



Cloud Foundry 新增功能

Jennifer Hickey



Jennifer 简介

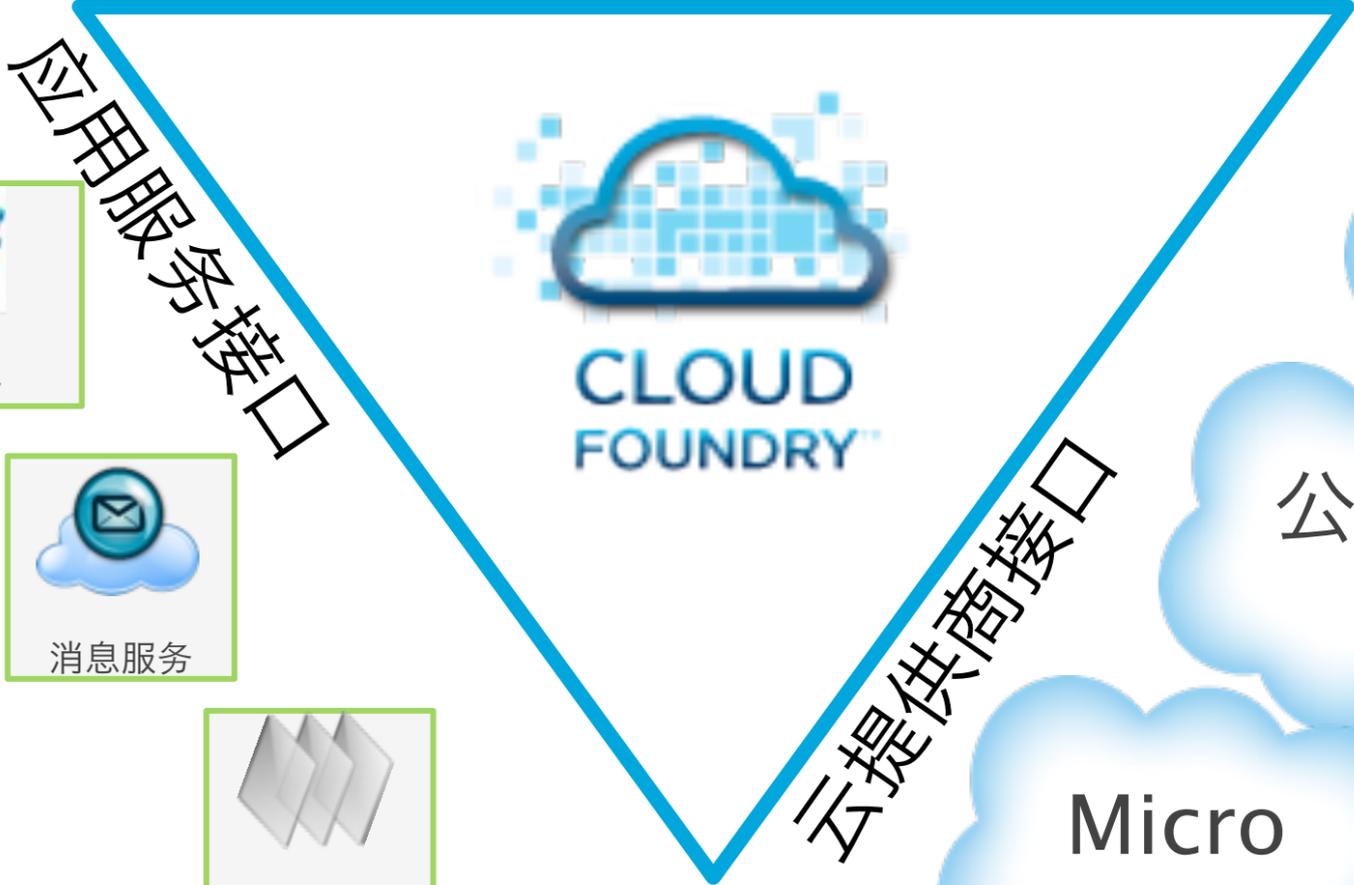
- Cloud Foundry 工程师
- 专注于框架支持
- 在 SpringSource 方面具有多年经验
- 参与过多个 Spring 和 SpringSource 项目
- 热衷于提升开发人员在云中的工作效率。

联系信息

Twitter : @jencompgeek

电子邮件 : jhickey@vmware.com

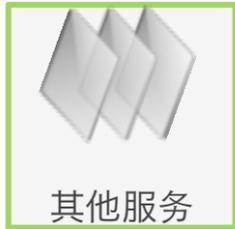
Cloud Foundry 开源 PaaS



数据服务



消息服务



其他服务

私有云

公共云

Micro Cloud Foundry

github
SOCIAL CODING



Apache2 license

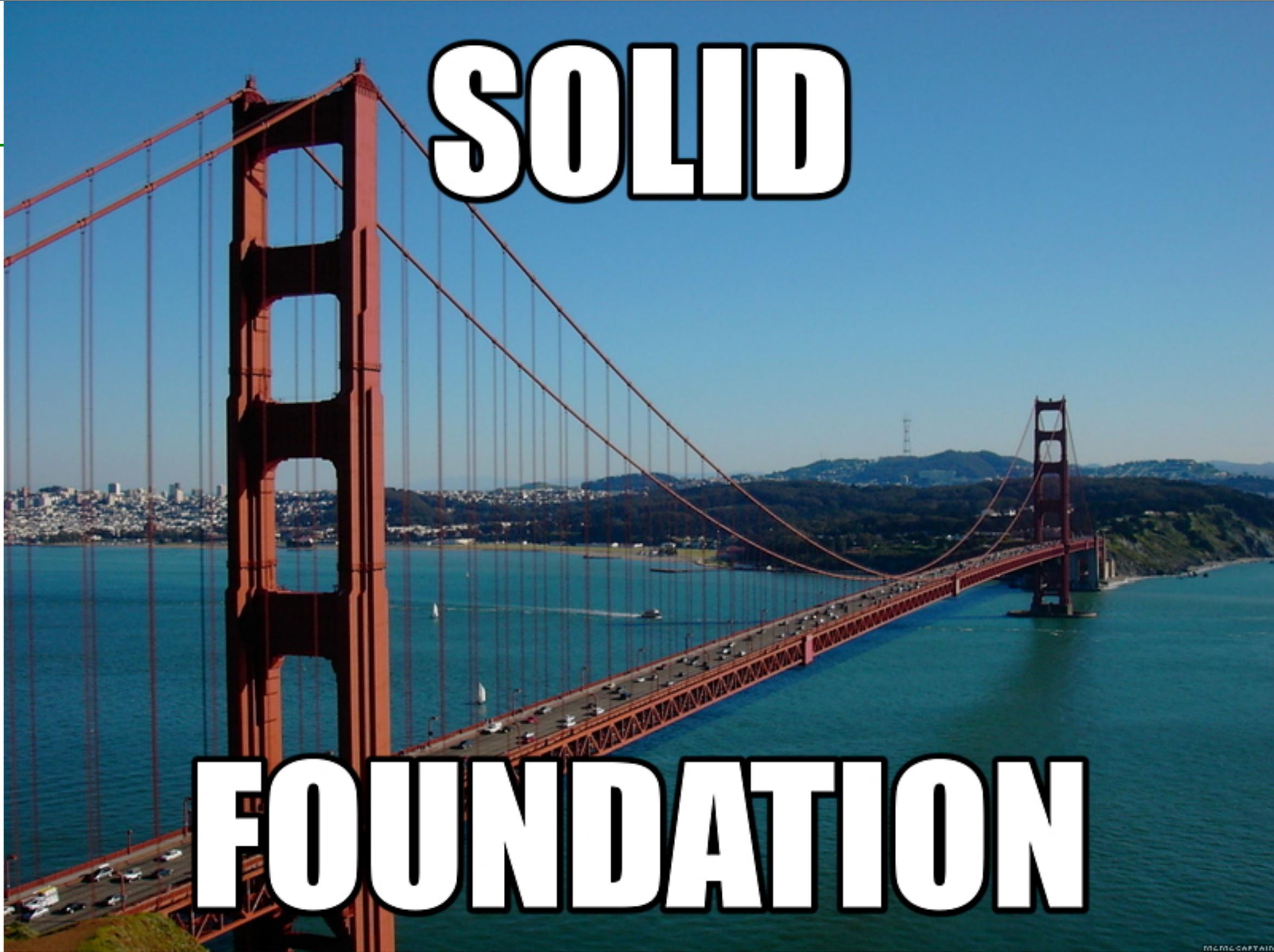
Cloud Foundry 意味着...

- 开源 BOSH
 - 用于进行发布、部署和生命周期管理的工具链
 - 用于部署 cloudfoundry.com
- 新的开源贡献流程
- 更多合作伙伴



自去年以来...

主题



SOLID

FOUNDATION

MEMOCAPTAIN



DISTRIBUTED



APPS

ENTERPRISE



READY

提纲

- 框架和运行时
- 安全性和帐户管理
- 服务
- 工具
- Micro Cloud Foundry
- CloudFoundry.com 商业化

框架和运行时

核心运行时和框架

- Java
 - Spring
 - Grails
 - Lift
 - Play
- Ruby
 - Rails
 - Sinatra
 - Rack
- Node.js

社区和合作伙伴贡献

- Python
 - Django
 - WSGI
- Erlang OTP/Rebar
- PHP
- Perl
- .NET

白金原则

以框架所需的方式
对其进行处理！



通用框架支持

- 针对应用程序管理的命令行
 - 创建应用程序和服务
 - 更新绑定、内存等
 - 扩展实例
- 通过环境变量访问服务和应用程序信息
 - 服务主机、端口、凭据
 - 应用程序 IP 和端口
- 运行时库

通用框架支持

- 对典型应用程序的自动重新配置
- 通过 Caldecott 访问服务
 - 针对本地客户端创建隧道
 - Mysql、Mongo、Redis CLI 等
- 清单支持

独立程序：“无框架”框架

- Cloud Foundry 现在支持独立应用程序！
- 选择运行时并提供启动命令

```
$ vmc push myapp
Detected a Standalone Application, is this correct? [Yn]:
1: java
2: node
3: ruby18
4: ruby19
Select Runtime [java]:
Start Command: java -jar myapp.jar
```

独立应用程序的用途

- 运行工作者进程
- 将框架与嵌入式 Jetty 或 Netty 结合使用
 - Unfiltered
 - Spray
 - vert.x

Cloud Foundry 上的 Scala

- Lift

新增功能

- Play
- 引入您自己的容器 (Web 应用程序)
 - Scalatra
 - Blue Eyes
 - Unfiltered
 - Spray
- 独立支持

Play 框架

- 支持 Play 2.0
- 将 Play 应用程序推送到 CloudFoundry.com

```
$ play dist  
$ vmc push --path=dist/dist.zip
```

- 自动应用演进
- 自动将数据库切换到 Postgres 或 MySQL
- 属性在服务连接的 CF 环境中可用
–`-${?cloud.services.mongo.connection.host}`

演示

通过 Cloud Foundry 乐享 Play !

Scala 支持：即将推出

- 针对 cloudfoundry-runtime 库的 Scala 打包程序
- 针对 Cloud Foundry 部署的 SBT 插件
 - 已具有 maven 插件
- giter8 模板
 - 用于配置 Web 端口和 Mongo 的 Blue Eyes
 - 其他？
- 其他针对Play的自动重新配置
- 针对 cloudfoundry-client-lib 的 Scala 打包程序

- Node 0.4 和 0.6 运行时

新增功能

- Node 0.8
- NPM 支持
 - 从 `npm-shrinkwrap.json` 下载并安装模块
 - 针对 `git` 模块的全新支持
 - 不论本地如何打包，具有本地依赖的模块均正确安装

Node.js : 自动重新配置和 cf-runtime

- 使用通用 Node 模块的应用程序自动连接到 CF 服务
 - 非常适合现有应用程序在 CF 上快速部署
- 用于更为精细地控制 svc 连接的 cf-runtime 模块
 - 用于获取应用程序/服务属性和连接的方法

```
// Obtain connection properties for single service of type Redis
app.serviceProps.redis
// Obtain connection properties for service named 'redis-service-name'
app.serviceProps['redis-service-name']
// Obtain the list of service names of specific type
app.serviceNamesOfType.redis
// Check if service of the given type is available
cf.RedisClient !== undefined
// Connect to a single service of type Redis
var redisClient = cf.RedisClient.create()
// Connect to redis service named 'redis-service-name'
var redisClient = cf.RedisClient.createFromSvc('redis-service-name')
```

Node 支持：即将推出

- ???

Cloud Foundry 上的 Java



- 支持 Java 6 的 Spring、Grails、Java Web 应用程序

- 自动重新配置 Spring 和 Grails
- 新增功能**

- Java 7
- 独立支持
 - Spring 集成 和 Batch 工作者进程
 - vert.x
 - Tomcat 7
 - Clojure、Groovy 脚本等

演示

使用 Spring 在 Cloud Foundry 上
运行工作者进程

Java 支持：即将推出

- 自动重新配置独立 Spring 应用程序
- 选择 Java Web 应用程序的容器
 - 即 Tomcat 6 或 Tomcat 7
- ???

运行时和框架：即将推出

- 运行时降级策略
- 更频繁地升级运行时
- 更频繁地添加新的运行时
- 直接从 nginx 响应静态内容
- 所有应用程序的 TMPDIR 集

新增功能

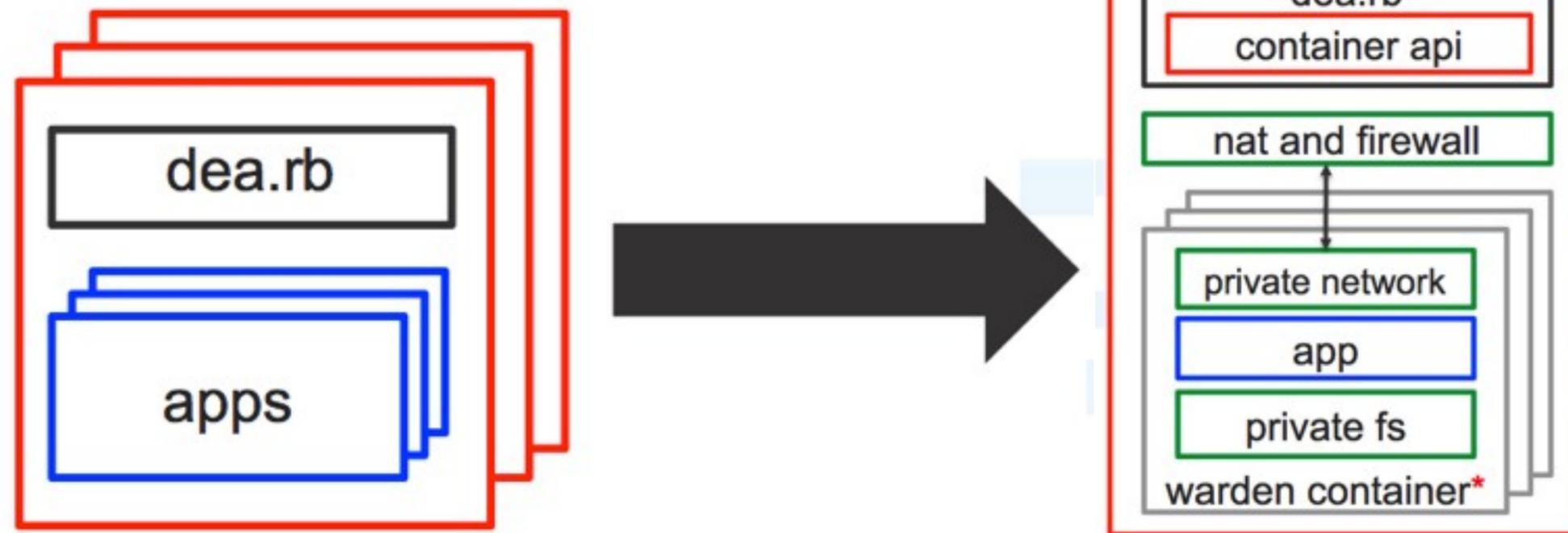
安全性和帐户管理

下一代云控制器

- 使用 Sinatra/Sequel 重写了云控制器
- 更多基于 REST 的交互
 - 通过有效负载的超链接进行导航
- 在一次调用中拉取所有信赖对象的能力
 - 减少所需的往返时间

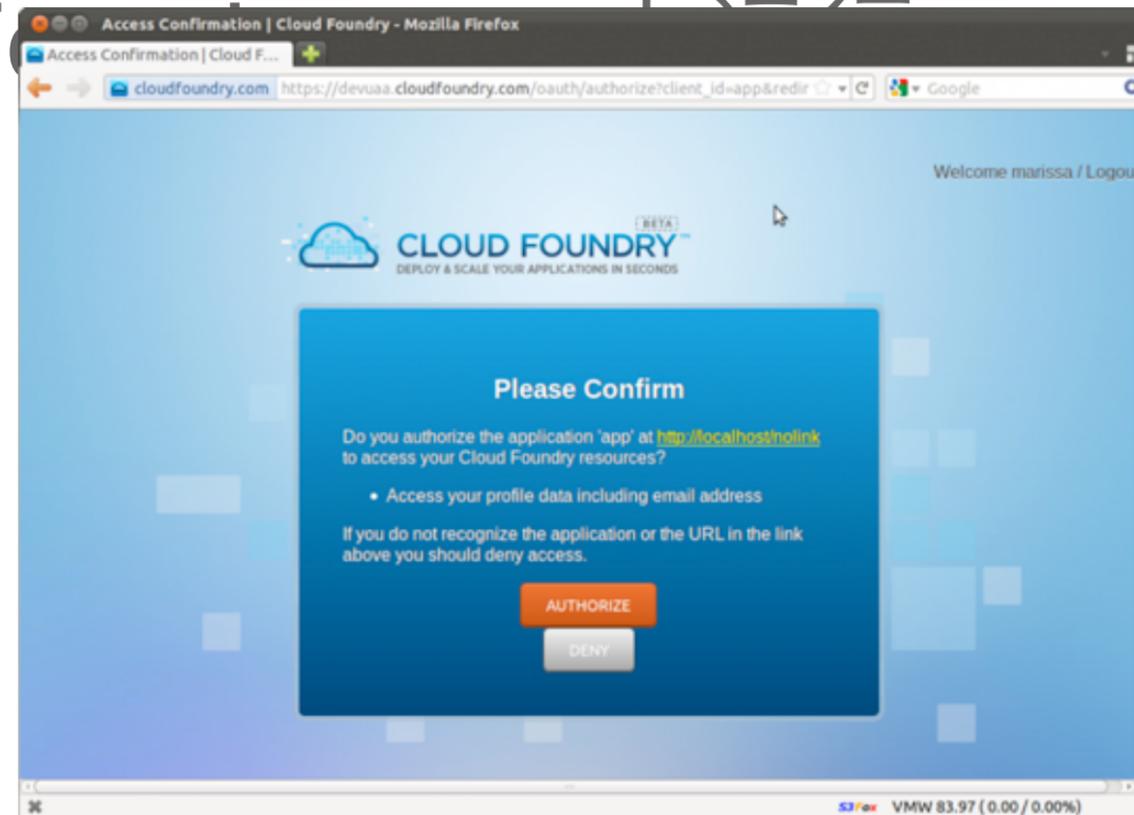
下一代DEA

- 使用“看守（warden）”提供用于应用程序执行的容器
 - 使用 Linux cgroups
 - 主要目标：隔离和资源控制
 - 隔离 CPU 使用、内存使用、磁盘使用



UAA (用户帐户和身份验证)

- 使用 Spring 和 Spring Security 构建的 Web 应用程序
- OAuth2
- 可加插件的身份验证提供者模型
- 现在在 CloudFoundry 上运行



UAA 高级特征

- 集中身份管理
- 单一登录
- 委派服务访问权
- 用户帐户管理
- 客户端应用程序注册

新增功能

服务

Cloud Foundry 服务

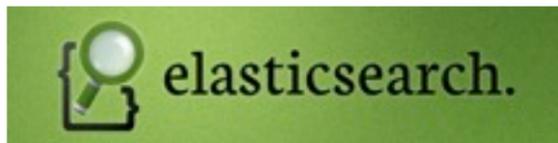
- CloudFoundry.com



PostgreSQL



- github.com/cloudfoundry



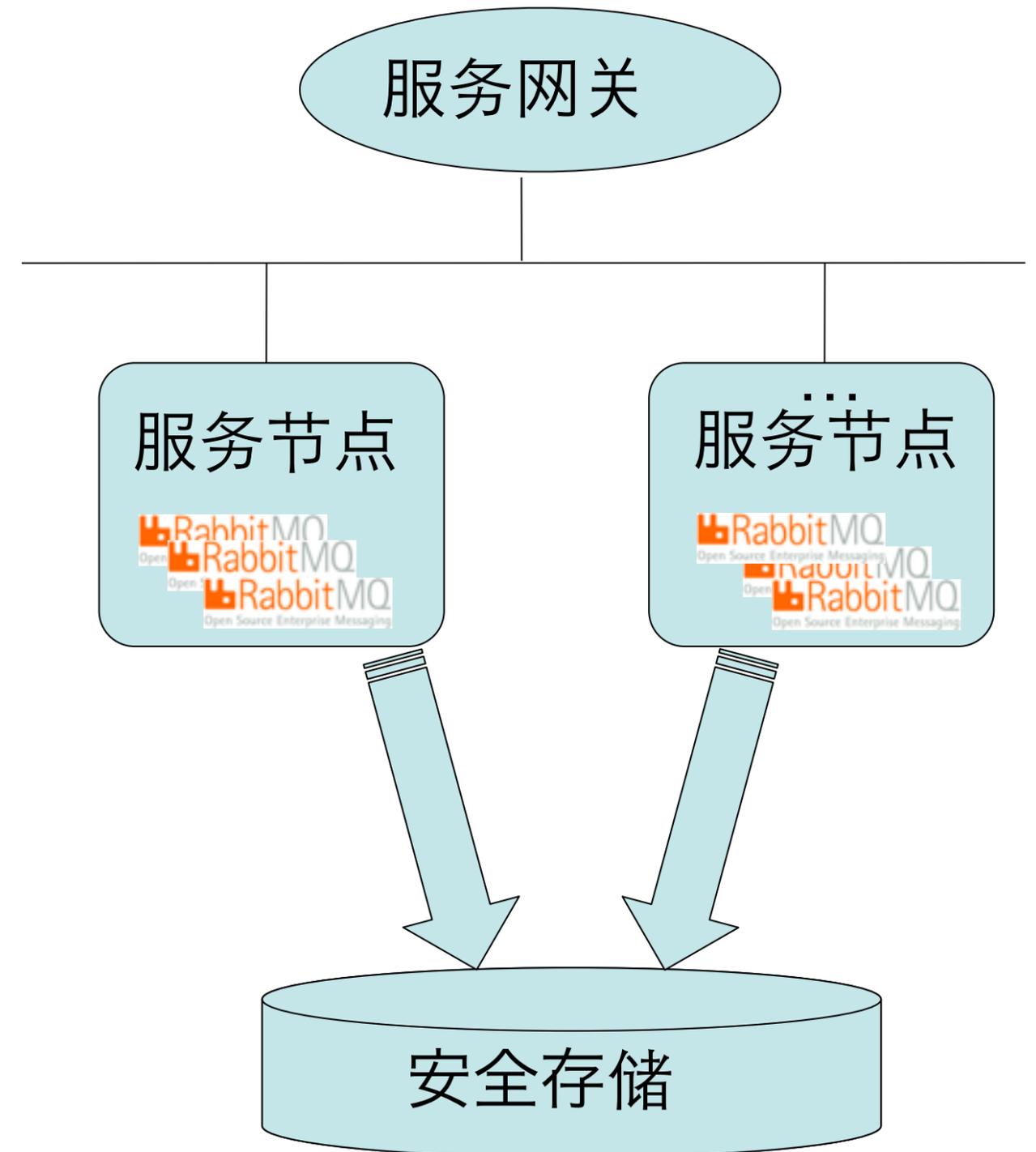
文件系统

Blob



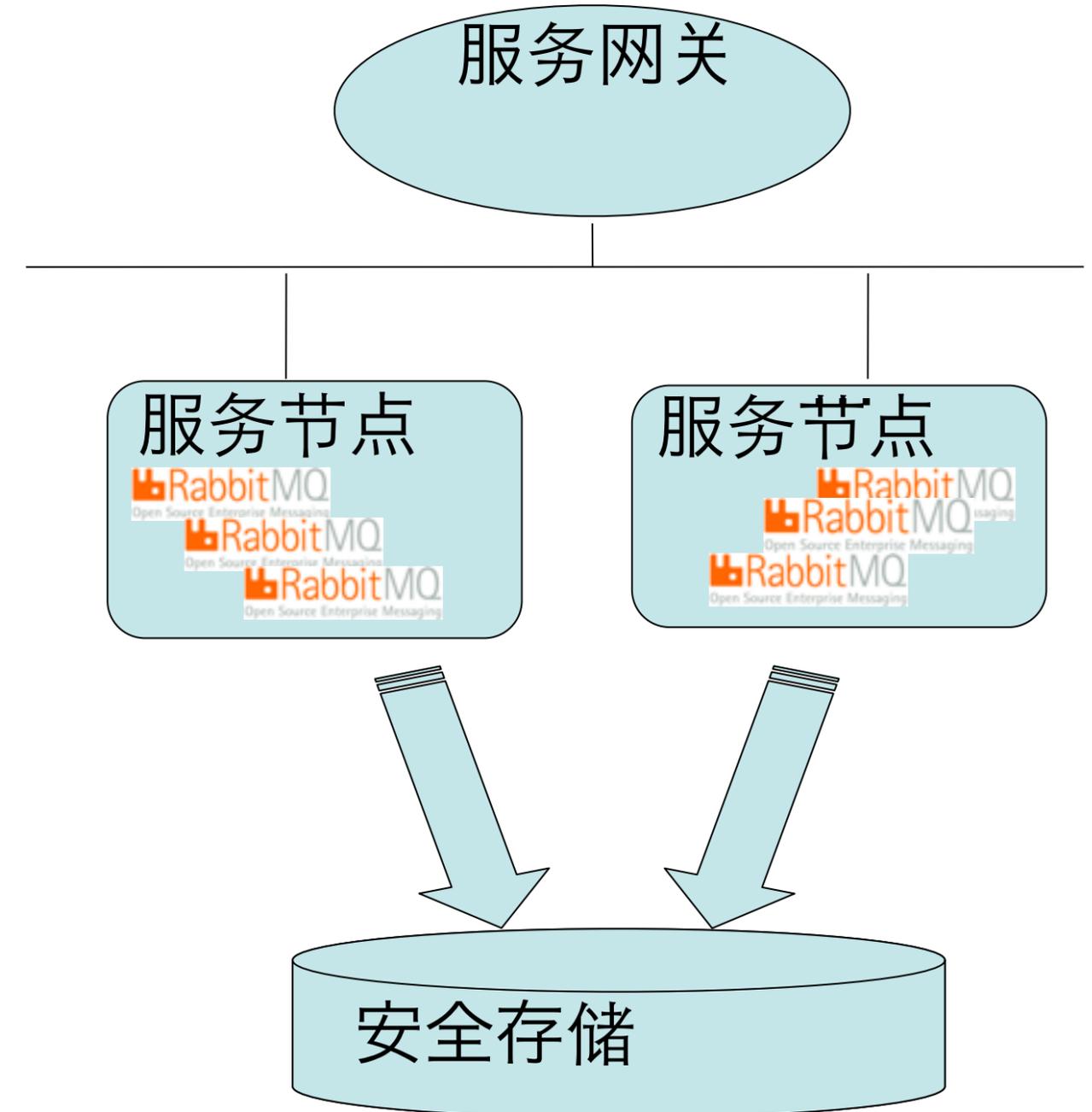
备份

- 定期将数据转储到安全存储
 - 使用服务实用程序处理事务等
- 自动循环
- 对用户不可见
- 现在在 cloudfoundry.com 上运行



快照

- 用户可见的服务实例上传/下载
 - 创建快照
 - 枚举快照
 - 下载快照
 - 上传快照（本地、云到云）
 - 切换到某一快照
- 即将在 CloudFoundry.com 推出



Warden

- 在沙盒中运行每个服务以防止其免受服务安全缺陷的影响
- Warden：“cgroups” 顶端的客户端/服务器
- 服务基础库有助于轻松地“看守”新服务
- 即将在 CloudFoundry.com 推出



版本

- 服务网关向控制器公布其服务的不同版本
- 服务 Node 向网关公布其支持的版本
- 用户选择所需的版本
- 基于请求版本的网关路由配置（类似于计划）
- 在 CloudFoundry.com 上正在进行

新增功能

工具

vmc-ng 简介

- 下一代 VMC, Cloud Foundry 的 CLI
- 多个交互和体系结构改进
- 仍在开发中
- `gem install vmc --pre`
- `mkdir ~/.vmc/use-ng`

vmc-ng : 改进的交互

- 更短的问题描述
- Tab 自动补全
- 后退
- 更好的脚本编写支持
- 更好的进度指示器

```
mightymouse:tentd-admin jencompgeek$ vmc push
Name> tentd

Instances> 1

1: rack
2: other
Framework> rack

1: ruby18
2: ruby19
3: other
Runtime> 2

URL> tentd.jencompgeek.cloudfoundry.me

1: 64M
2: 128M
3: 256M
4: 512M
5: 1G
Memory Limit> 128M

Creating tentd... OK
```

vmc-ng : 改进的体系结构

- 小核心委派给不同的高级别库
 - 高级别 = "cfoundry" gem
 - cfoundry 提供对象层, 而非解析的 JSON

- 扩展的插件体系结构

- 使 vmc 代码库保持

- 示例

- caldecott
 - 清单

```
desc "delete [APP]", "Delete an application"
flag(:really) { |name|
  force? || ask("Really delete #{c(name, :blue)}?", :default => false)
}
def delete(name)
  return unless input(:really, name)

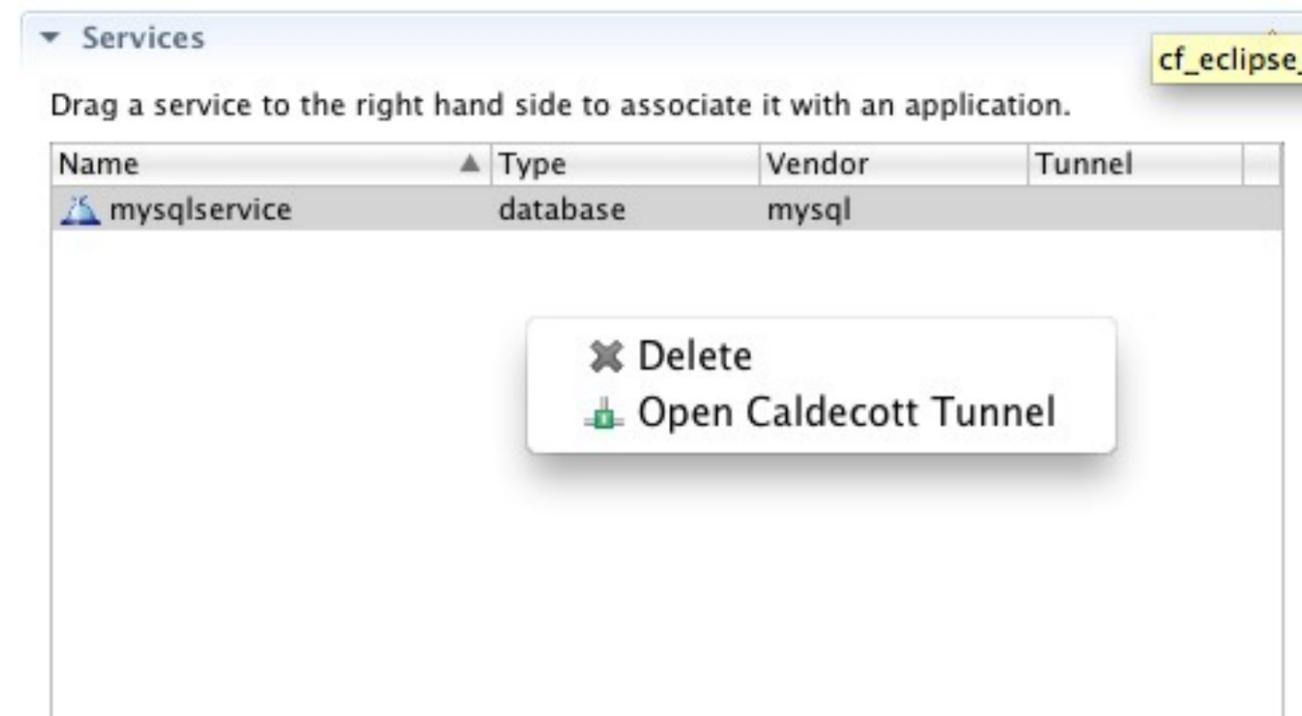
  with_progress("Deleting #{c(name, :blue)}") do
    client.app(name).delete!
  end
ensure
  forget(:really)
end
```

STS

- 部署并调试 Java Web、Spring、Grails、Lift 应用程序

新增功能

- 部署并调试独立 Java 应用程序
 - 包括 Scala 和 Groovy
- Caldecott 支持
 - 直接从 STS 建立通往所有服务的隧道



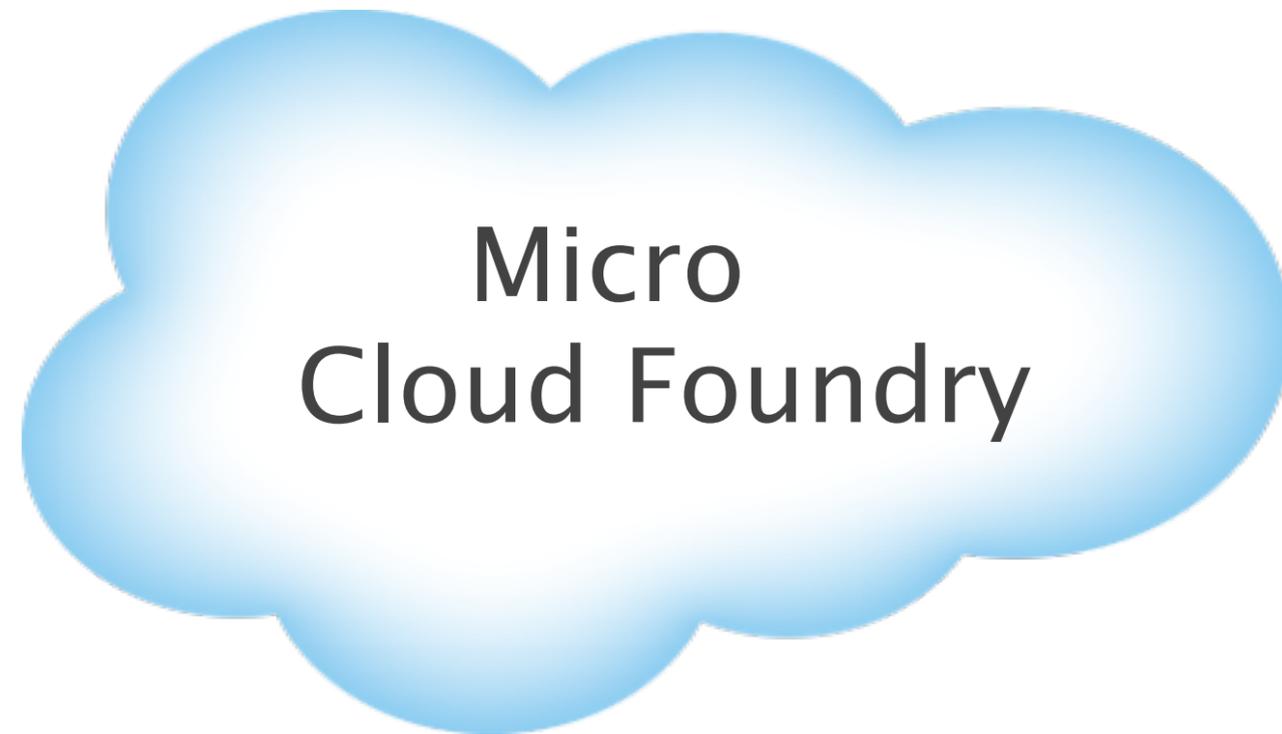
演示

使用 STS 部署独立应用程序

工具箱的其他工具

- Java
 - cloudfoundry-client-lib
 - 云控制器 REST API 的 Java 客户端
 - cloudfoundry-maven-plugin
 - CF Gradle 插件
 - cloudfoundry-caldecott-lib
 - 建立通往服务的隧道的 Java 库
 - cloudfoundry-runtime
- Node
 - cf-runtime 模块

新增功能



Micro Cloud Foundry

- 使用虚拟机映像在个人计算机上运行 CF 的完整实例
- 最新发行版 – 最新版本和最新的 cf.com 一致

即将推出

- 改进了更频繁推出发行版的流程
- 更新机制
- 用于配置管理的 REST API
 - 之前在 MCF 控制台中可用的功能

CloudFoundry.com 商业化应用

CloudFoundry.com 即将完成Beta测试

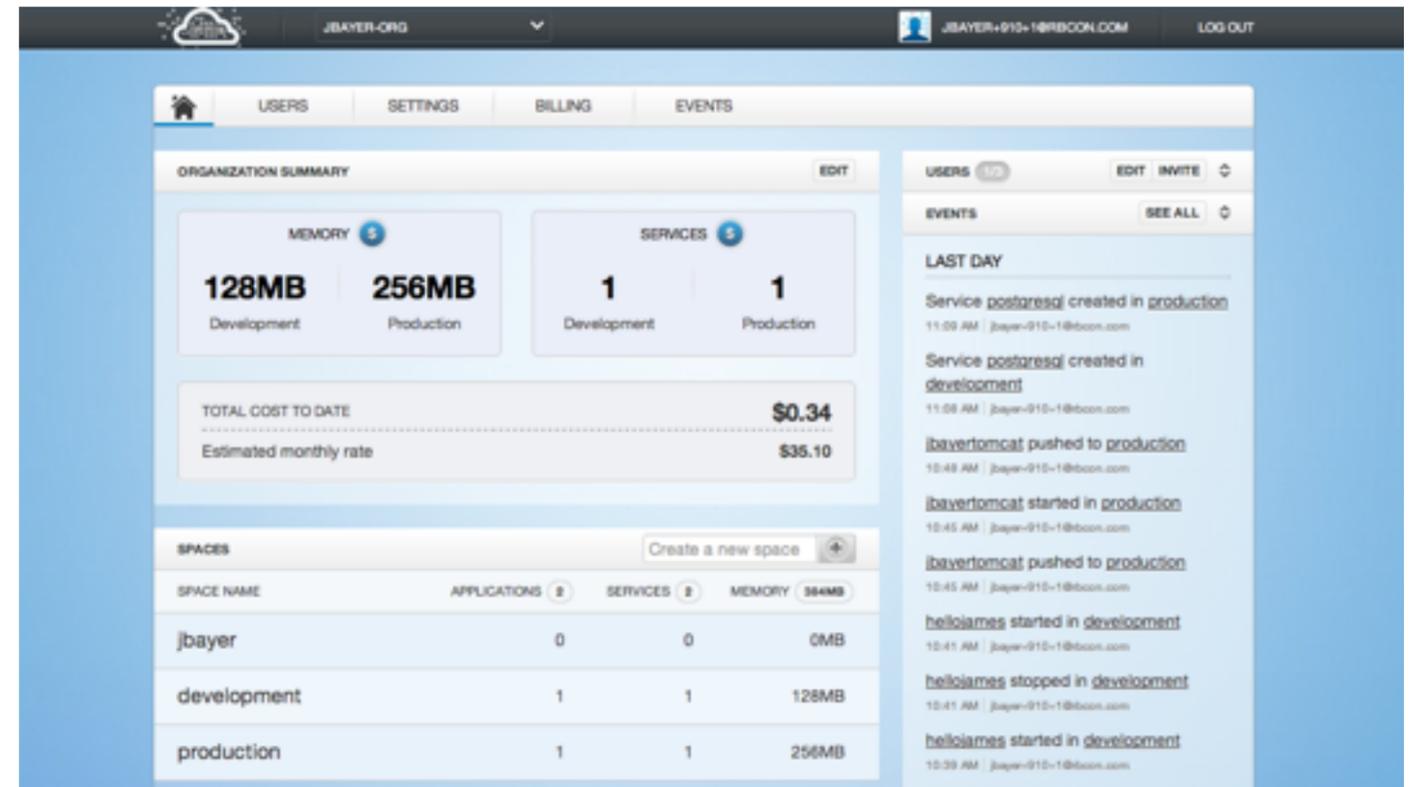
- CloudFoundry.com 商业化应用近在咫尺
- 目标：自给自足团队 3-20 人
 - 专为应用程序的协同操作而设计
 - 包括开发人员、架构师、管理员、QA、操作员等
- 付费层提供
 - 额外的容量和功能
 - 信用卡支付
- 保留免费层

创建、协作、控制

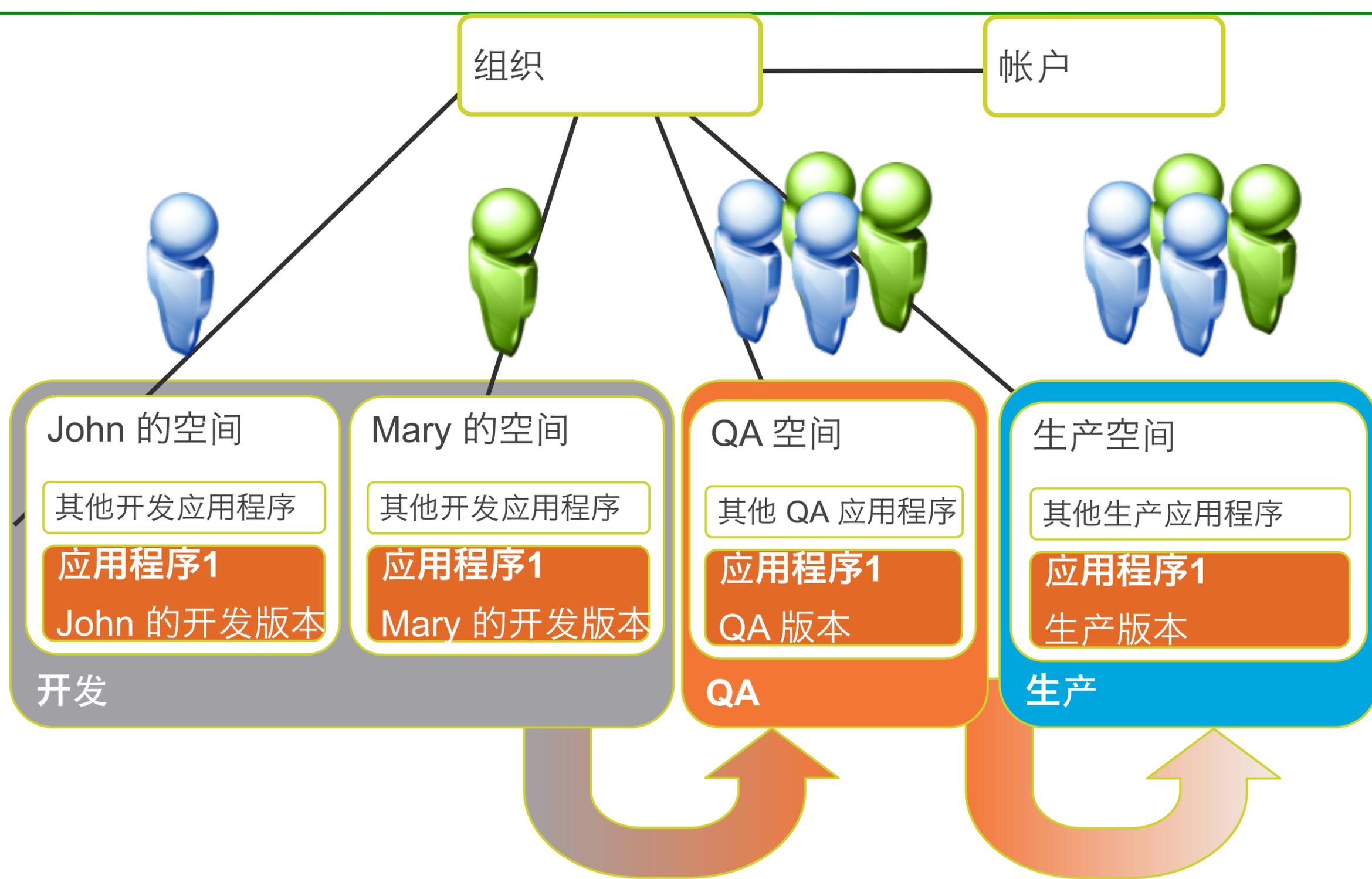
- 基于 Web 的门户
- 协作功能
- 控制成本

基于 Web 的门户

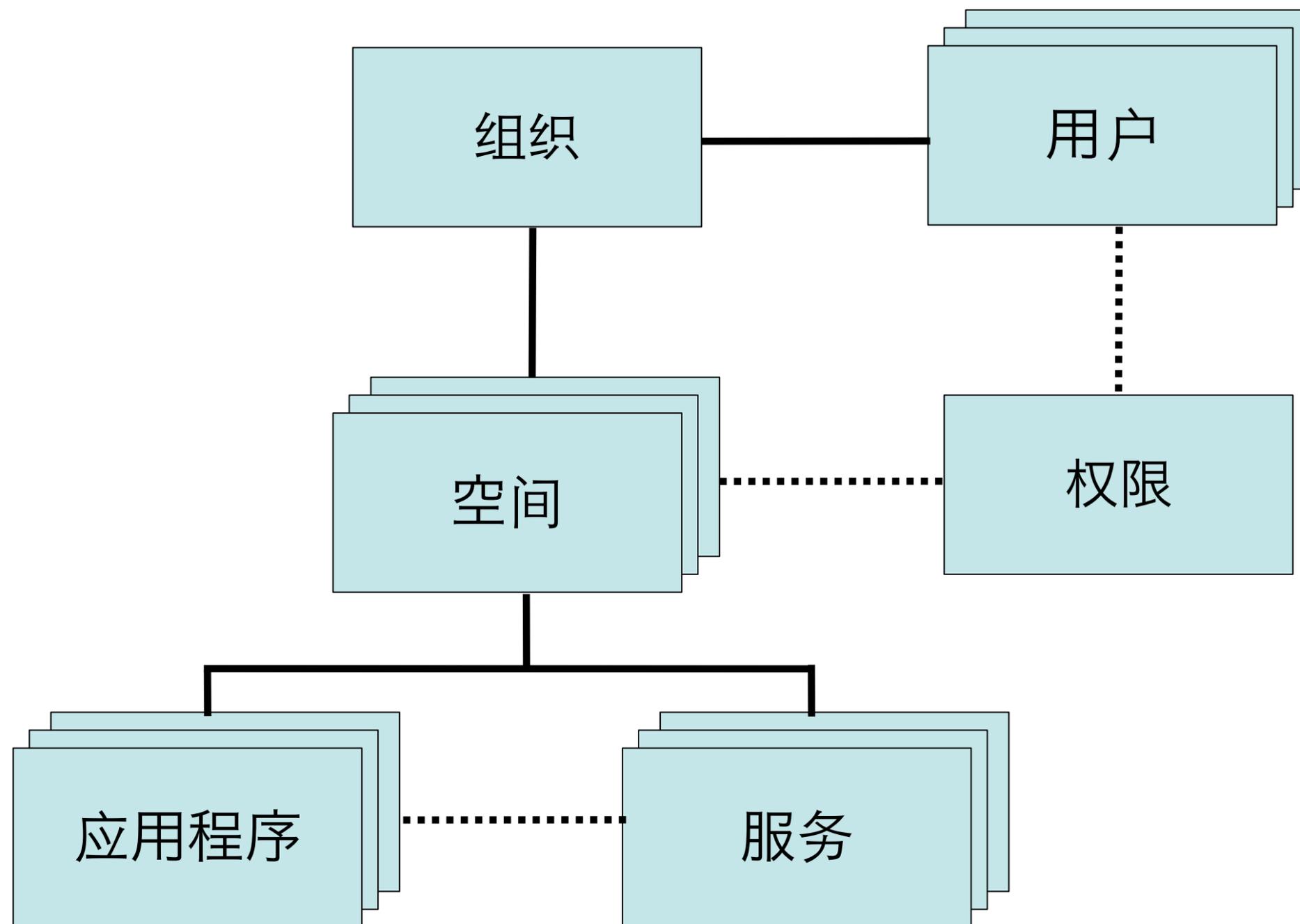
- 创建团队，分配角色
- 管理应用程序实例
- 配置服务增长的目录
- 审核事件和活动



操作协同



协作模型



vmc-ng : 组织和空间

```
# show my current context (cloud, org, and space)
$ vmc target
target: https://api.fakedomain.com
organization: pds
space: production

# switch to the staging space
$ vmc target -s staging
Switching to space staging... OK
target: https://api.fakedomain.com
organization: pds
space: staging

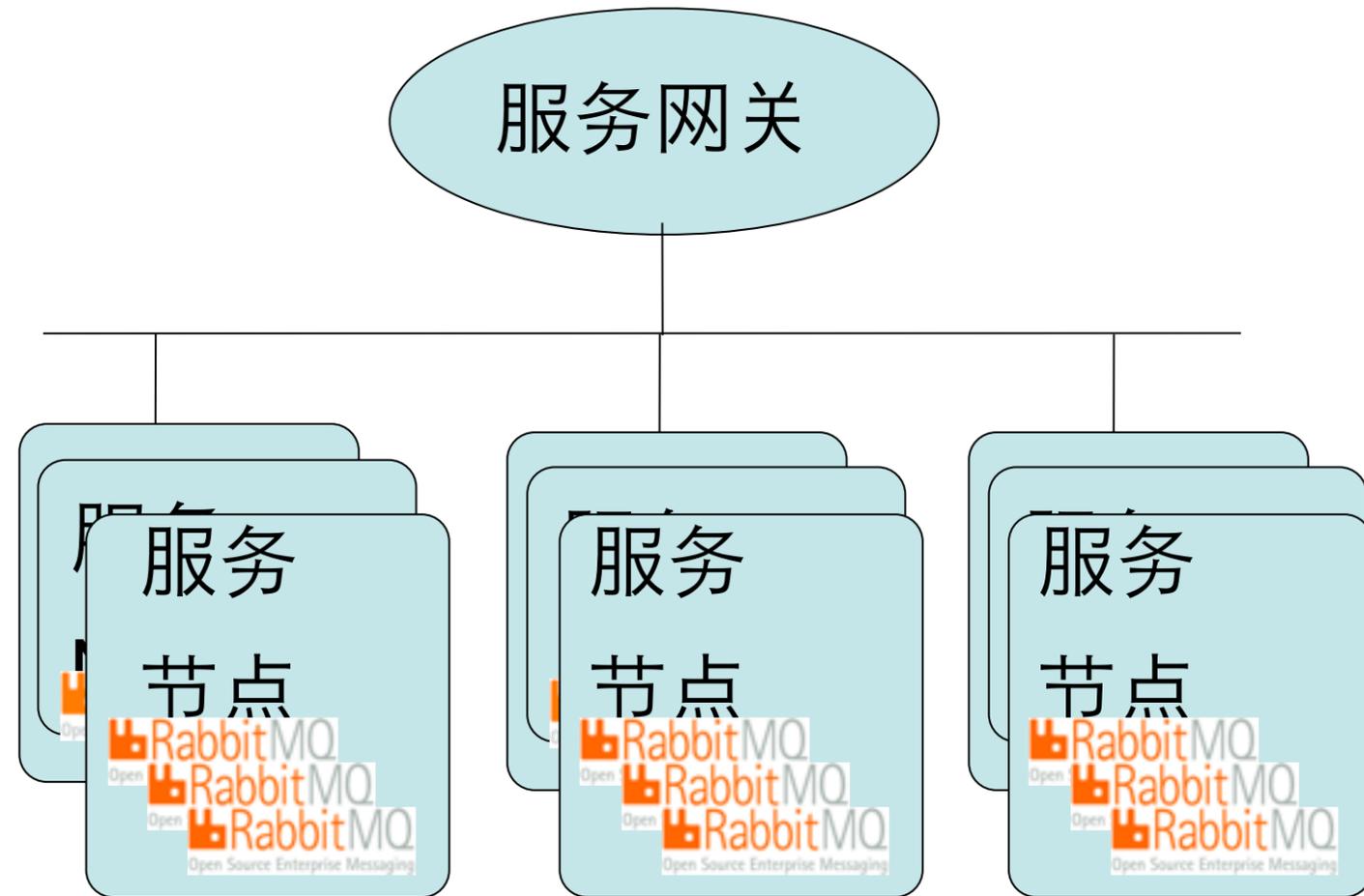
# switch to my other org
$ vmc target -o mhlsoft
Switching to organization mhlsoft... OK
Switching to space blaster... OK
target: https://api.fakedomain.com
organization: mhlsoft
space: blaster
```

成本管理

- 管理您的资金和资源使用
- 可配置的资金超支警报

计划

- `vmc create-service mysql --plan <plan>`
- 计划 = 服务参数、配置和选项的集合：
 - 大小
 - 最大并发连接数
 - 查询的超时时间
 - 查询的缓存大小
- 基于计划的服务网关路由配置请求



单租户实例

- 现在, cloudfoundry.com 服务节点是多租户的 (一个 Node 上托管多个实例)

- 大型/有安全需求的用户可能要求将他们的服务实例在其自己专用的 Node 上运行

- 计划参数：
 - capacity = 最大实例数

- 单租户实例：

capacity = 1



可按需增长的平台

- 开发内存和服务
 - 不昂贵
 - 低性能
- 生产内存和服务
 - 付费选项
 - 更好的性能

演示

- Play !
 - <https://github.com/playframework/Play20/tree/master/samples/java/zentasks>
- Spring Integration 工作者进程
 - <https://github.com/cloudfoundry-samples/twitter-rabbit-socks-sample>
- Unfiltered
 - <https://github.com/cloudfoundry-samples/bitshow>

问题？

Cloud Foundry 启动营

在 www.cloudfoundry.com 注册账号并成功上传应用程序,
即可于 12 月 8 日中午后凭账号 ID 和应用 URL 到签到处换取 Cloud Foundry 主题卫衣一件。



iPhone5 等你拿

第二天大会结束前，请不要提前离开，将填写完整的意见反馈表投到签到处抽奖箱内，即可参与“iPhone5”抽奖活动。



Birds of a Feather 专家面对面

所有讲师都会在课程结束后，到紫兰厅与来宾讨论课程上的问题

