

Pivotal®

Pivotal Web Service DevOps实践

吴疆

Pivotal中国研发中心资深产品经理

jwu@pivotal.io



Agenda

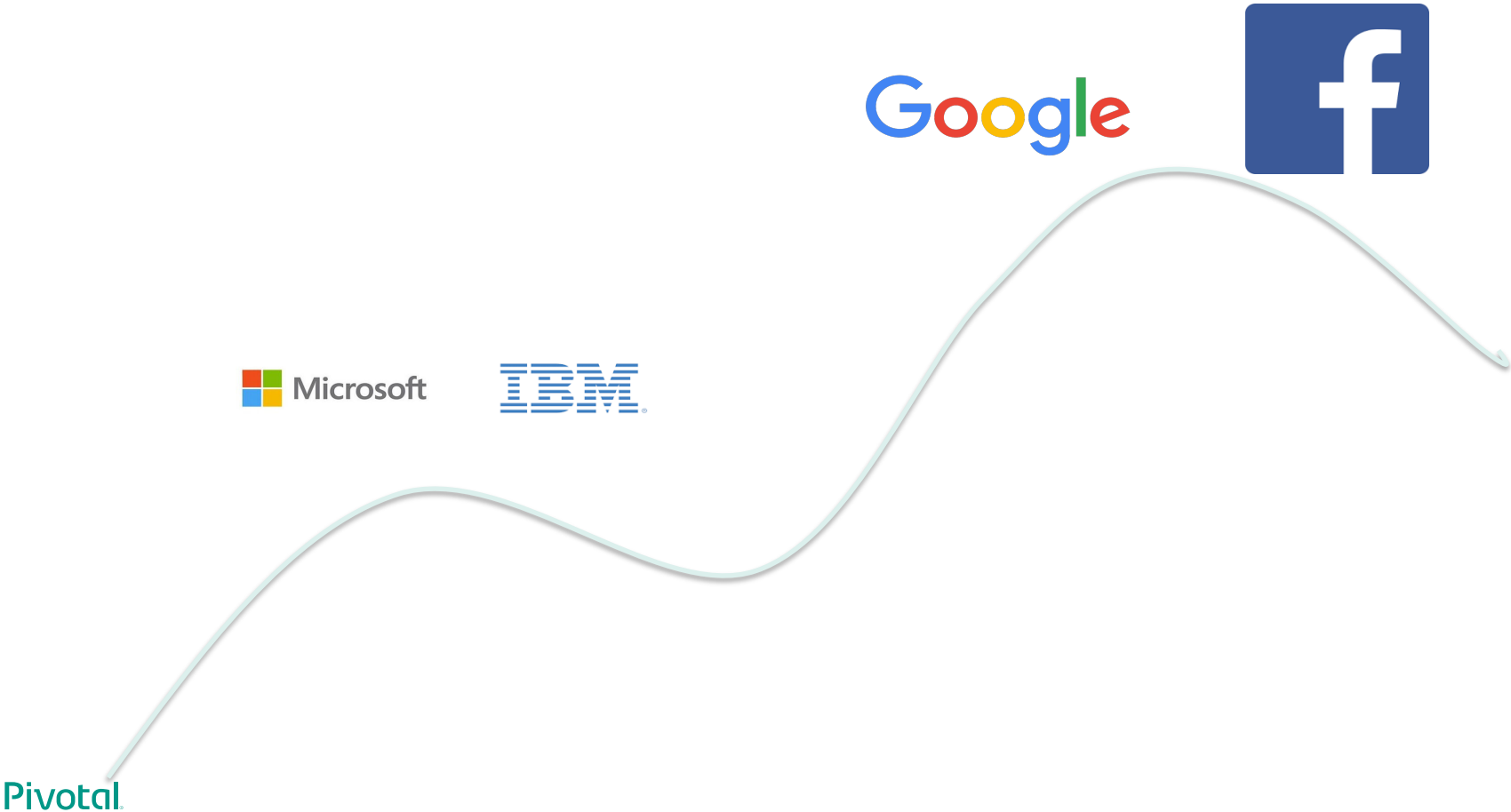
- 数字化战略和PaaS平台
- Cloud Foundry简介
- Pivotal Web Service (PWS) 简介
- Pivotal Web Service DevOps实践
- 小结
- Q+A



PIVOTAL Web Service DevOps实践

数字化战略和PaaS平台

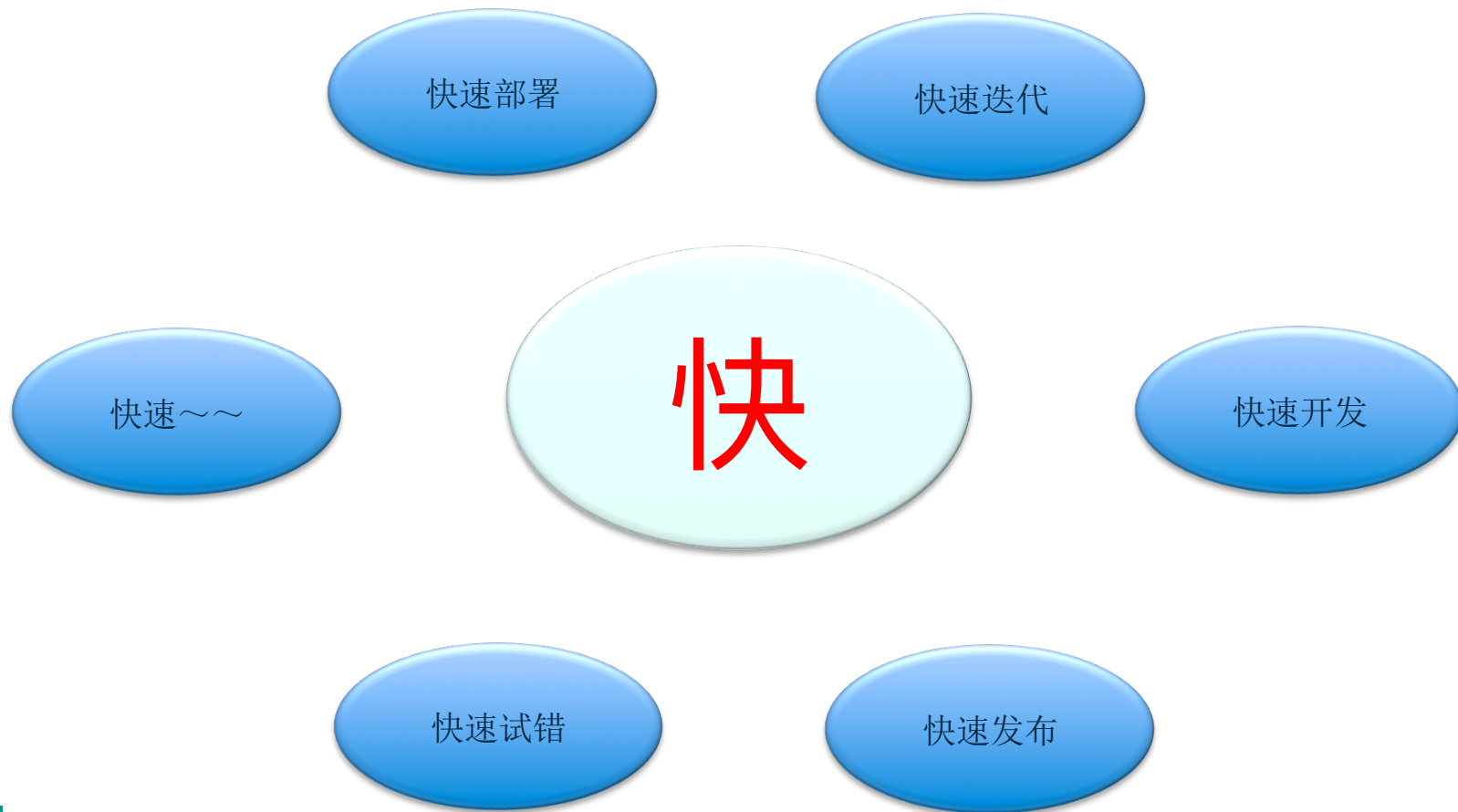
当今世界进入了互联网时代



PaaS和数字化战略



互联网应用的特点?



互联网应用的特点？



如何快速开发互联网应用?



IaaS云平台



我需要一台
服务器

服务器准备
好了



太好了，我可以开始了：

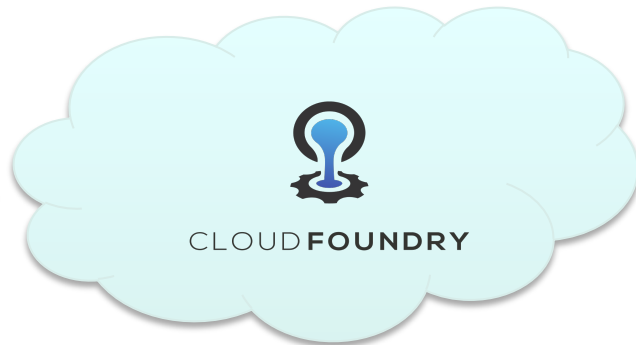
1. 安装数据库
2. 编译打包
3. 安装依赖的库
4. 部署

PaaS云平台



这是我的代码，
请帮我运行

好了

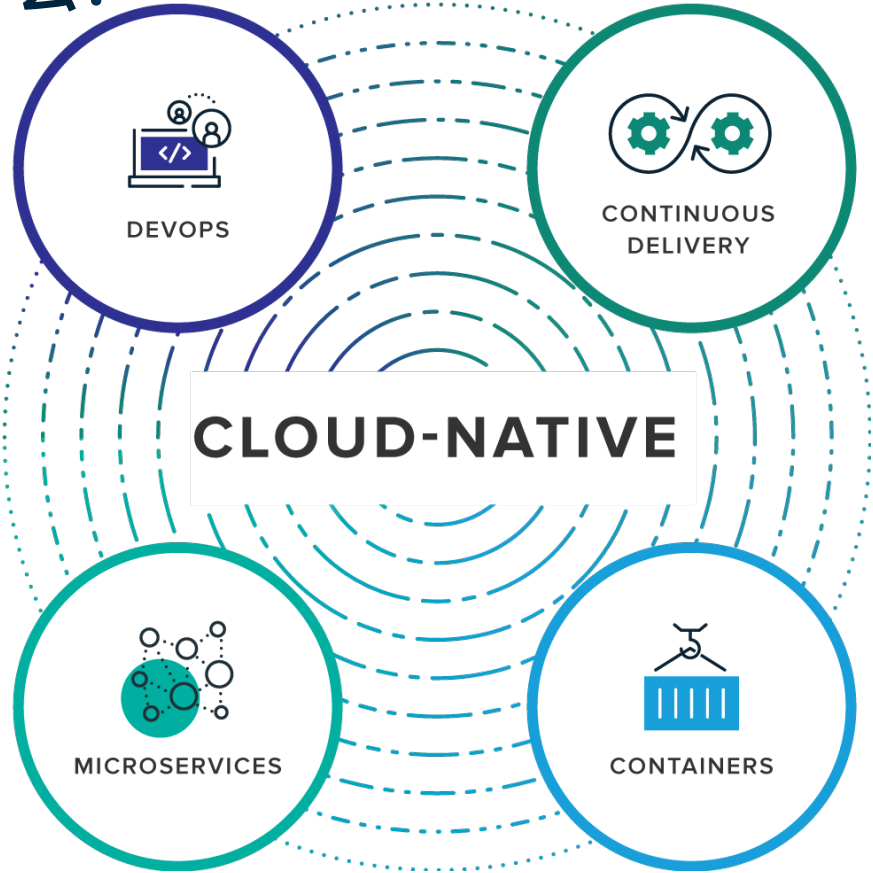


Platform as a Service平台是什么？

Platform as a Service（简称PaaS）是一种为云原生应用提供运行时环境的平台。



云原生应用是什么？



PIVOTAL Web Service DevOps实践

Cloud Foundry简介

Cloud Foundry是什么？

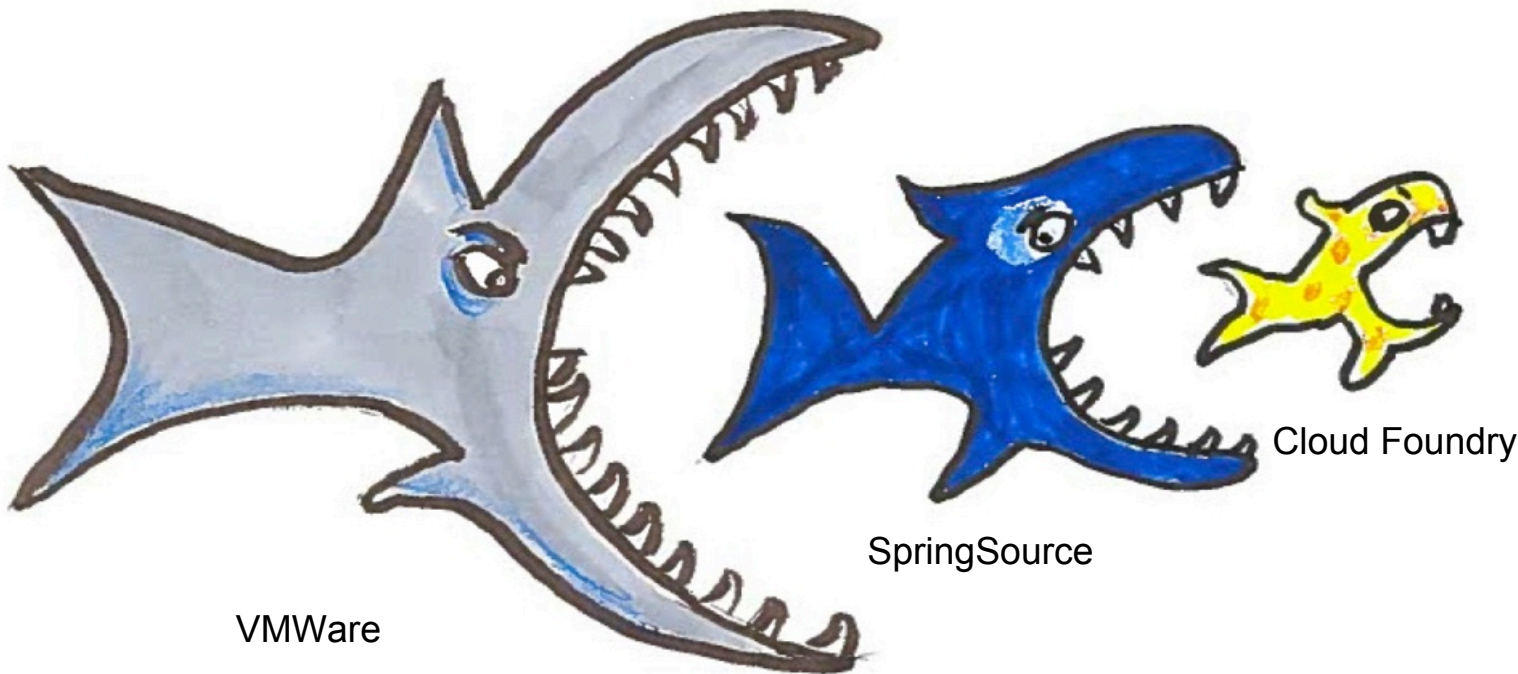
Cloud Foundry是一个开源的Platform as a Service (PaaS) 平台。

- 开源：<http://github.com/cloudfoundry>
- 社区驱动：Cloud Foundry基金会
- 成熟、Production-Ready
 - Pivotal CF
 - IBM Cloud Foundry
 - Huawei FusionStage
 - SAP Cloud Platform
 - etc



CLOUDFOUNDRY

Cloud Foundry的历史



Cloud Foundry的设计目标

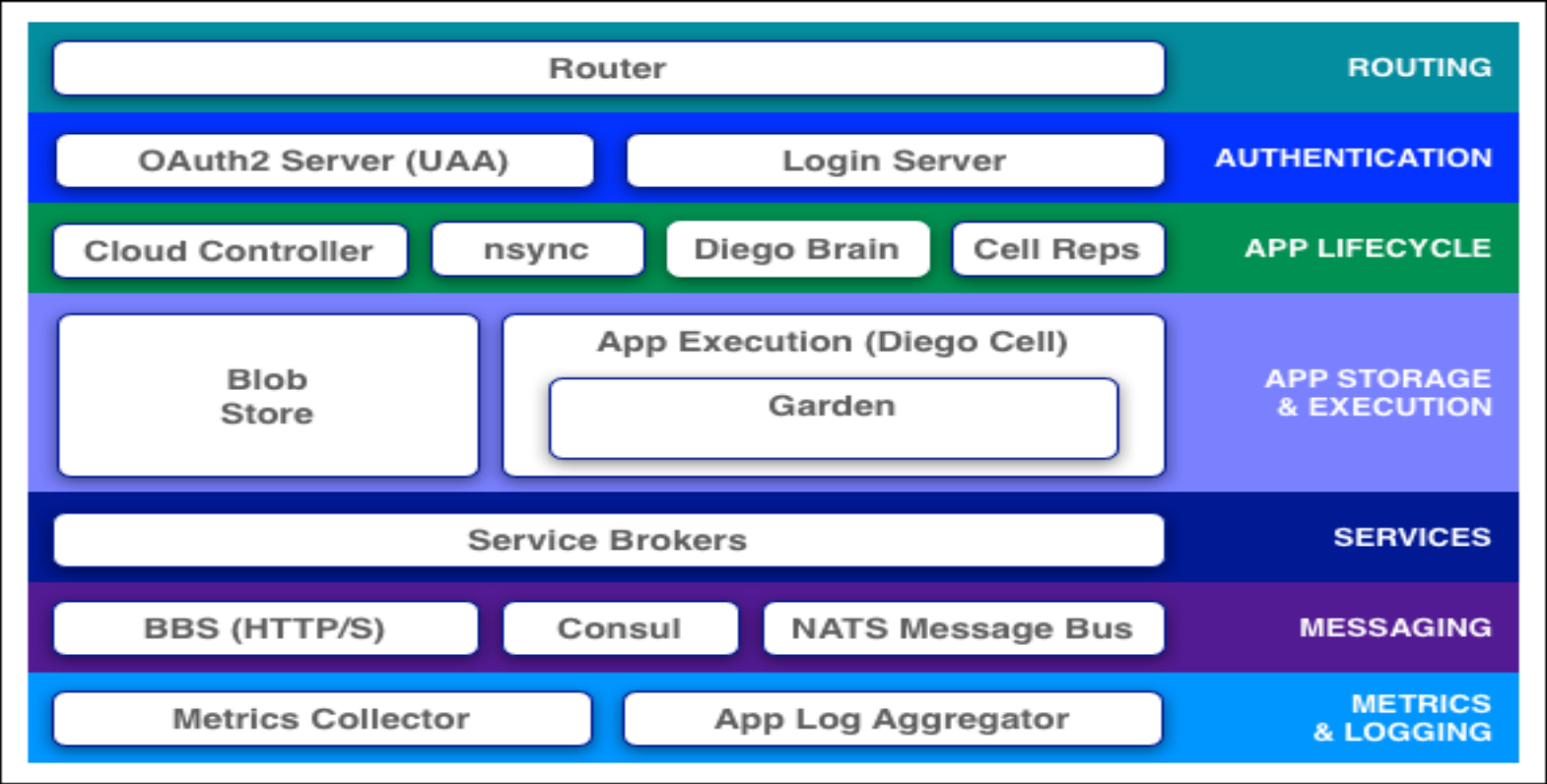
Cloud Foundry是一个为云原生应用设计的云原生平台：

- 云原生
- 易于扩展
- 快速发布 快速升级
- Multi-Cloud
 - vSphere/OpenStack
 - AWS/Azure/Google Cloud



CLOUD FOUNDRY

Cloud Foundry架构图



PIVOTAL Web Service DevOps实践

Pivotal Web Service (PWS) 简介

Pivotal Web Service (PWS)是什么？

Pivotal Web Service（简称PWS）是一个在线的Cloud Foundry实例。

- 部署在AWS
- 由Pivotal运行维护
- 提供共有云服务
- 网址：<https://run.pivotal.io>



Pivotal Web Services

Pivotal Web Service (PWS)的设计目标

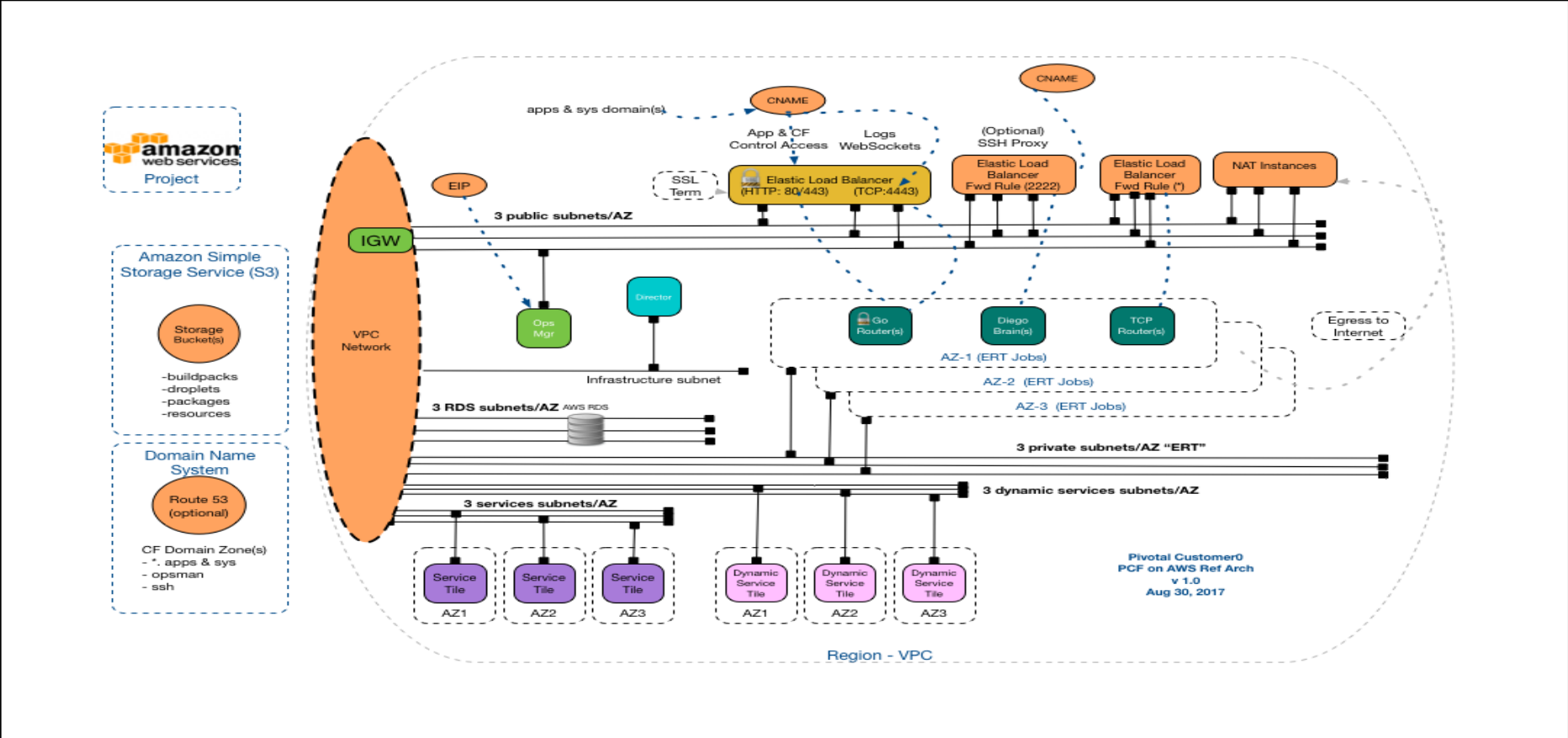
Pivotal Web Service (简称PWS) 是一个在线的 Cloud Foundry实例.

- 可扩展
- 多租户
- 高可用



Pivotal Web Services

Pivotal Web Service架构图



Pivotal Web Service (PWS)的规模

Pivotal Web Service (简称PWS) 是目前规模最大的 Cloud Foundry实例.

- 部署在Amazon AWS的3个可用区
- 400+虚拟机
- 18000+ 应用实例
- 20,000,000+ request/天



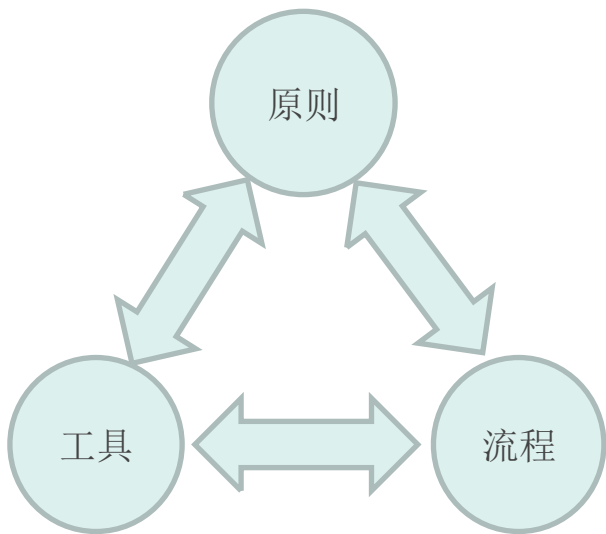
Pivotal Web Services

PIVOTAL Web Service DevOps实践

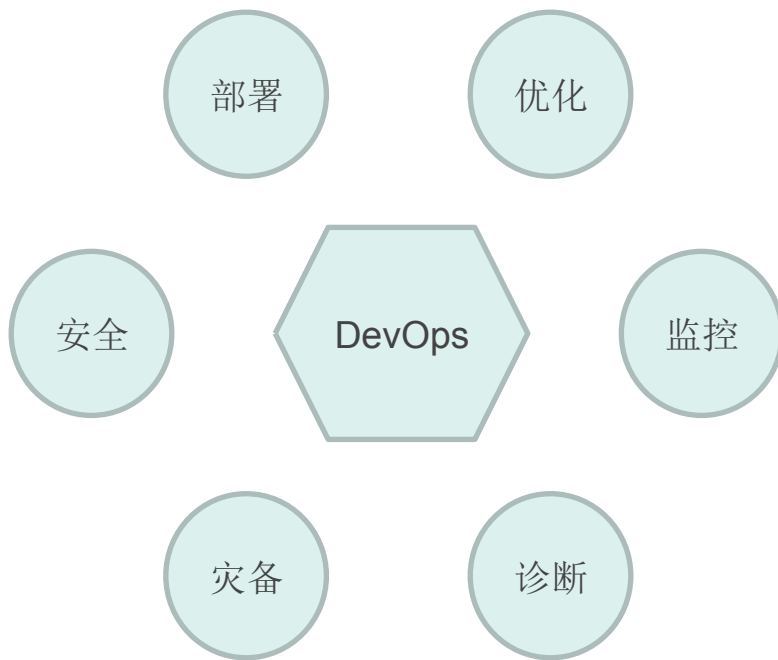
PWS DevOps实践

PWS DevOps实践

PWS DevOps实践框架



PWS DevOps内容



PWS DevOps的原则

原则是指导行动的“最高准则”或者“标准”.

- 系统化，流程化，框架化
- 消除单点故障 (single point of failure)
- 消除手动事务，自动化所有操作
- 监控跟踪所有模块
- 不断进化



Pivotal Web Services

PWS DevOps工具

制作和使用工具，是人类区别于其他动物的基本特征



PWS DevOps的工具原则

所有的操作都要工具化，使用工具达到自动化

- 尽可能的使用已有成熟工具
- 没有成熟工具的情况下要敢于开发新工具
- 注重工具之间的集成
- 运维系统的产品化



Pivotal Web Services

PWS DevOps的流程

流程可以使成功变得可重复，是一个组织成熟的重要标志

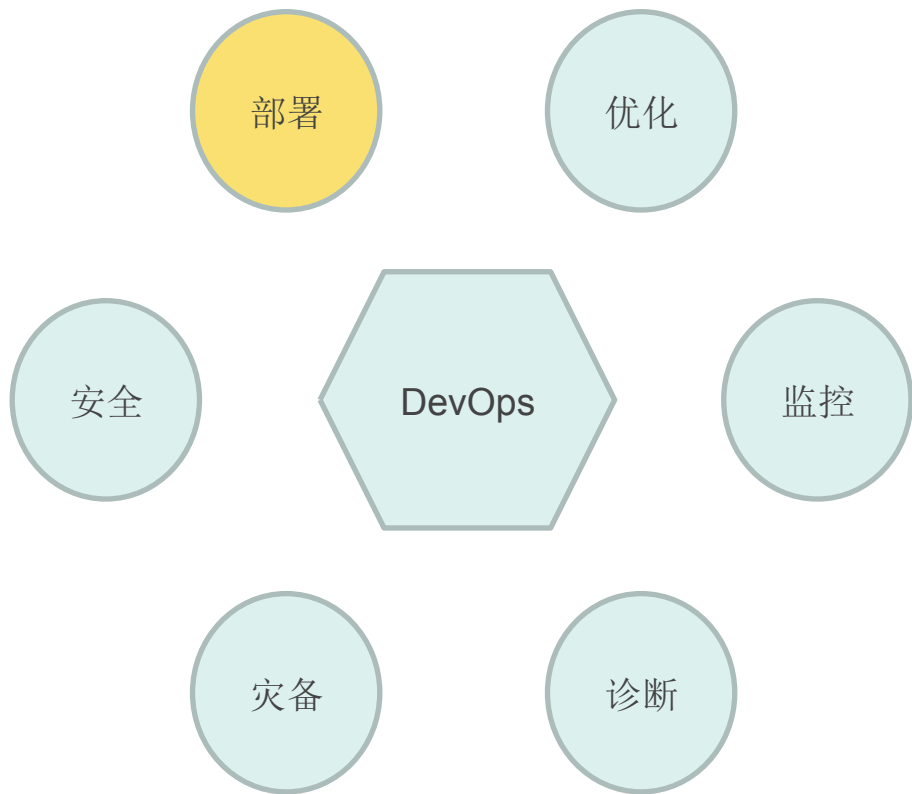
- 敏捷
- 高效
- 进化



Pivotal Web Services

PIVOTAL Web Service DevOps实践

PWS DevOps最佳实践



PWS的部署

- PWS的升级
- PWS的CVE补丁
- PWS升级的回滚



Pivotal Web Services

PWS的部署

原则

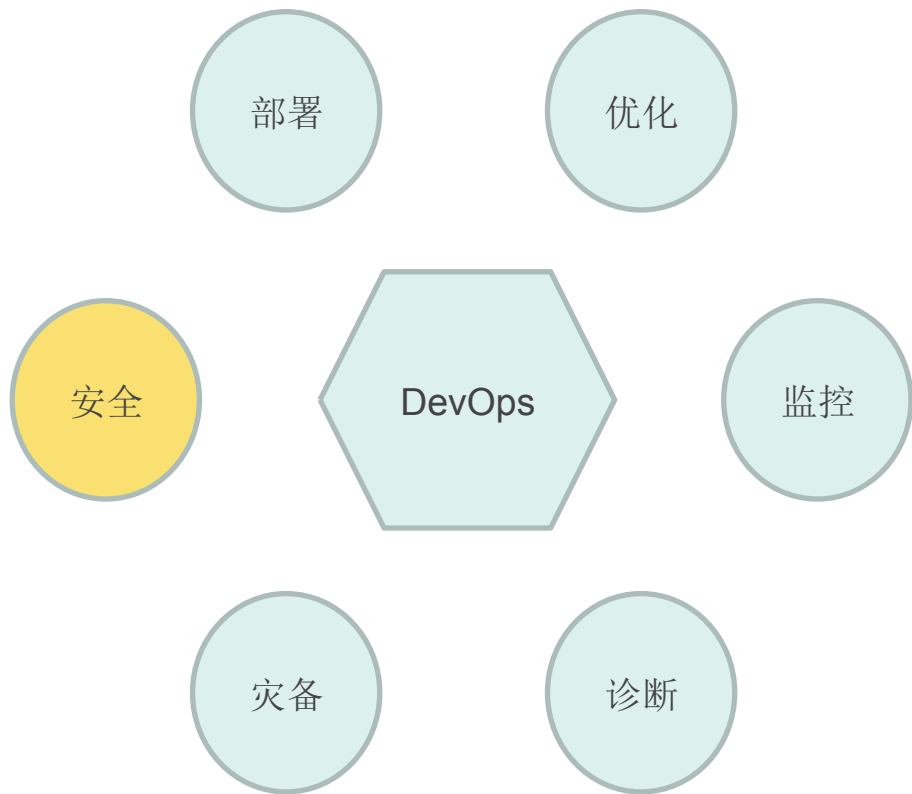
- 任何部署都预先被验证
- 任何部署都必须是可回滚的
- 部署必须是自动化的
- 部署不能影响系统的可用性
- 灰度部署

工具

- Concourse
- BOSH
- Github
- Smoke Test

流程

- 部署前验证
- 部署中的监控
- 部署后的验证
- 部署的回滚
- 降低风险



PWS的安全最佳实践

- 3R安全实践：Rotate, Repave, Repair
 - Rotate the credentials
 - Repave the servers and applications
 - Repair the vulnerable OS and applications
- PWS的各模块之间的通信加密
- PWS的credential的集中管理



Pivotal Web Services

PWS的安全

原则

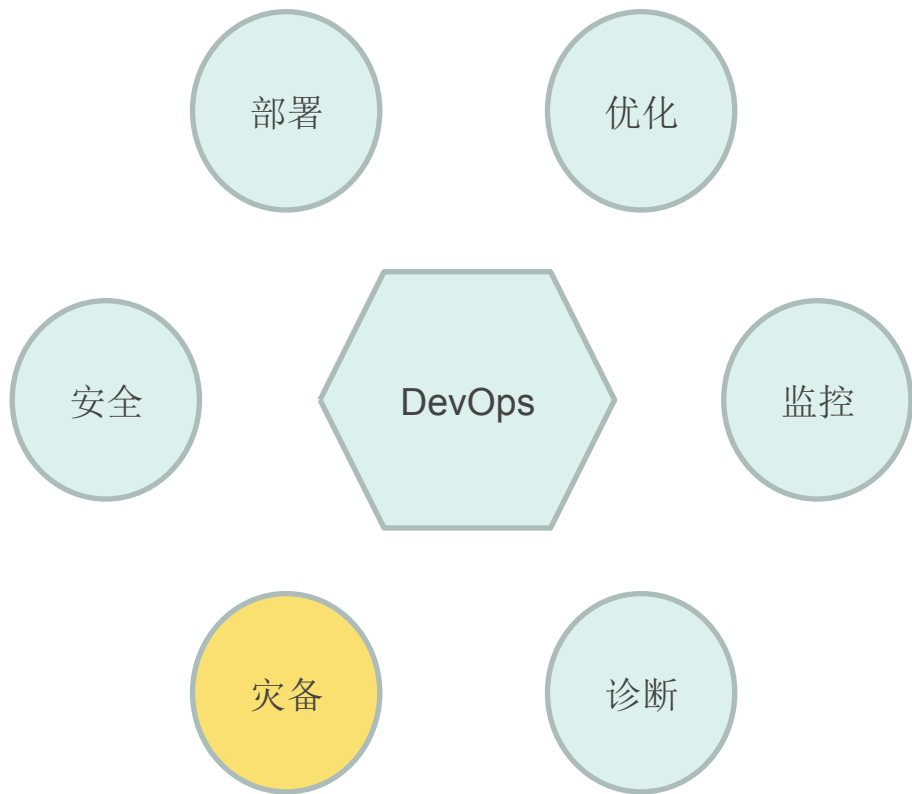
- 安全性是区分产品和玩具的一个重要标准
- 任何外部通讯必须是加密的
- 任何内部通讯也必须是加密的

工具

- Concourse
- BOSH
- CredHub

流程

- Rotate流程
- Repave流程
- CVE的流程



PWS的灾备

- PWS的高可用设计
- PWS的数据库的高可用和灾备
- PWS的App的灾备



Pivotal Web Services

PWS的灾备

原则

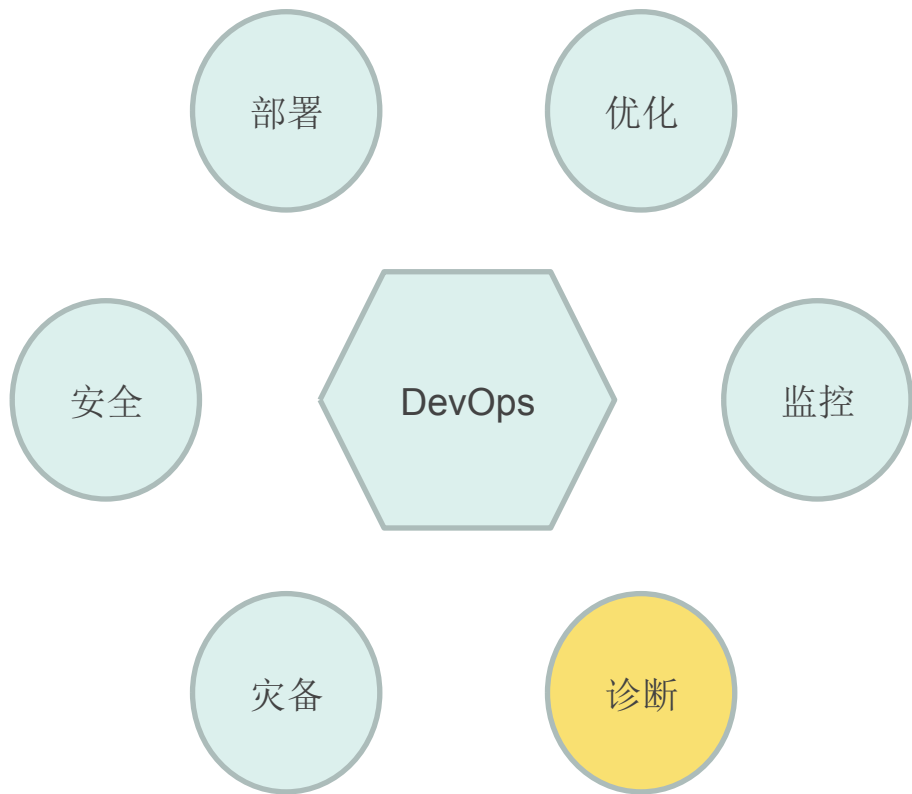
- 无单点故障
- 任何数据都要备份
- AWS的一个可用区的故障不影响系统的可用性

工具

- Concourse
- BOSH
- S3

流程

- 灾备计划



PWS的故障诊断

- PWS的故障定位
- PWS的日志收集
- PWS的日志集中管理



Pivotal Web Services

PWS的故障诊断

原则

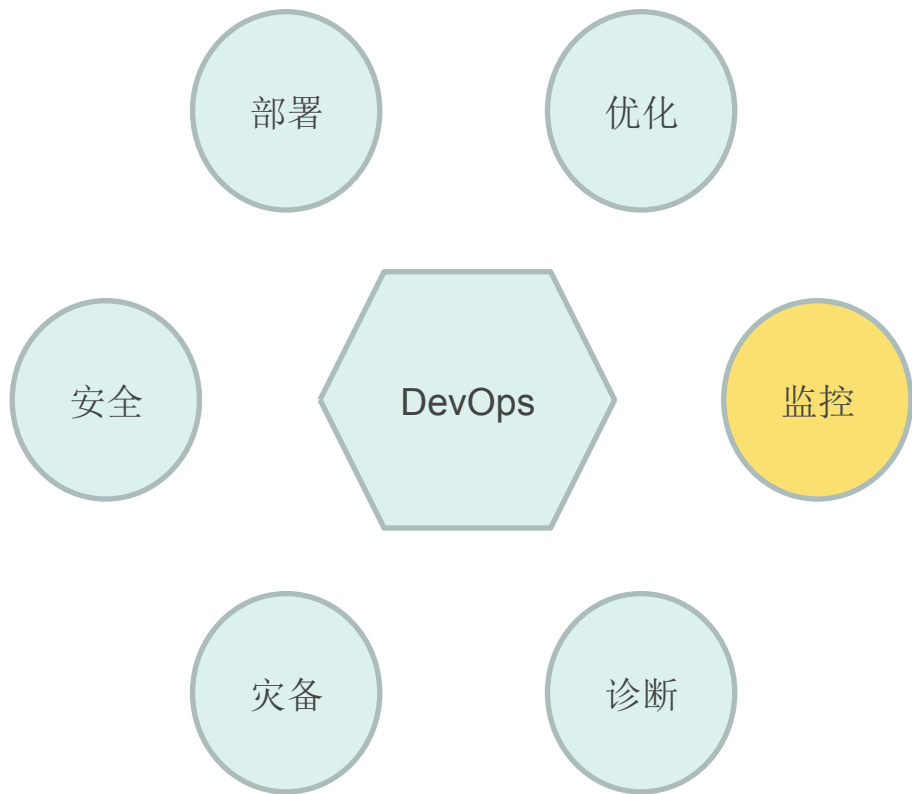
- 任何日志都要收集
- 日志要长期保存
- 任何故障只能发生一次

工具

- Concourse
- BOSH
- ELK: elasticsearch/logstash/kibana
- PCF metrics

流程

- 日志收集流程
- 日志管理流程
- 故障定位流程
- 故障反思流程



PWS的监控和报警

- 监控的层次
 - 系统级：cpu/memory/disk/network
 - 平台级：数据库/缓存/队列长度
 - 应用计：访问次数/在线用户数
- PWS的故障报警
 - 警报的级别
 - 复杂事件处理



Pivotal Web Services

PWS的故障诊断

原则

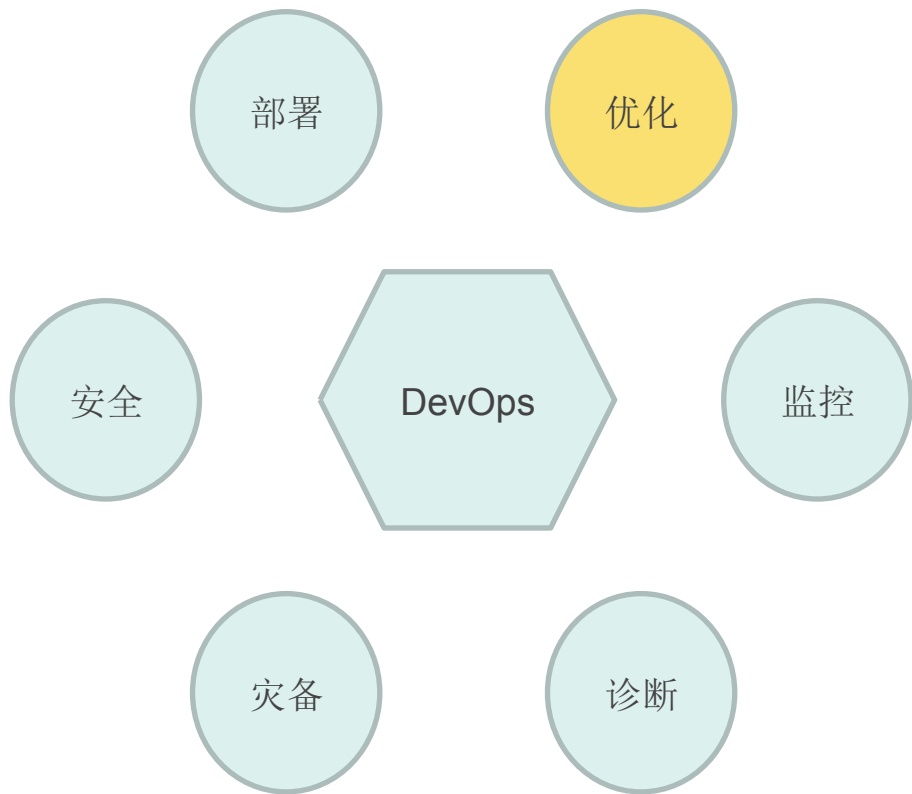
- 所有的功能模块都要监控
- 警报要准确
- 故障要能自动恢复

工具

- Concourse
- BOSH
- Datadog
- Pagerduty

流程

- 监控流程
- 故障处理流程
- Oncall管理流程
- 故障反思流程



PWS的优化

- 优化的角度
 - 性能优化
 - 容量优化
 - 成本优化



Pivotal Web Services

PWS的优化

原则

- 更好的性能
- 合适的容量
- 最低的成本

工具

- Concourse
- BOSH
- Excel

流程

- 运维数据分析流程
- 成本分析流程
- 容量规划流程

PIVOTAL Web Service DevOps实践

总结

困境和矛盾

历史趋势不可阻挡

- 系统变得越来越大
- 系统变得越来越复杂
- 系统越来越分布式化
- 系统的可用性要求越来越高
- ...

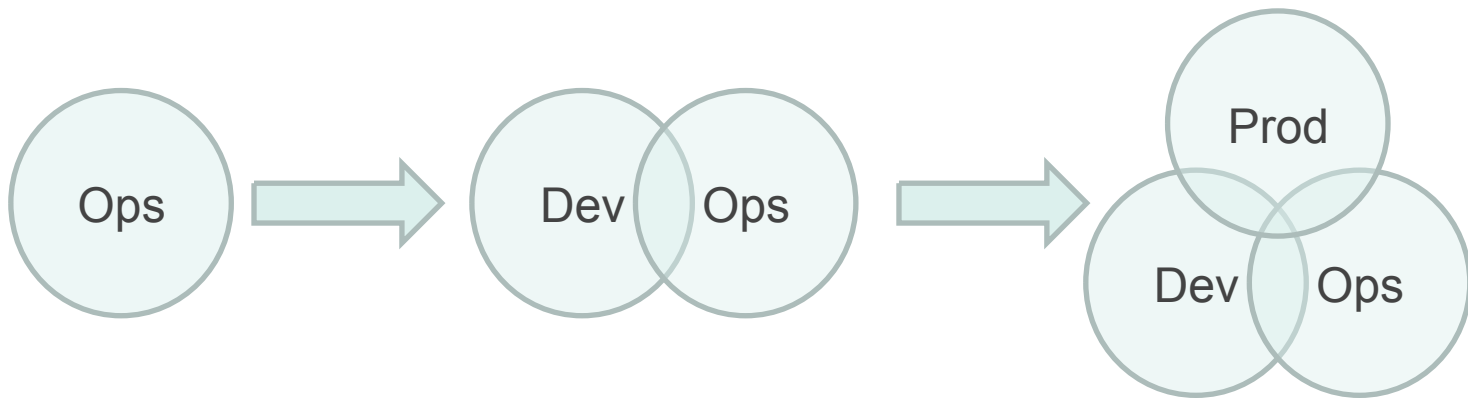
我们的困境

- 系统运维很难
- 分布式系统的运维更难
- 各种指标要求越来越高
- 成熟的工具理念很老旧
- 理念很先进的工具很不成熟
- ...

世界变了 我们也要改变



世界变了 我们也要改变



Pivotal®

Transforming How The World Builds Software