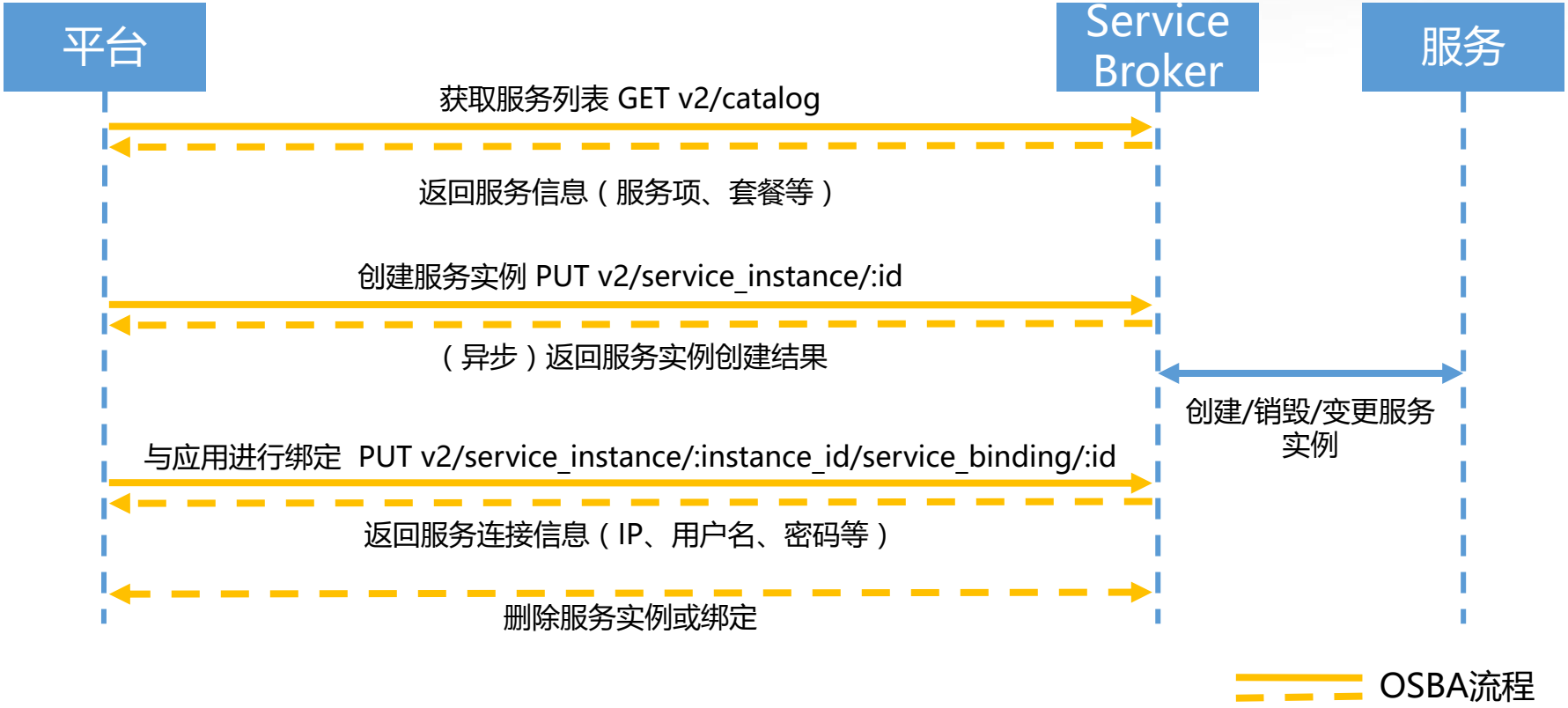
- Google Cloud/Deis
- OPEN ServiceBroker API
- 扩大使用范围

满足“12因子”应用中“应用数据应存储在后端服务中”

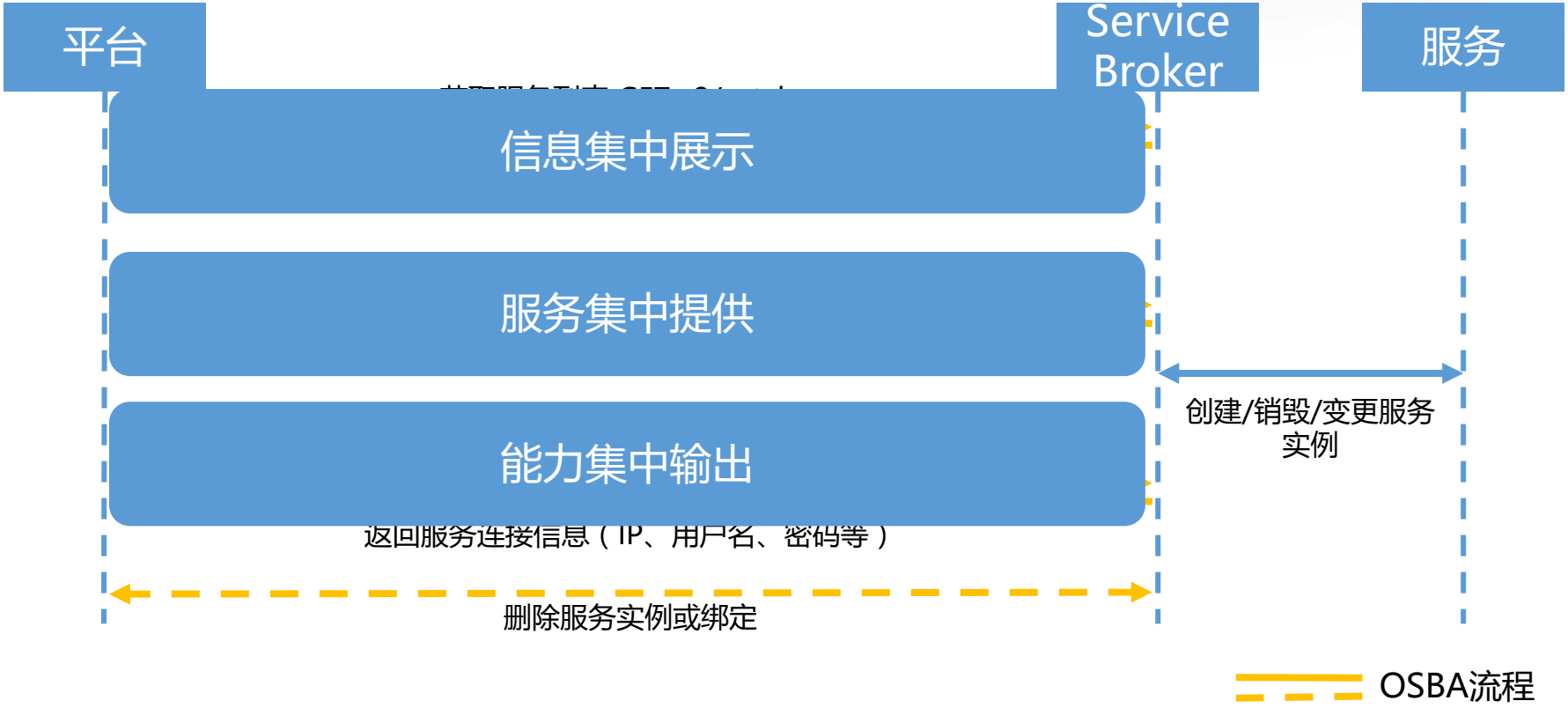
Open Service Broker API基本概念

- 应用平台（App Platform）
  - 应用托管平台（K8S、CF等），由应用平台承载的应用需要使用相关服务
- Service Broker
  - 平台与服务间的消息通道，用来管理服务平台
- 服务（Services）：
  - 由服务平台提供的能力列表，MySQL、Hadoop等
- 套餐（Plan）
  - 服务平台与服务最终用户间约定的服务标准、能力指标，例如空间、性能、安全等约束条件
- 服务实例（Service Instance）
  - 按照套餐约定向指定用户提供的服务实体，例如一个MySQL数据库，一个YARN资源队列

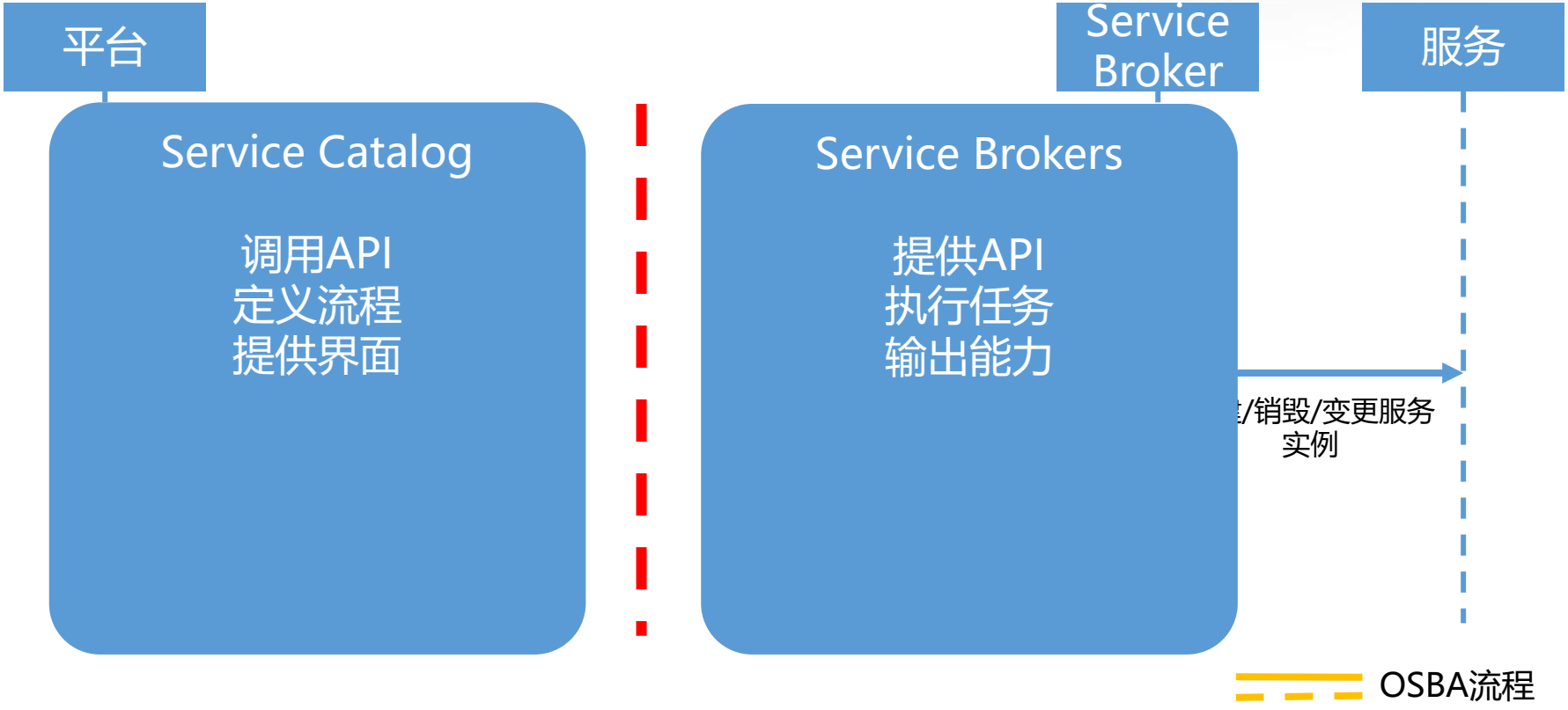
Open Service Broker API基本流程



Open Service Broker API基本流程



Open Service Broker API基本流程



目录

- 什么是Open Service Broker API
- 什么是Service Catalog
- 什么是Service Broker
- 服务能力中心的实践

Roadmap*

- RBAC default, Network Policy
- Stateful upgrades, GPUs
- Multi-workload scheduling
- Cloud providers, CRI
- Service catalog

Kubernetes1.6 and the Open Source Roadmap - Aparna Sinha

OSB API在K8S中的实现-Service Catalog

- 2016年9月成立Kubernetes Service Catalog SIG
- 最新版本0.0.6
- Deis/steward项目

service-catalog

build passing

Introduction

The service-catalog project is in incubation to bring integration with service brokers to the Kubernetes ecosystem via the Open Service Broker API. A service broker is an endpoint that manages a set of services. The end-goal of the service-catalog project is to provide a way for Kubernetes users to consume services from brokers and easily configure their applications to use those services, without needing detailed knowledge about how those services are created / managed.

0.0.6

arschles released this 7 days ago · 23 commits to master since this release

Version 0.0.6 is the sixth Alpha-quality release of the service catalog. Below are some highlights from this release:

- We've promoted third party resource storage for the API server to alpha. It is still in development, but we've amended the walkthrough to indicate how to use this storage mode
- We now only allow `Binding`s to `Instance`s that are ready and to `Instance`s that point to `ServiceClass`es that are marked as `bindable: true`
- We've made further investments into our test suites, including:
  - Go-based end-to-end tests running in Jenkins
  - Full integration test support for third party resource storage
  - Improved unit coverage for third party resource storage

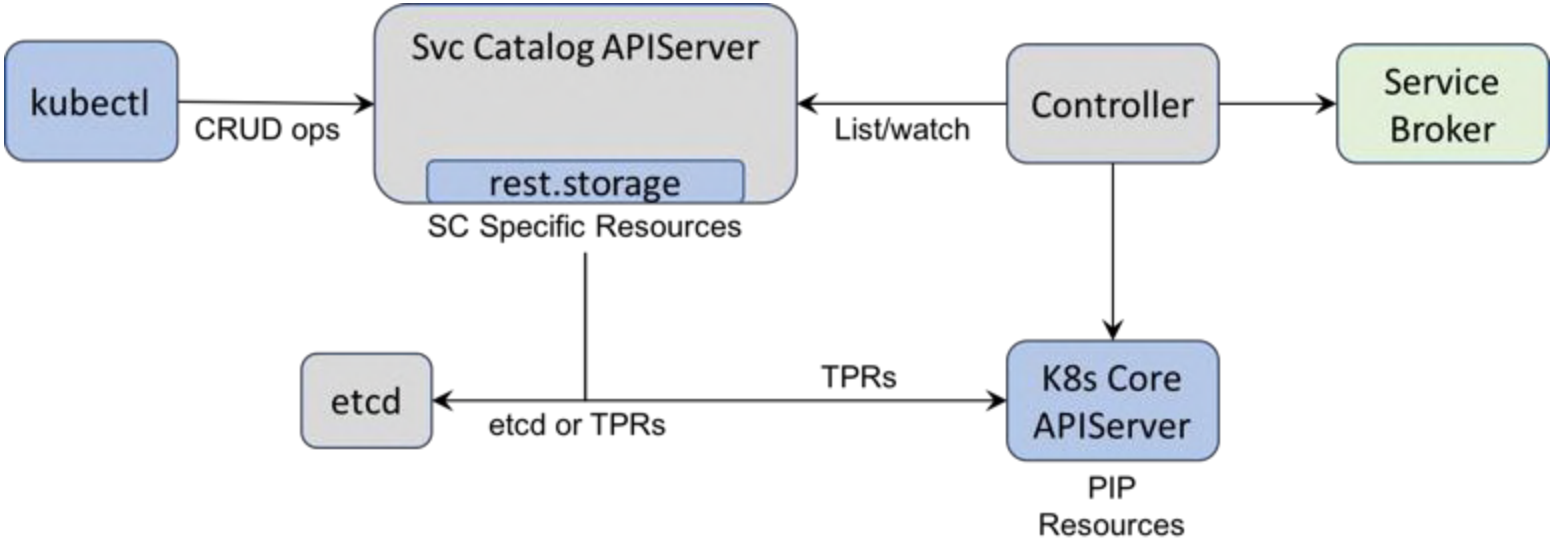
Steward

This repository contains the original Deis-built Proof of Concept for a Kubernetes-native Service Broker. Given broad interest in the Service Catalog space, Deis has joined forces with Google, Red Hat, IBM and others, under the Service Catalog special interest group.

Ongoing development is now happening in the service-catalog repository.

The original concepts which informed Deis' upstream involvement live in README-OLD.md.

ServiceCatalog设计



安装命令

$ helm install charts/catalog --name catalog --namespace catalog

注意设置几个参数

apiserver.image = quay.io/kubernetes-service-catalog/apiserver:canary
controllerManager.image = quay.io/kubernetes-service-catalog/controller-manager:canary
apiserver.storage.type = etcd
etcd_image = quay.io/coreos/etcd:latest

$ helm install charts/catalog --name catalog --namespace catalog --set key=value[,key=value]

https://github.com/kubernetes-incubator/service-catalog/tree/master/charts/catalog

ServiceCatalog基本概念

ServiceCatalog	OSB API	
Application		部署在K8S中的程序包
Binding	Binding	表示应用与服务实例之间的连接关系
Broker	Broker	用来管理一组服务的实体
Credentials		应用连接服务的鉴权信息
Instance	Service Instance	服务实例
Service Class	Service	通过Broker提供的服务能力列表项
Plan	Plan	套餐，用来标明服务特性的列表项

https://github.com/kubernetes-incubator/service-catalog/blob/master/docs/design.md

关于鉴权信息向应用的注入

- 借助于K8S新发布的PodPreset特新来管理鉴权信息的注入

```
kind: PodPreset
apiVersion: settings.k8s.io/v1alpha1
metadata:
  name: allow-database
  namespace: myns
spec:
  selector:
    matchLabels:
      role: frontend
  env:
    - name: DB_PORT
      value: 6379
    - name: duplicate_key
      value: FROM_ENV
    - name: expansion
      value: $(REPLACE_ME)
  envFrom:
    - configMapRef:
        name: etcd-env-config
  volumeMounts:
    - mountPath: /cache
      name: cache-volume
    - mountPath: /etc/app/config.json
      readOnly: true
      name: secret-volume
  volumes:
    - name: cache-volume
      emptyDir: {}
    - name: secret-volume
      secretName: config-details
```

- 将用户信息与系统信息分离
- 简化Binding设计
- 方便提供多种注入方式

目录

- 什么是Open Service Broker API
- 什么是Service Catalog
- 什么是Service Broker
- 服务能力中心的实践

Service Brokers是服务的中介

Service Catalog
统一
流程

OSBA

Service Broker
个性
执行

如何开发一个Broker

- 在CloudFoundry中给我们提供了足够的参考
  - https://github.com/cloudfoundry-community?language=&page=2&q=broker&type=&utf8=%E2%9C%93
- 当然也有更容易与K8S适配的方案
  - https://github.com/openshift/open-service-broker-sdk
- 如果你有一些Openshift的模板（openshift template）
  - 在openshift 3.6版本之后新增了一个template Service Broker
- 如果你使用ansible来自动化你的日常工作
  - https://github.com/fusor/ansible-service-broker

关于Service Catalog和Brokers的更多信息

- FROM Paul Morie @RedHat
- https://github.com/pmorie/catalog-links

目录

- 什么是Open Service Broker API
- 什么是Service Catalog
- 什么是Service Broker
- 服务能力中心的实践

传统服务交付效率不高

开发者

基础设施管理员

系统管理员

安全管理员

TASK	TIME (MINS)
准备服务器	30
准备存储和网络	30
工作任务排期	4 Days
安装操作系统	90
安装后检查	60
配置操作系统	120
安装应用运行时	180
配置应用程序	90
工作任务排期	5 Days
安全配置和扫描	270
工期	**12 hours**
人日	**10 Days**

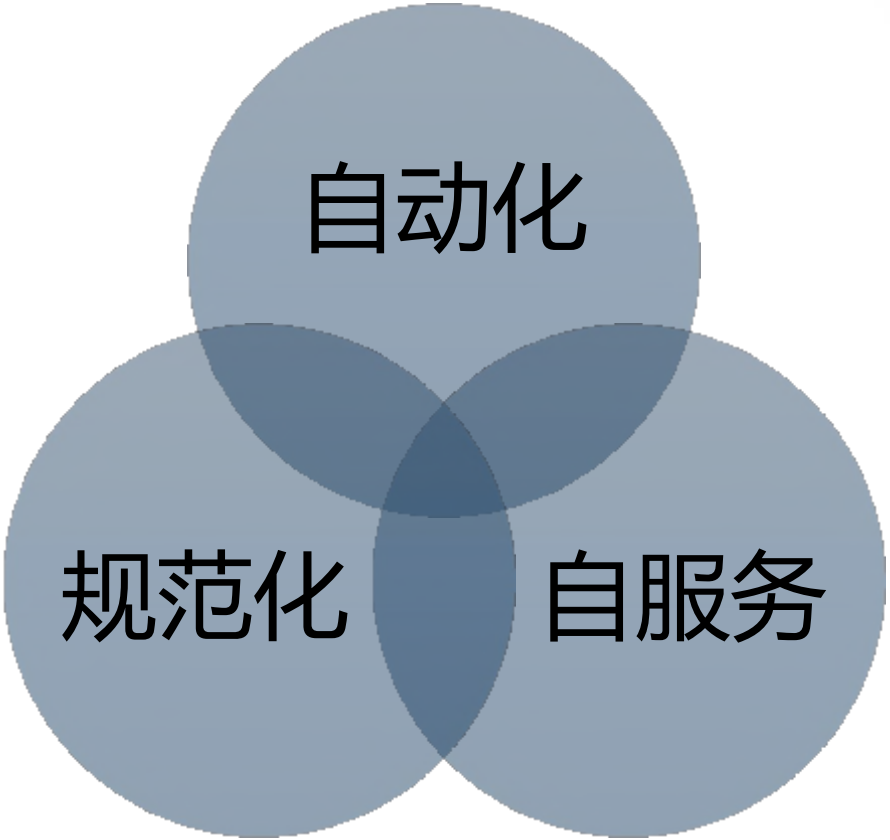
信息传递效率不高

需要交付的服务越来越多

存储 → 分析 → 预测

GBASE
APACHE STORM
mongoDB
PostgreSQL
H2O.ai
HBASE
Spark
Apache Solr
neo4j
MySQL
redis
kafka
TensorFlow
RabbitMQ
GREENPLUM DATABASE
cassandra
elasticsearch.

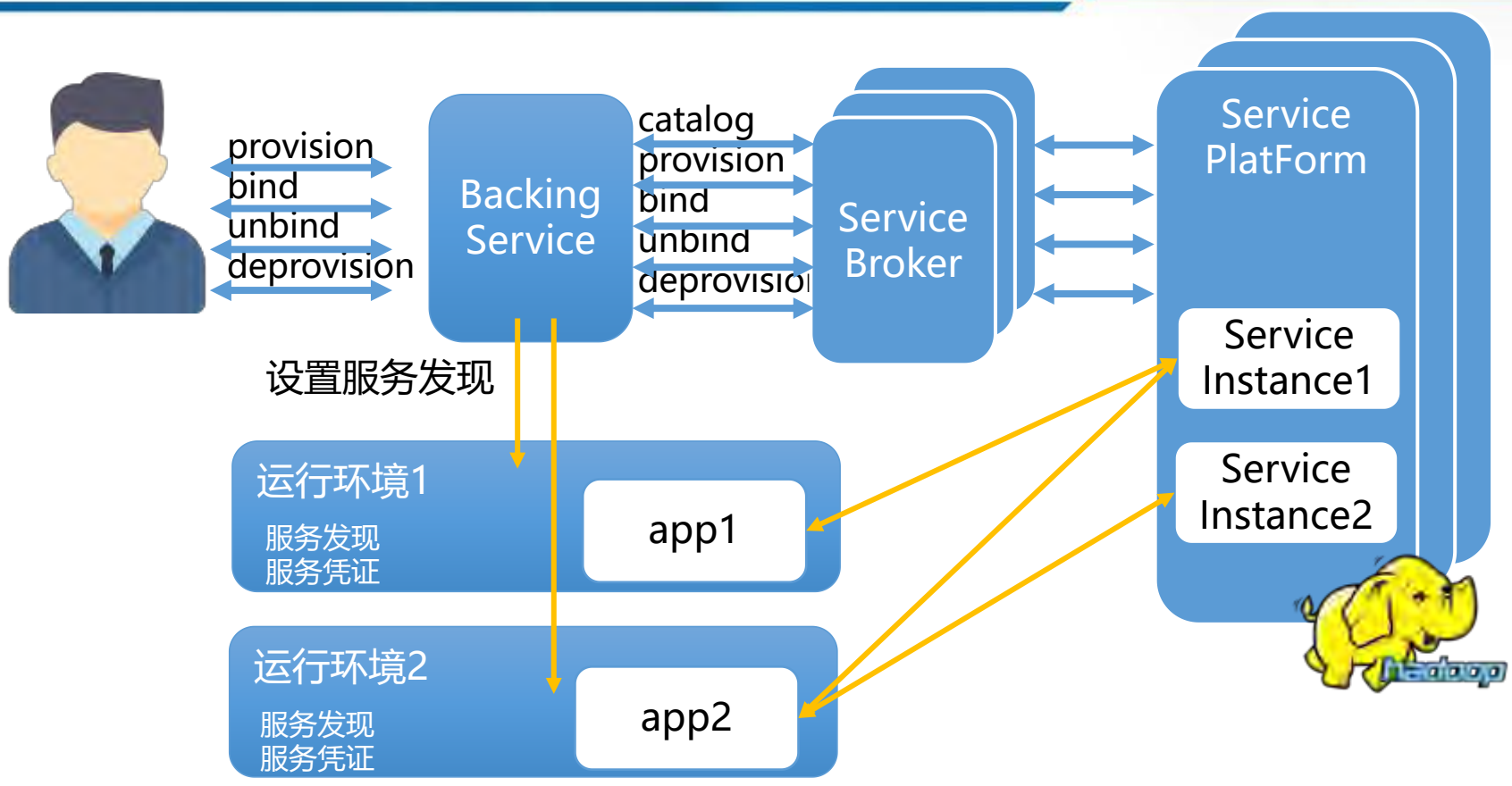
服务能力中心达成目标



为什么选择Open Service Broker API

- Service Broker API在CloudFoundry中的成功应用
  - BlueMix、pivotal.io通过Service Broker集成了众多服务
- 协议简单，易于实现，易于沟通
- 有众多适用于CF的Broker代码可以参考

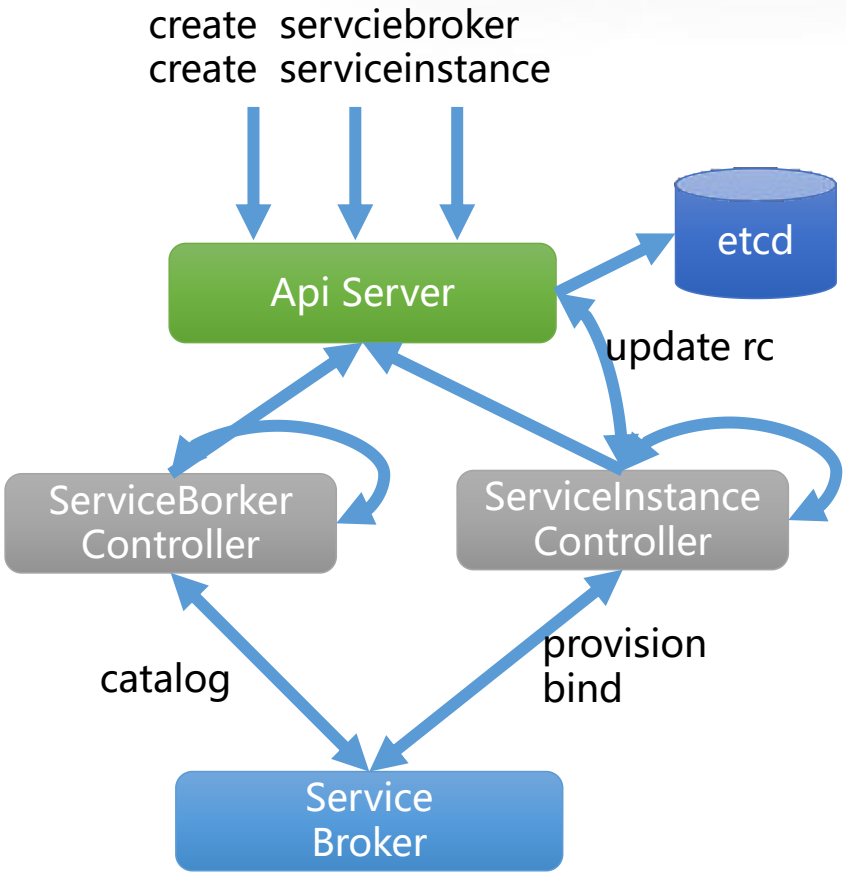
使用流程


provision
bind
unbind
deprovision
Backing Service
catalog
provision
bind
unbind
deprovisio
Service Broker
Service PlatForm
Service Instance1
Service Instance2
设置服务发现
运行环境1
服务发现
服务凭证
app1
运行环境2
服务发现
服务凭证
app2

新增K8S资源

- type Service struct {...}
- type Pod struct {...}
- type ReplicationController struct {...}
- …
- …
- type ServiceBroker struct {...}
- type BackingService struct {...}
- type ServiceInstance struct {...}

- RunNodeController()
- RunScheduler()
- RunReplicationController()
- …
- …
- RunServiceBrokerController()
- RunServiceInstanceController()

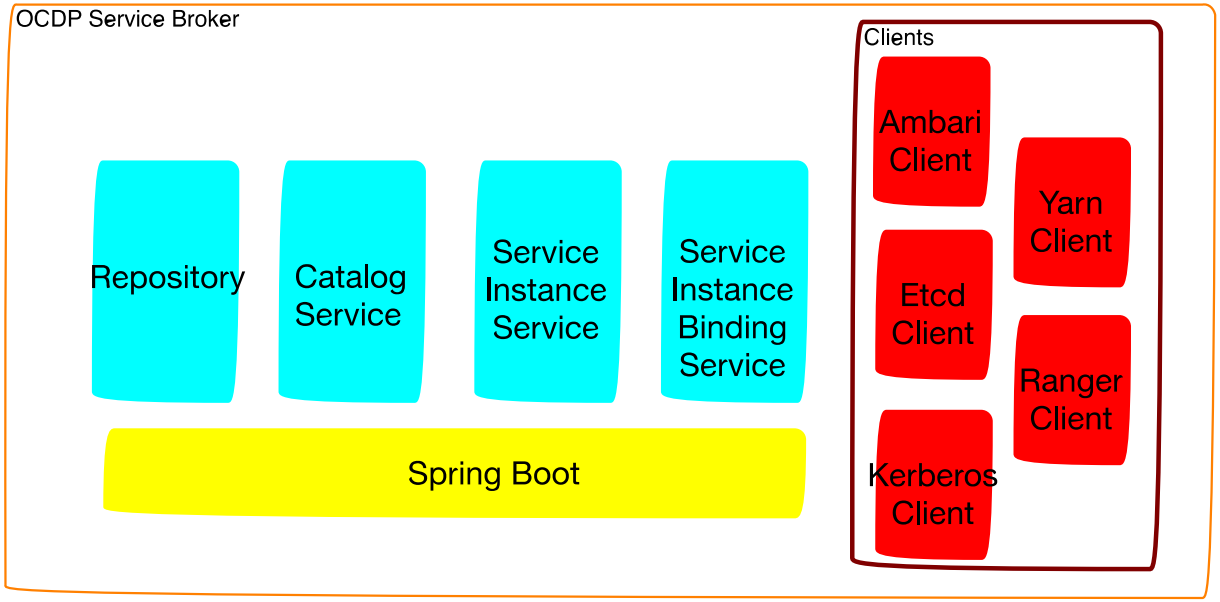


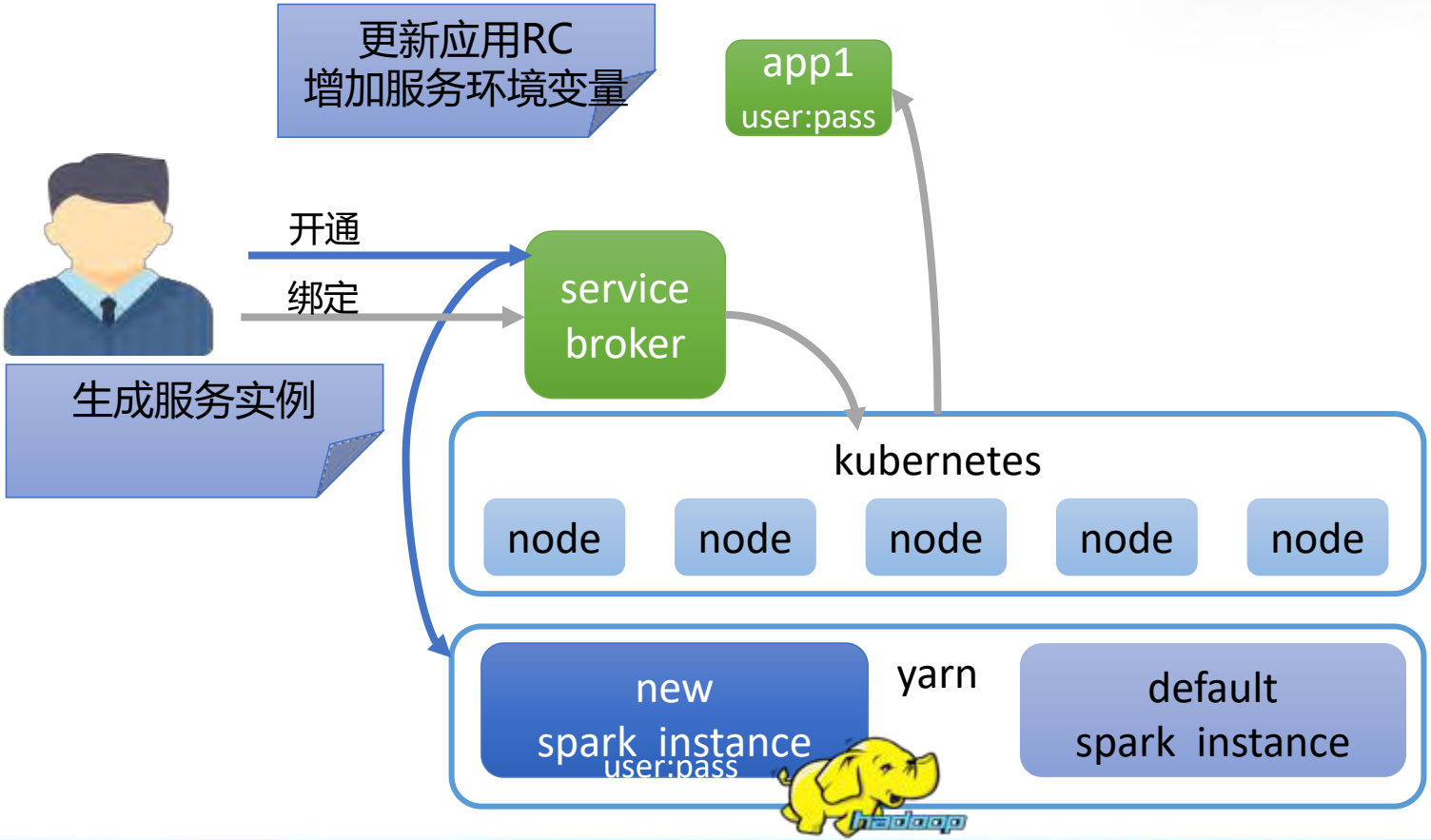
Controller调用ServiceBroker API

- ServiceBrokerController
  - Fetch catalog
    (GET /v2/catalog)

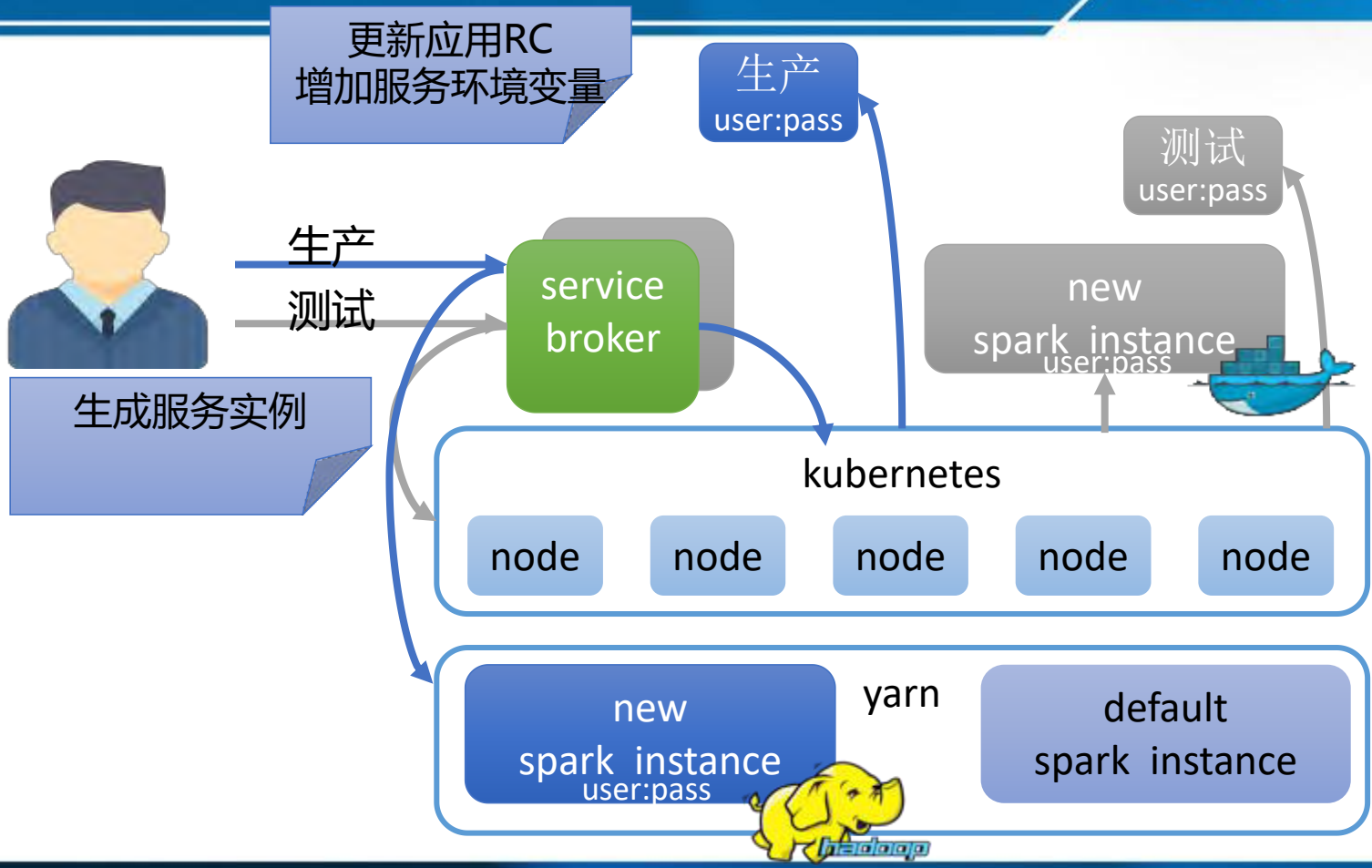
- ServiceInstanecController
  - Provision instance
    (PUT /v2/service_instances/:id)
  - Creating binding
    (PUT /v2/service_instances/:id/service_bindings/:id)
    - Update RC
  - Remove binding
    (DELETE /v2/service_instances/:id/service_bindings/:id)
    - Update RC
  - Remove instance
    (DELETE /v2/service_instances/:id)

大数据服务ServiceBroker实现

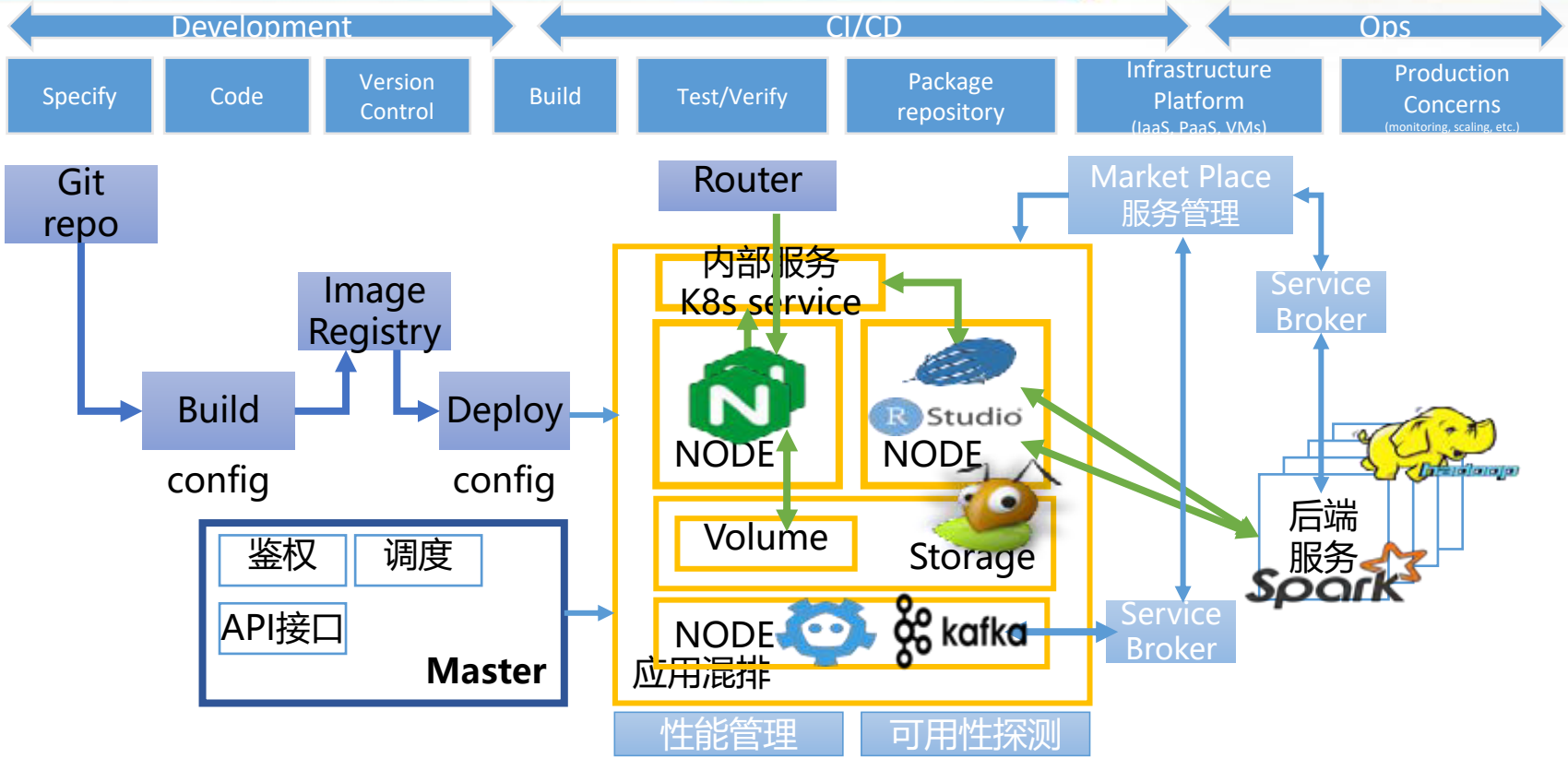



更新应用RC
增加服务环境变量
app1
user:pass
开通
绑定
service
broker
生成服务实例
kubernetes
node
node
node
node
node
new
spark instance
user:pass
yarn
default
spark instance
hadoop

应用与大数据服务编排


更新应用RC
增加服务环境变量
生产
user:pass
测试
user:pass
生产
测试
service
broker
new
spark instance
user:pass
生成服务实例
kubernetes
node
node
node
node
node
yarn
new
spark instance
user:pass
default
spark instance
hadoop

大数据应用和常规应用的混合编排



对Open Service Broker API的加强

- 及时反馈连接信息
  - 在服务创建时即返回连接信息，方便应用调试和初始化
- 提供自定义资源申请能力
  - 在Plan中增加自定义选项，满足用户个性化要求

ServiceCatalog在某运营商省公司中的应用（1）

以Storm的编排举例，https://github.com/asiainfoLDP/storm-openshift-orchestration

- kubernetes-nimbus-rc.yaml
- kubernetes-nimbus-service.yaml
- kubernetes-supervisor-rc.yaml
- kubernetes-ui-rc.yaml
- kubernetes-ui-service.yaml

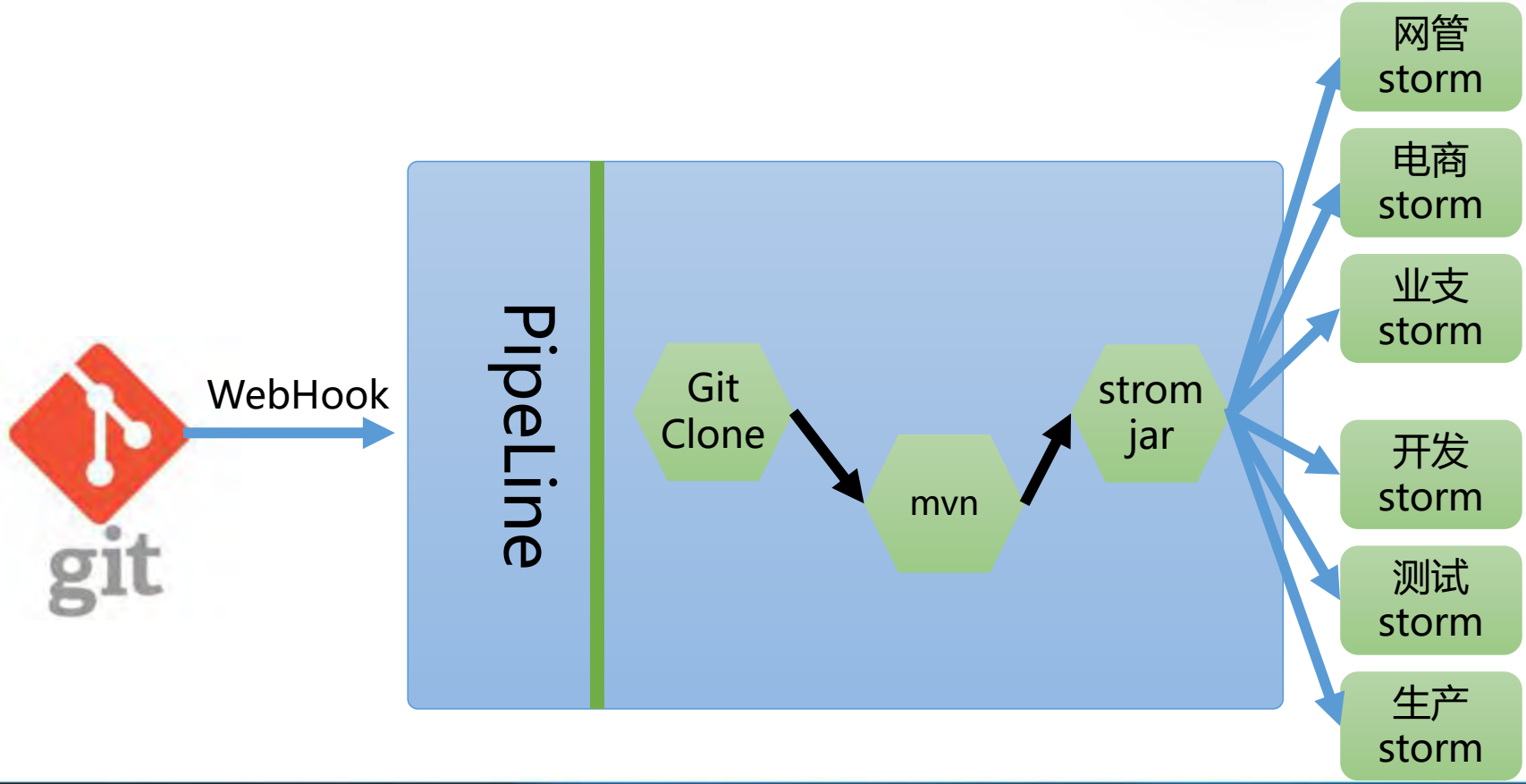
```
oc create -f kubernetes-nimbus-rc.yaml
oc create -f kubernetes-supervisor-rc.yaml
oc create -f kubernetes-ui-rc.yaml
```

```
oc create -f kubernetes-nimbus-service.yaml
reate -f kubernetes-ui-service.yaml
```

oc new-instance storm-cluster --service=Strom --plan=standalone

```
[root@rhel73 ~]# oc get pods
NAME                                    READY     STATUS    RESTARTS   AGE
sb-b6dugiiqbccme-kafka-d89ay            1/1       Running   0          20h
sb-b6dugiiqbccme-kafka-n3fjo            1/1       Running   0          20h
sb-b6dugiiqbccme-kafka-prsbk            1/1       Running   0          20h
sb-b6dugiiqbccme-zk-1-52t0u             1/1       Running   0          20h
sb-b6dugiiqbccme-zk-2-x7rml             1/1       Running   0          20h
sb-b6dugiiqbccme-zk-3-blupr             1/1       Running   0          20h
sb-fqlawbicqfaca-redis-j55iv            1/1       Running   1          4d
sb-fqlawbicqfaca-redis-master-2u5t4     2/2       Running   0          4d
sb-fqlawbicqfaca-redis-sentinel-rqy7w   1/1       Running   0          4d
sb-fqlawbicqfaca-redis-sentinel-z42iv   1/1       Running   0          4d
sb-fqlawbicqfaca-redis-u5gqi            1/1       Running   1          4d
sb-j2tvgkmujisze-stormnb-ngy90          1/1       Running   0          21h
sb-j2tvgkmujisze-zk-1-38rna             1/1       Running   0          22h
sb-j2tvgkmujisze-zk-2-bqvjn             1/1       Running   0          22h
```

ServiceCatalog在某运营商省公司中的应用（2）



ServiceBroker在某大型集团云平台项目中的应用(1)

HDFS

申请实例

Hive

申请实例

Kafka

申请实例

MapReduce

申请实例

RabbitMQ

申请实例

Redis

申请实例

Spark

申请实例

Storm

申请实例

ServiceBroker在某大型集团云平台项目中的应用(2)

谢谢