

# 云计算开发运维DevOps实践指南

--亚马逊开发运维的最佳实践和工具

张侠博士

AWS首席云计算企业顾问

2015年7月17日

深圳

# 议题

- 开发运维DevOps介绍
  - 基础架构即代码 Infrastructure as Code
  - 持续集成持续交付，IT自动化
  - 微服务 Microservice
- 用AWS实现DevOps的框架和工具
  - CodeCommit
  - CodePipeline
  - CodeDeploy
  - API Gateway
  - CloudFormation
  - Beanstalk
  - OpsWorks
  - Cloudwatch/CloudTrail

# 什么是DevOps ?

# DevOps的理念

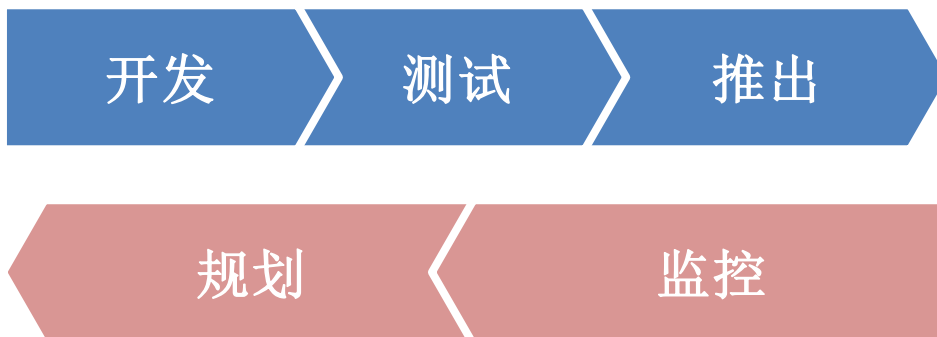


## 软件开发周期

交付管道



开发者



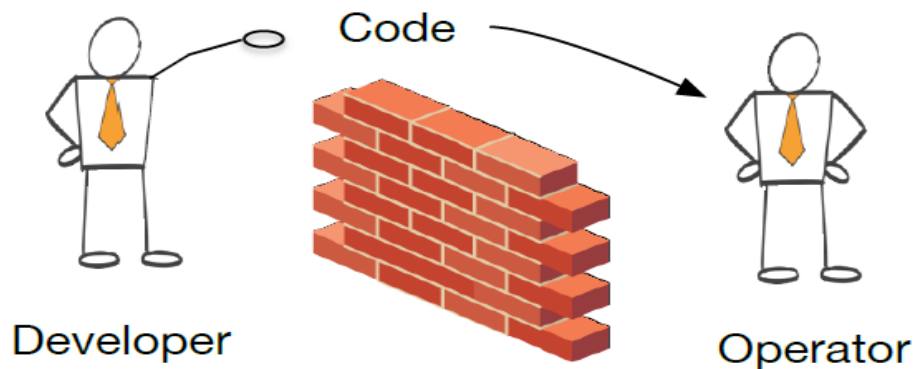
使用者

反馈

**DevOps** = 高效的加速开发周期

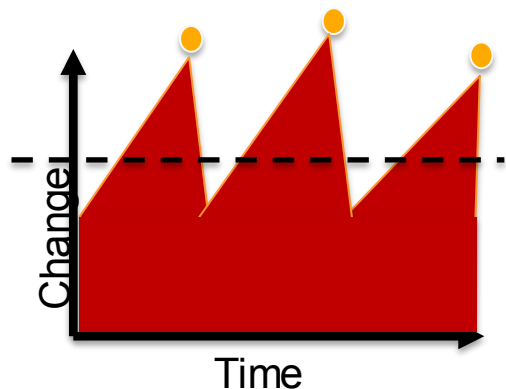
# 为什么需要DevOps?

因为我们不希望事情是这样的...

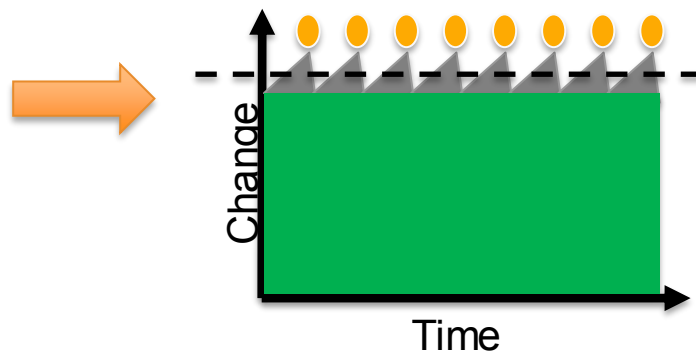


我们希望...

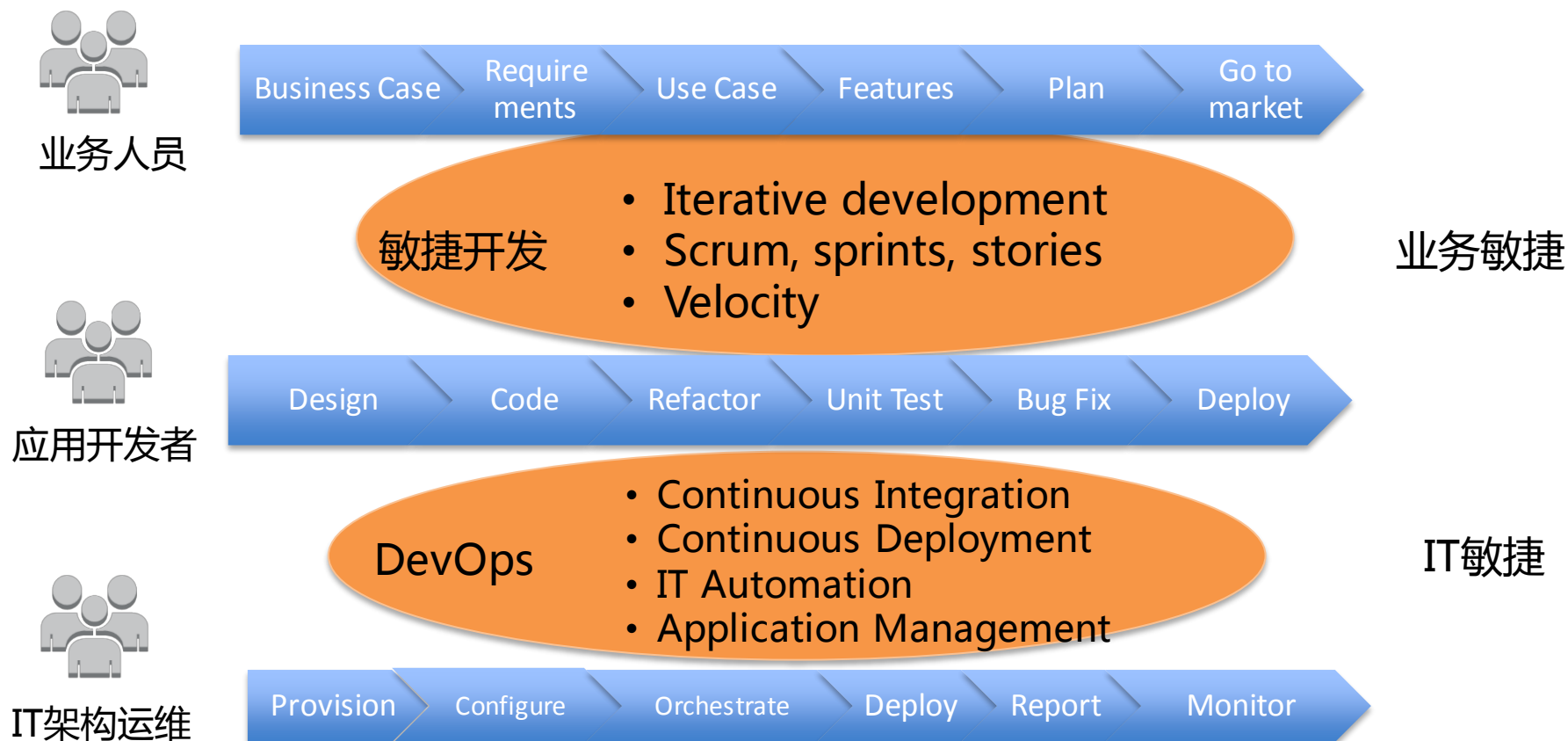
瀑布式开发，版本发布少



快速迭代, 敏捷开发



# DevOps来自于敏捷开发



# DevOps 主要原理

- 协同合作
- 打通壁垒
- 端到端的团队合作
- 基础设施即代码 （ Infrastructure as a Code ）
- 支持业务和技术灵活性
- 自动化一切
- 测试一切
- 测试和监控一切

# DevOps 实践

- 基础设施即代码 Infrastructure as a Code
- IT自动化
- 持续集成
- 持续部署
- 版本控制集成
- 应用和架构的版本管理
- 监控和日志



# 基础设施即代码 Infrastructure as code

- 可扩展性 （任何手动的都不具备好的扩展型）
- 可靠性
- 可重复、可复制
- 环境的一致性
- 可审计、记录保存
- 安全性
- 管理治理

# Infrastructure as Code 基础设施即代码示例

```
"WebServer": {
  "Type": "AWS::EC2::Instance",
  "Metadata" : {
    "AWS::CloudFormation::Init" : {
      "config" : {
        "packages" : {
          "yum" : {
            "httpd"      : [],
            "php"        : [],
            "php-mysql"  : [],
            "php-gd"     : [],
            "php-xml"    : [],
            "php-mbstring" : [],
            "mysql"      : []
          }
        }
      },
      "sources" : {
        "/var/www/html" : "http://ftp.drupal.org/files/projects/drupal-7.8.tar.gz",
        "/home/ec2-user" : "http://ftp.drupal.org/files/projects/drush-7.x-4.5.tar.gz"
      }
    }
  }
}
```



AWS  
CloudFormation



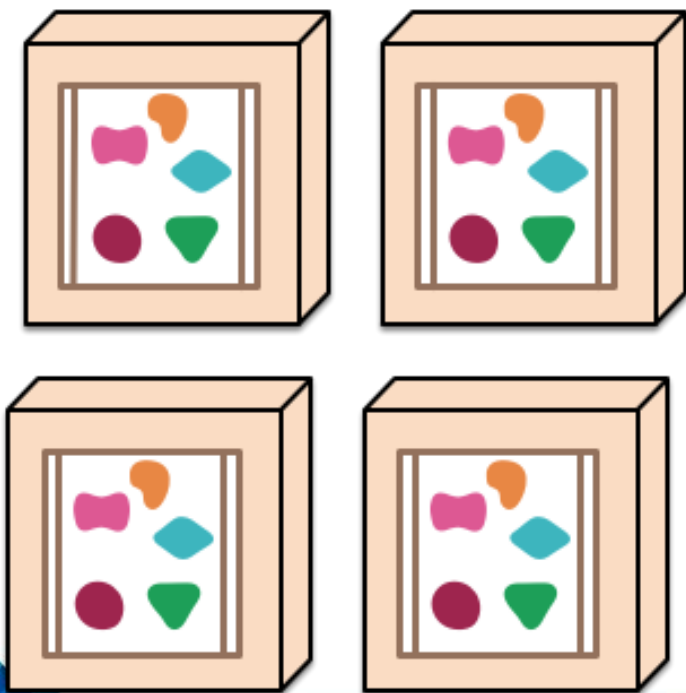
template

# 单体Monolithic vs 微服务架构

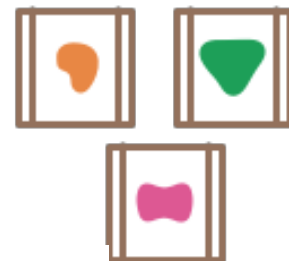
单体应用将所有功能  
集于一体一个线程...



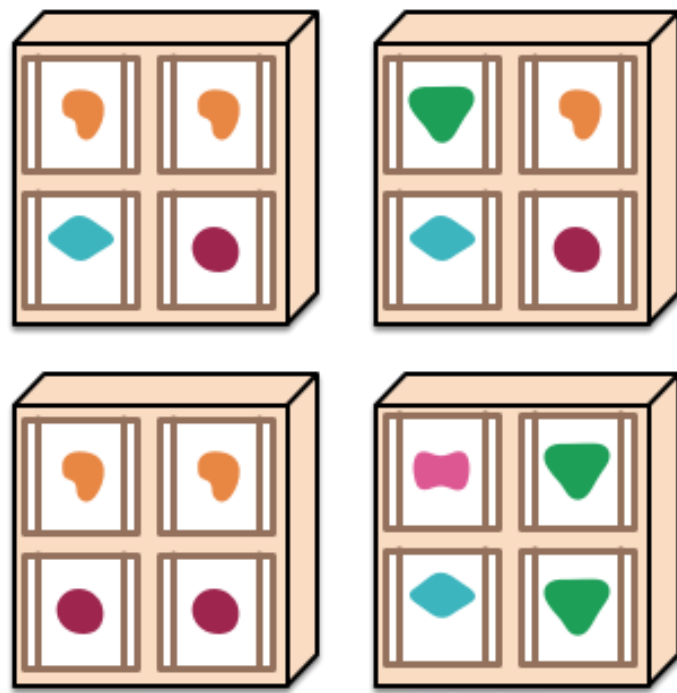
...通过复制整体应用在  
多个服务器上来扩展



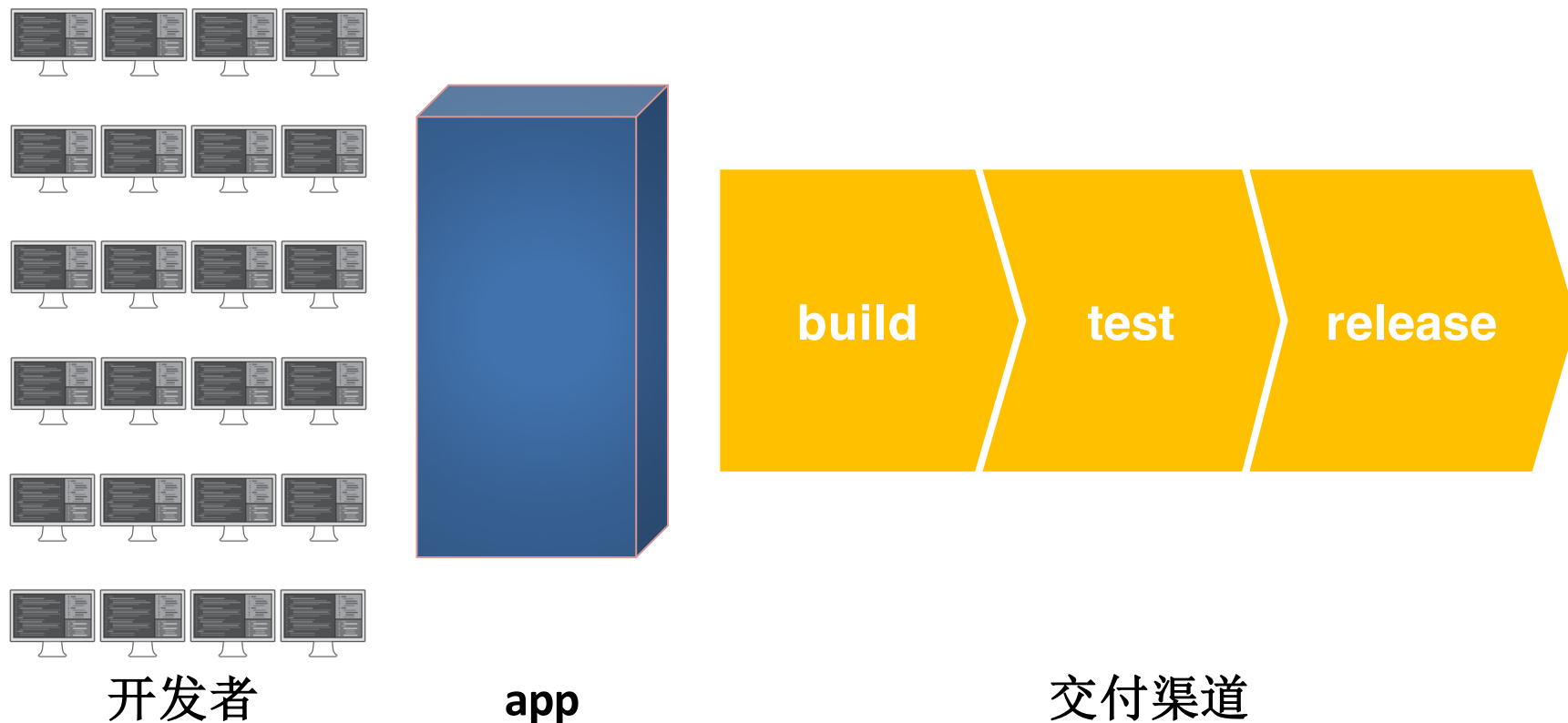
微服务将每个功能集做  
成一个独立的服务...



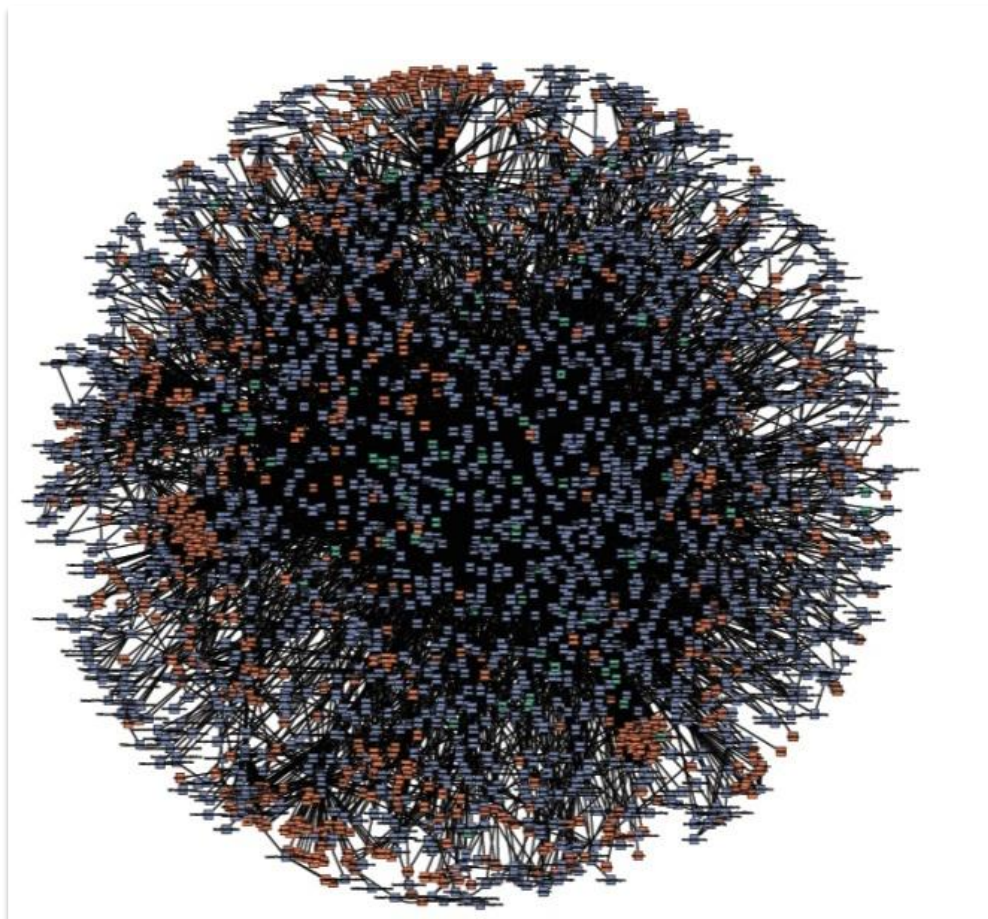
...通过将为服务按需复制分布  
在多个服务器上来扩展



# 单体Monolithic开发生命周期



# 微服务 Microservices



- 面向服务的架构
- 目的单一原生小组 (Primitives)
- 仅通过APIs 连接
- 微服务  
“Microservices”

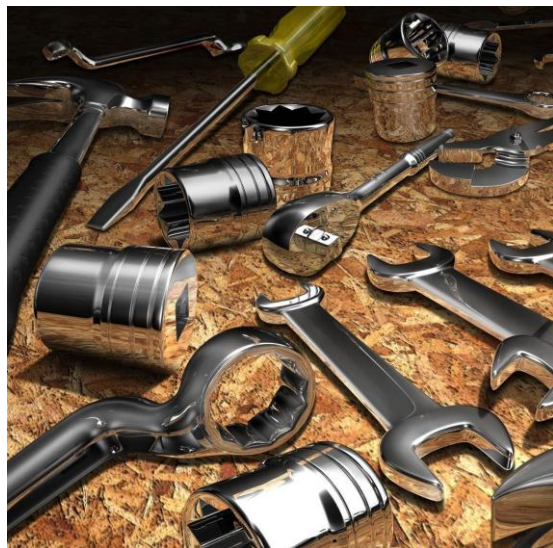
# Two-Pizza Team 小团队



- 两个比萨饼团队
- 自主负责
- 承担责任
- 定向激励
- “DevOps”



# 特别任务，特别团队



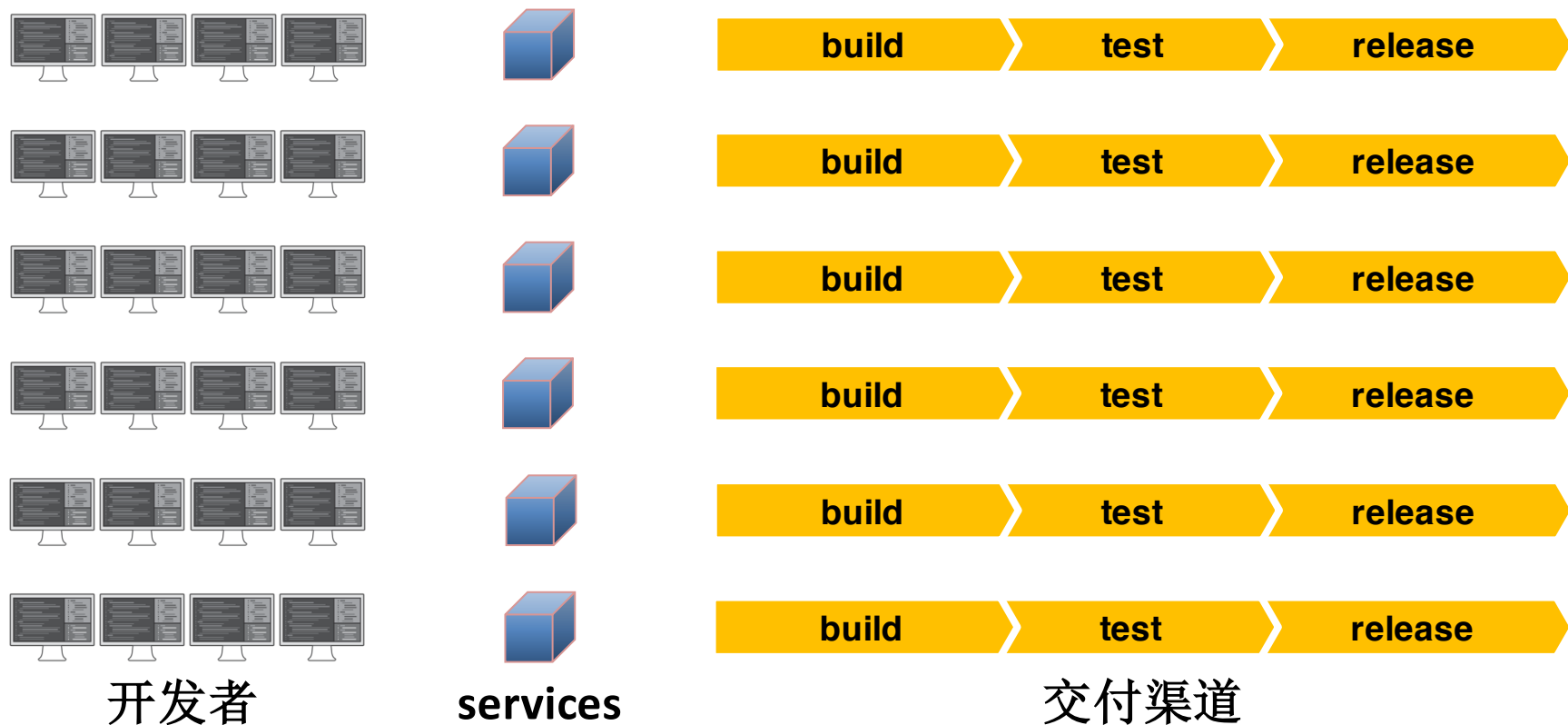
Services, 各种服务

Apollo, 部署引擎



Pipelines, 持续交付

# 微服务Microservice开发生命周期





# 亚马逊开发团队最佳实践

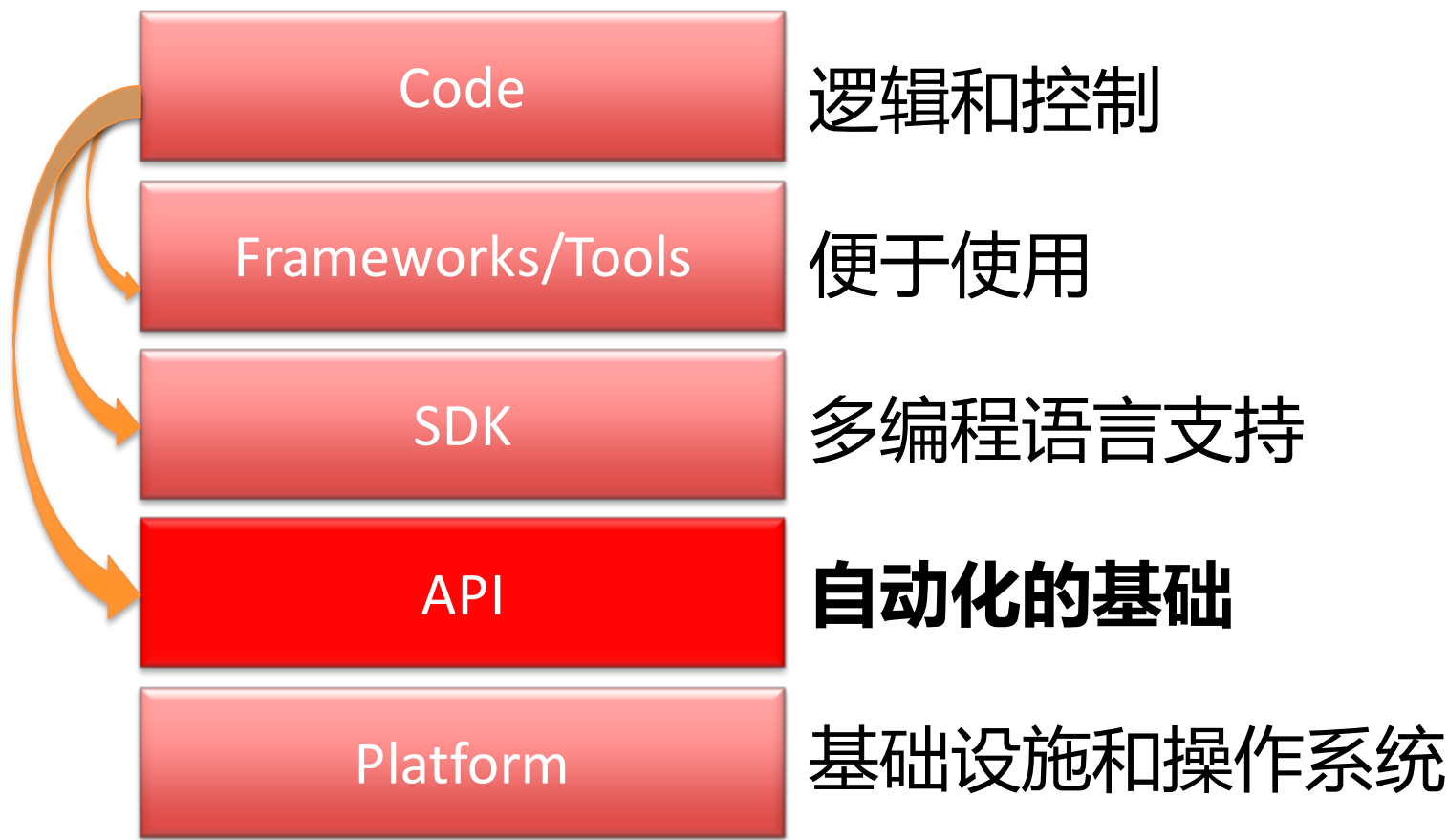
数千个团队  
× 微服务架构  
× 持续交付  
× 多环境部署

---

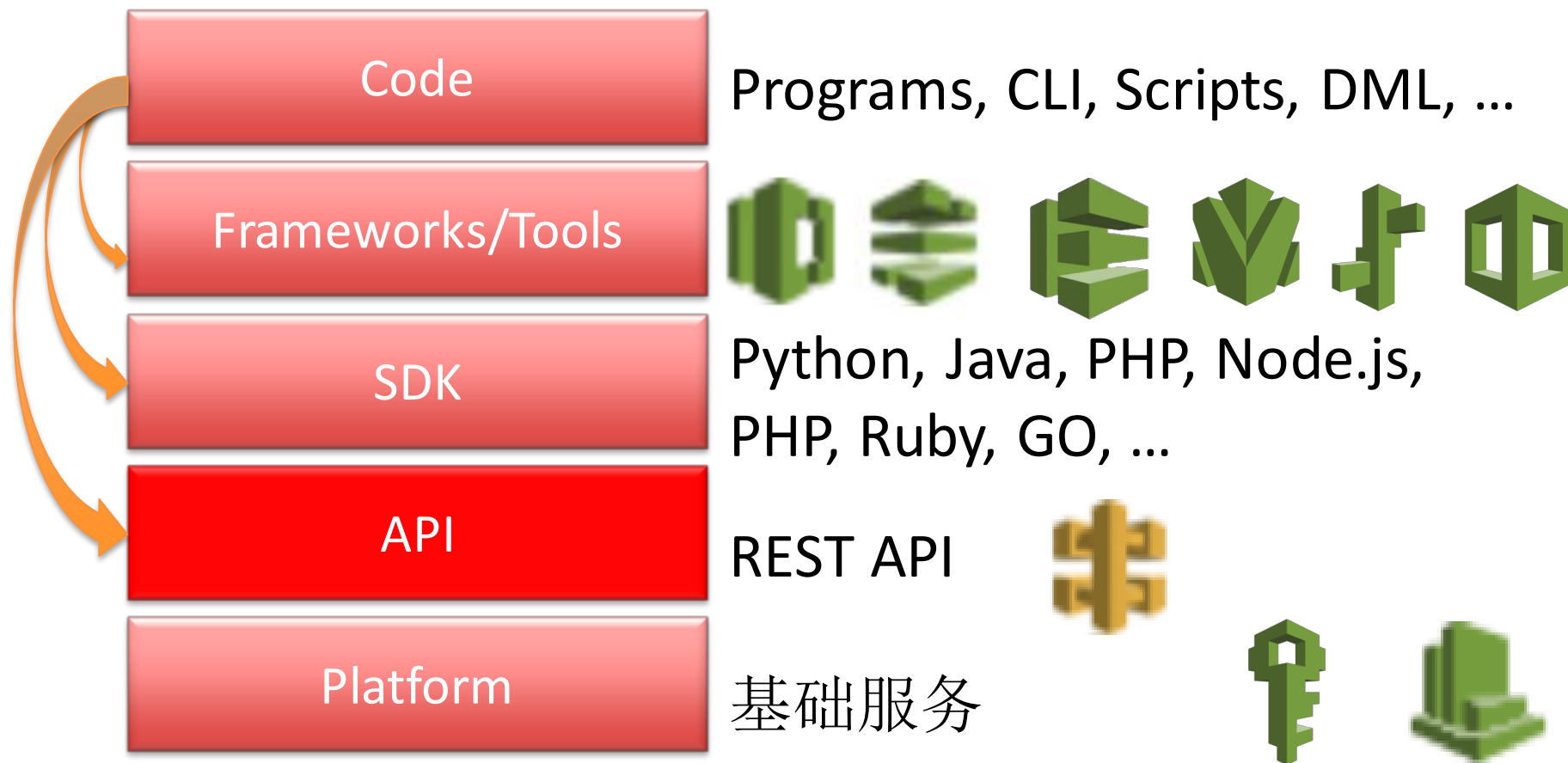
= 每年5千万次交付

# 怎么实现DevOps ?

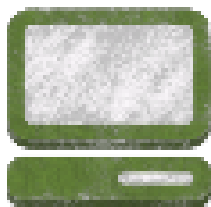
# 从软件开发的角度看DevOps



# AWS对DevOps的全面支持



# 操作AWS服务的三种方式



Management Console



APIs and SDKs



Command Line Interface

DevOps基础

AWS Tools (SDK, CLI, IDE, etc.): <http://aws.amazon.com/tools/>

# Python 代码 --启动2个EC2 实例

```
runec2.py x
1  import boto.ec2
2
3  conn = boto.ec2.connect_to_region("cn-north-1")
4  conn.run_instances(
5      'ami-981d8fa1',
6      min_count=2,
7      max_count=2,
8      key_name='xiazhan-cn'
9      instance_type='t2.micro',
10     security_groups=['wslinux'],
11 )
12
```



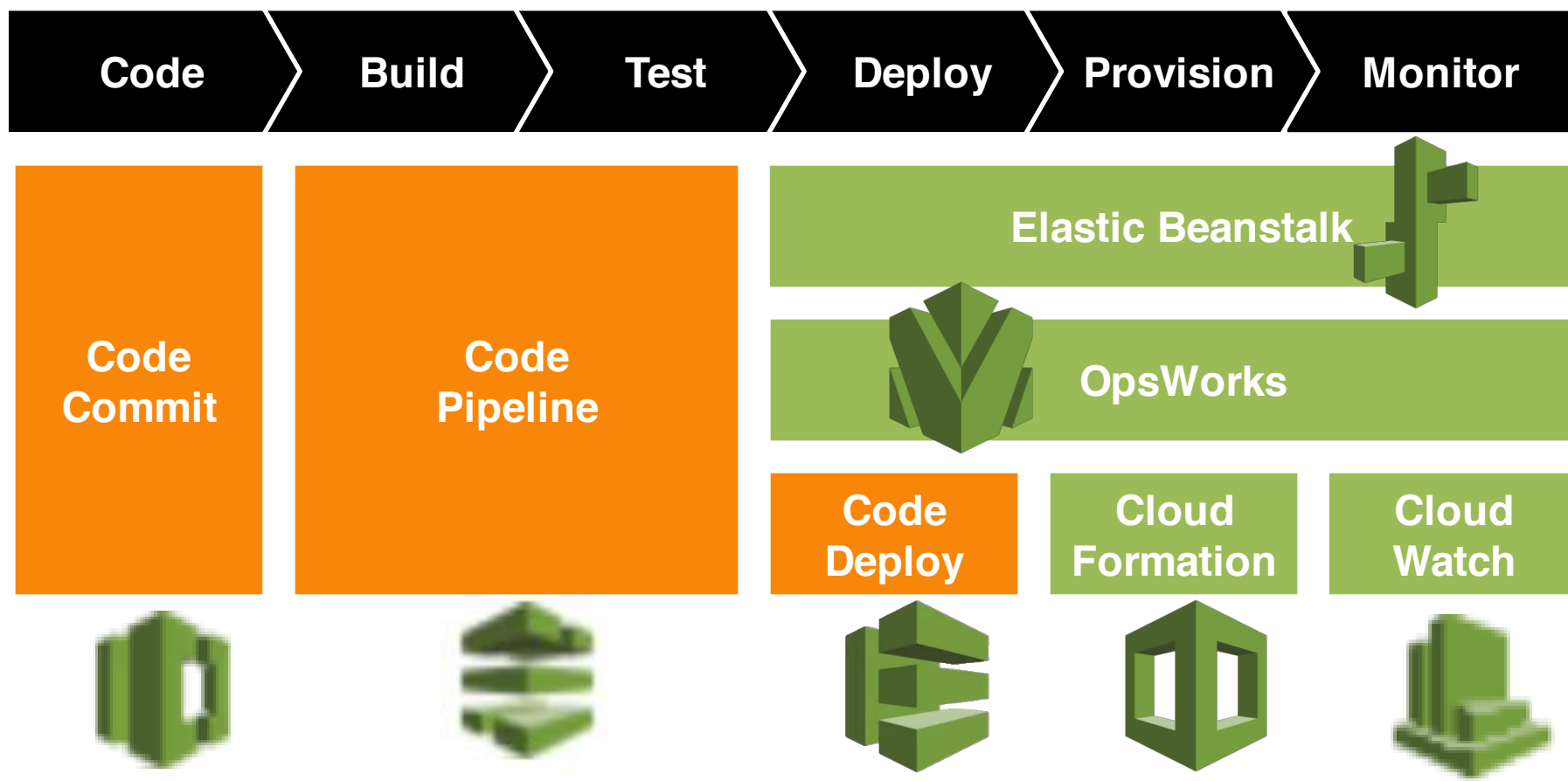
# 命令行CLI- 给EC2 实例加标签

```
aws ec2 create-tags  
--resources i-fffb064c7 i-8eb561b6  
-tags Key=Name,Value=ArchSummit
```

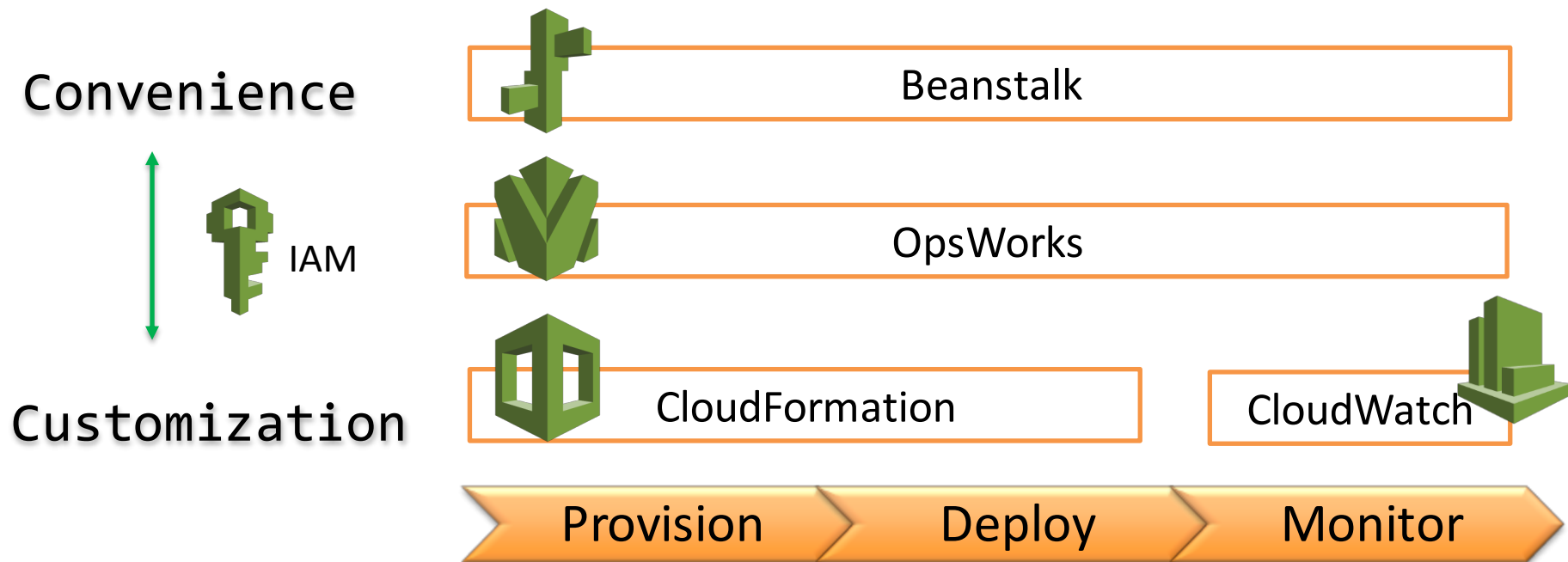
# 大规模基础架构的DevOps 需要框架和工具



# 基于AWS的DevOps实践要素

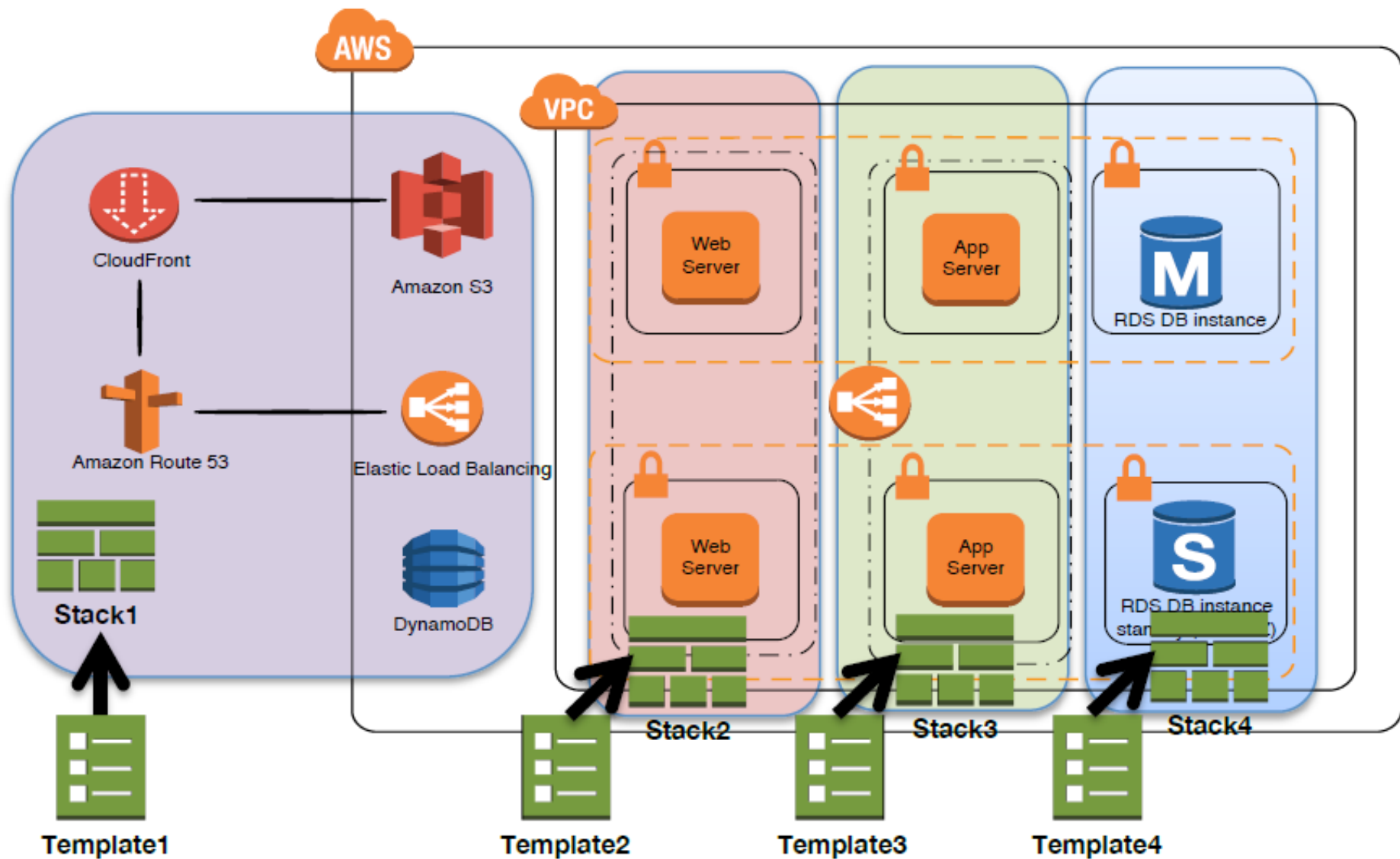


# 应用管理的服务

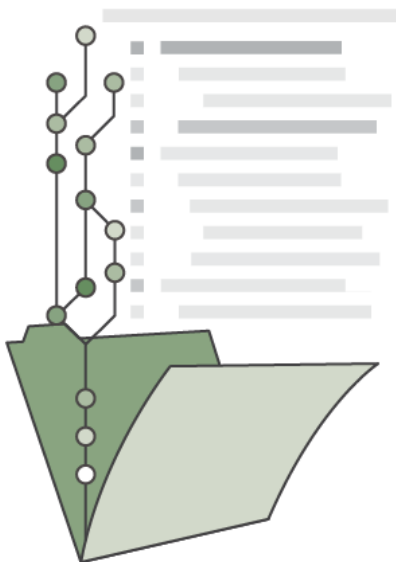


参见2015北京QCON: 《基于AWS的DevOps实践指南》 Dai Wen

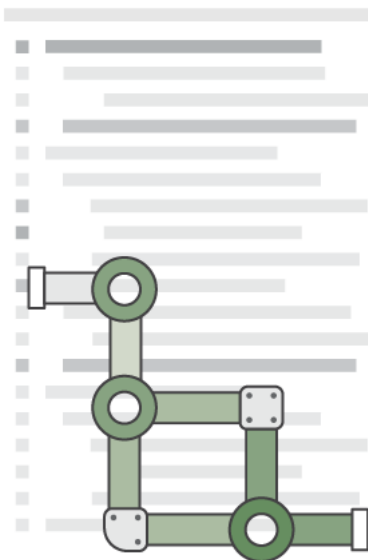
# 基于模板的快速部署



# AWS 代码服务



CodeCommit  
上周五发布



CodePipeline  
上周五发布



CodeDeploy  
Nov 2014 发布

[aws.amazon.com/awscode](https://aws.amazon.com/awscode)



# CodeCommit--托管的源代码控制服务



可扩展的托管型源代码控制服务，托管私有的 Git 存储库。

管理云中的代码库

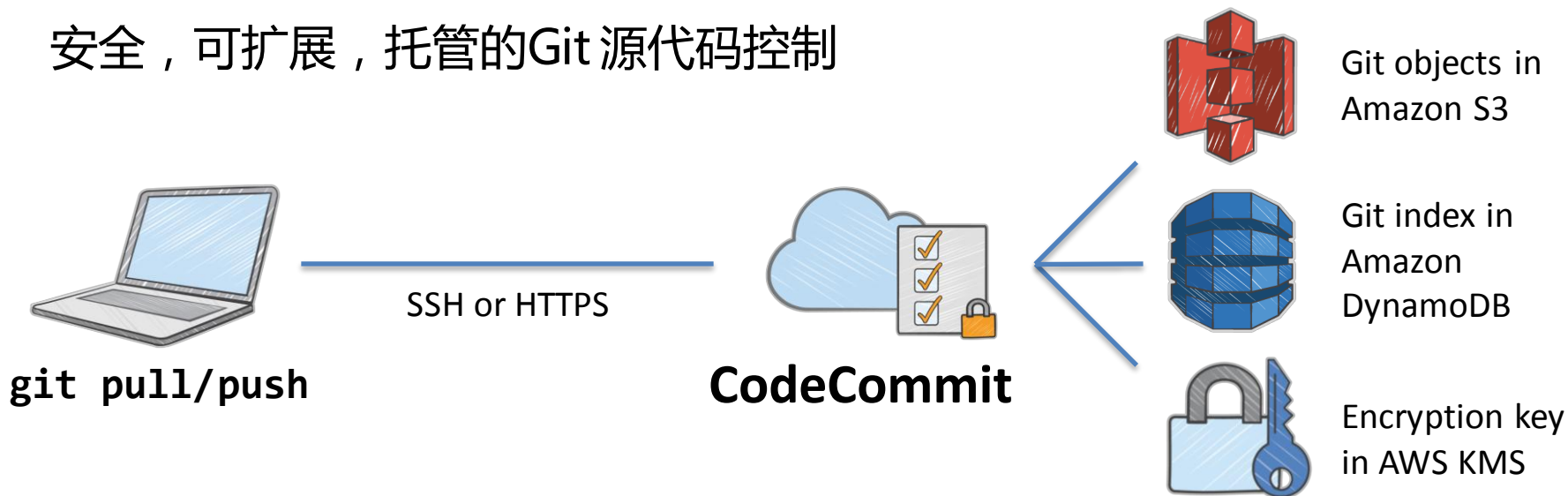
最接近开发，测试和生成环境

没有任何容量限制和文件的限制



# AWS CodeCommit

安全，可扩展，托管的Git 源代码控制



- 使用标准的Git 工具
- 具有Amazon S3可扩展，高可用，高持久
- 数据静止时可用客户专用秘钥加密
- 没有存储大小的限制

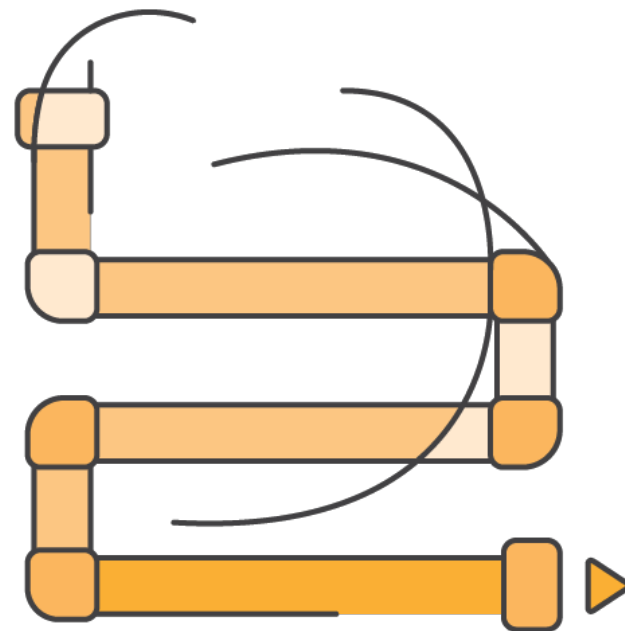
# CodePipeline--亚马逊自动交付分发服务

- 持续自动交付和发布服务，可以自行设计开发的工作流程进而到发行制作，可以定制 workflow 模型并可视化，也可以与已有的构建测试工具集成。

可重复的，自动化的集成和部署管道

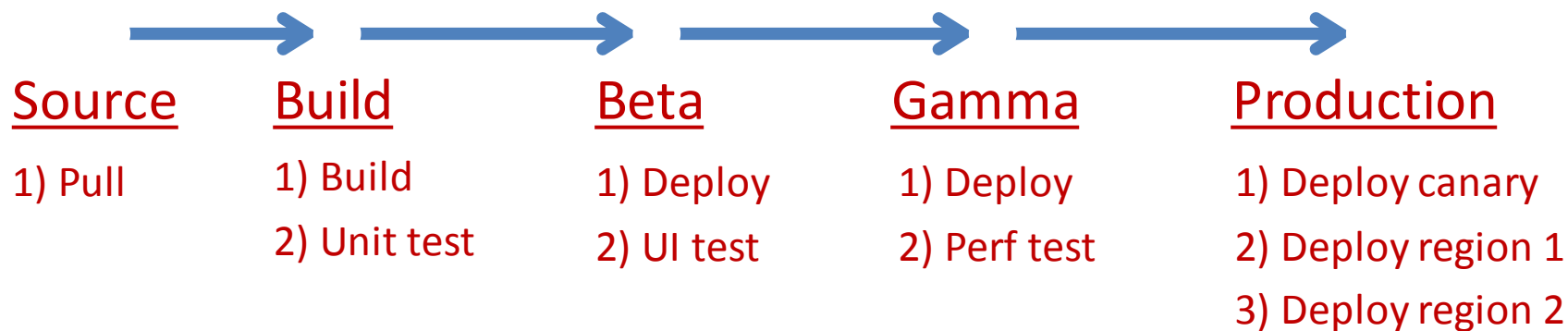
workflow 建模和可视化

80% 的开发团队这样进行集成和部署



# AWS CodePipeline

持续交付和发布自动化



- 建模管理你的发布过程
- 与你的系统整合
- 可视性的监控你的pipeline



# CodeDeploy-代码部署自动化服务

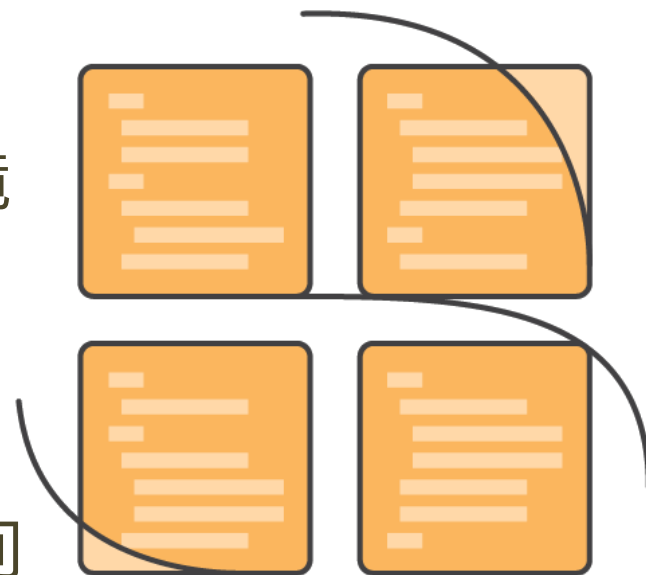


- 提供功能包括滚动更新、部署健康监测、回滚、集中化的管理、历史检索。支持的语言包括JS、Python、Java、C++、Shell脚本，可集成原有的工具链。

将代码快速部署到任何EC2上的环境

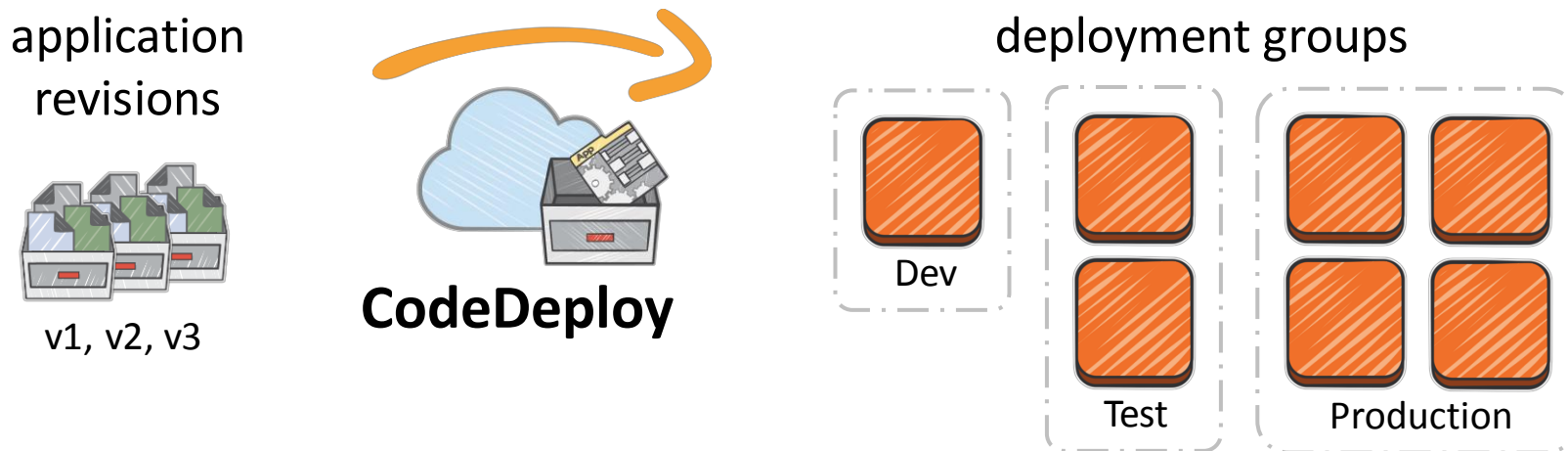
自动, 可复用且提供完全管理的服务

减少部署或应用错误带来的停机时间



# AWS CodeDeploy

## 协调自动化部署的过程



- 从单个实例扩展到上千个实例
- 部署到任何服务器上: AWS or on-premises
- 中央集中控制和监控
- 代码滚动升级

# 1) 包装 app    2) 指定目标    3) 部署!

version: 0.0

os: linux

files:

- source: chef/  
destination: /etc/chef/codedeploy
- source: target/hello.war  
destination: /var/lib/tomcat6/webapps

hooks:

ApplicationStop:

- location: deploy\_hooks/stop-tomcat.sh

BeforeInstall:

- location: deploy\_hooks/install-chef.sh
- location: deploy\_hooks/chef-solo.sh

ApplicationStart:

- location: deploy\_hooks/start-tomcat.sh

ValidateService:

- location: deploy\_hooks/verify\_service.sh



# 1) 包装 app    2) 指定目标    3) 部署!

version: 0.0

os: linux

files:

- source: chef/  
destination: /etc/chef/codedeploy
- source: target/hello.war  
destination: /var/lib/tomcat6/webapps

hooks:

ApplicationStop:

- location: deploy\_hooks/stop-tomcat.sh

BeforeInstall:

- location: deploy\_hooks/install-chef.sh
- location: deploy\_hooks/chef-solo.sh

ApplicationStart:

- location: deploy\_hooks/start-tomcat.sh

ValidateService:

- location: deploy\_hooks/verify\_service.sh

# 1) 包装 app    2) 指定目标    3) 部署!

**version: 0.0**

**os: linux**

**files:**

- **source: chef/  
destination: /etc/chef/codedeploy**
- **source: target/hello.war  
destination: /var/lib/tomcat6/webapps**

**hooks:**

**ApplicationStop:**

- **location: deploy\_hooks/stop-tomcat.sh**

**BeforeInstall:**

- **location: deploy\_hooks/install-chef.sh**
- **location: deploy\_hooks/chef-solo.sh**

**ApplicationStart:**

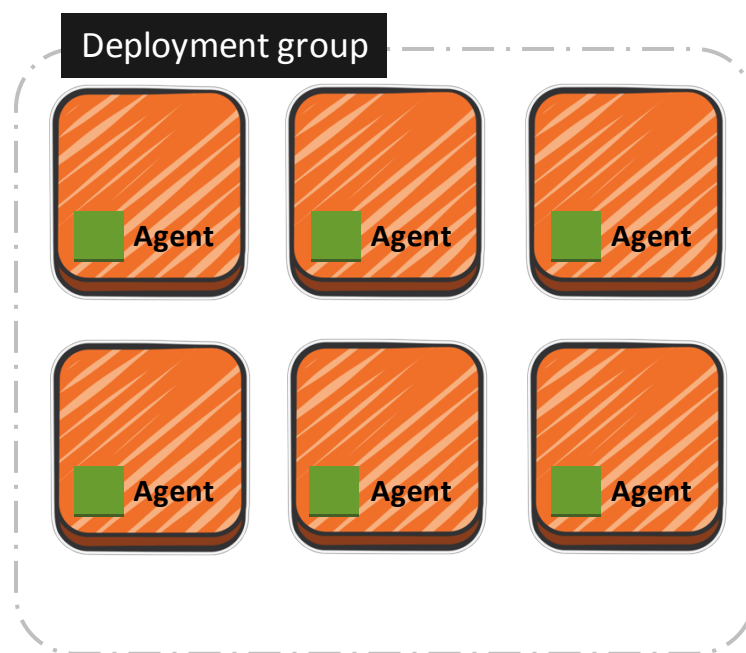
- **location: deploy\_hooks/start-tomcat.sh**

**ValidateService:**

- **location: deploy\_hooks/verify\_service.sh**



# 1) 包装 app    2) 指定目标    3) 部署!



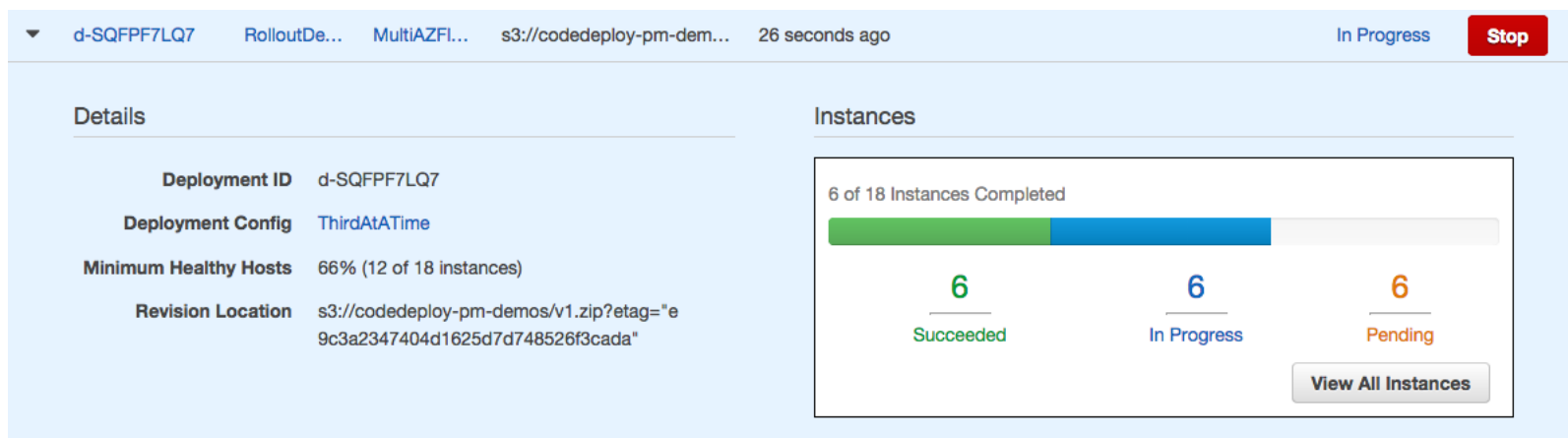
将实例按下述编组:

- Auto Scaling group
- Amazon EC2 tag
- On-premises tag

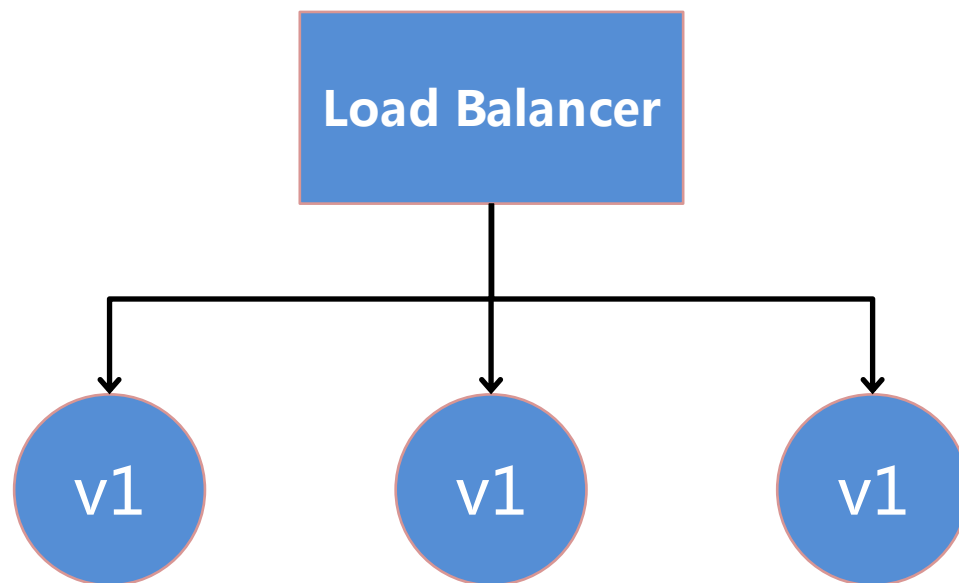
# 1) 包装 app    2) 指定目标    3) 部署!

**AWS CLI & SDKs**  
**AWS Console**  
**CI / CD Partners**  
**GitHub**

```
aws deploy create-deployment \  
--application-name MyApp \  
--deployment-group-name TargetGroup \  
--s3-location bucket=MyBucket,key=MyApp.zip
```

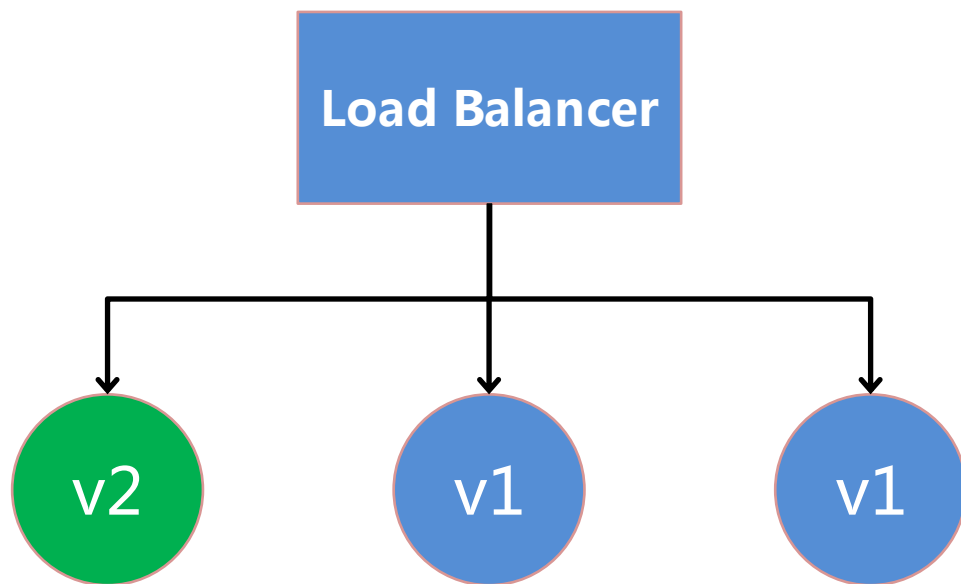


# 滚动升级—不下线部署

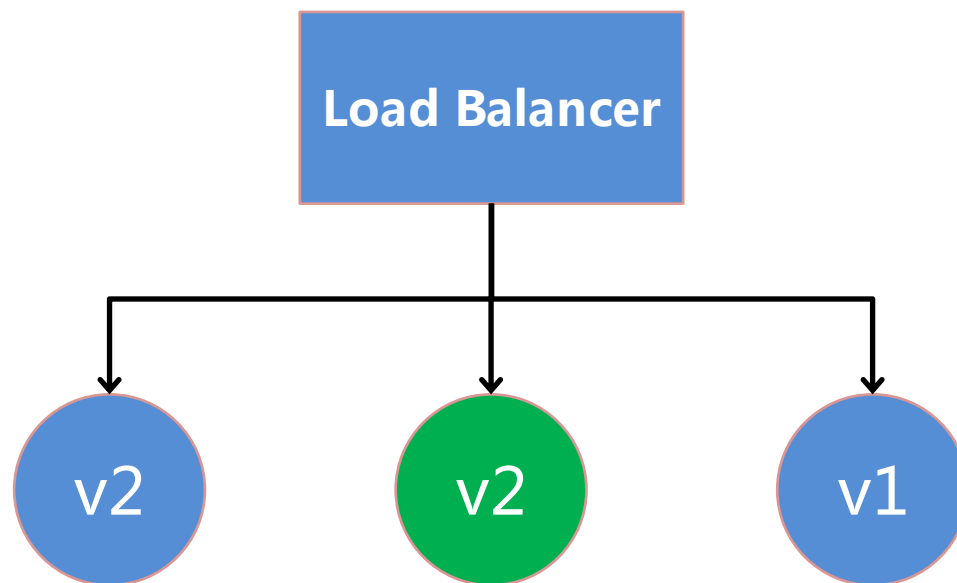




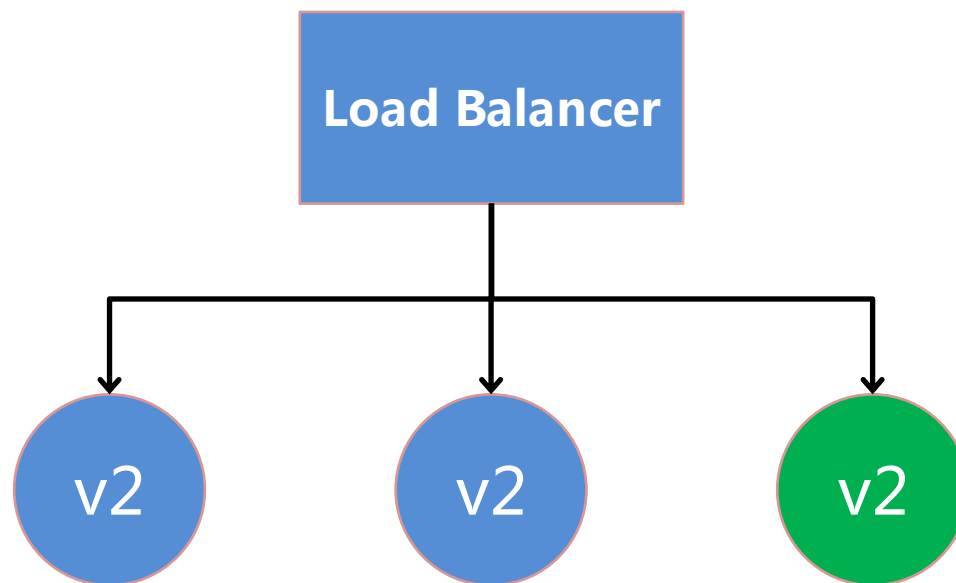
# 滚动升级—不下线部署



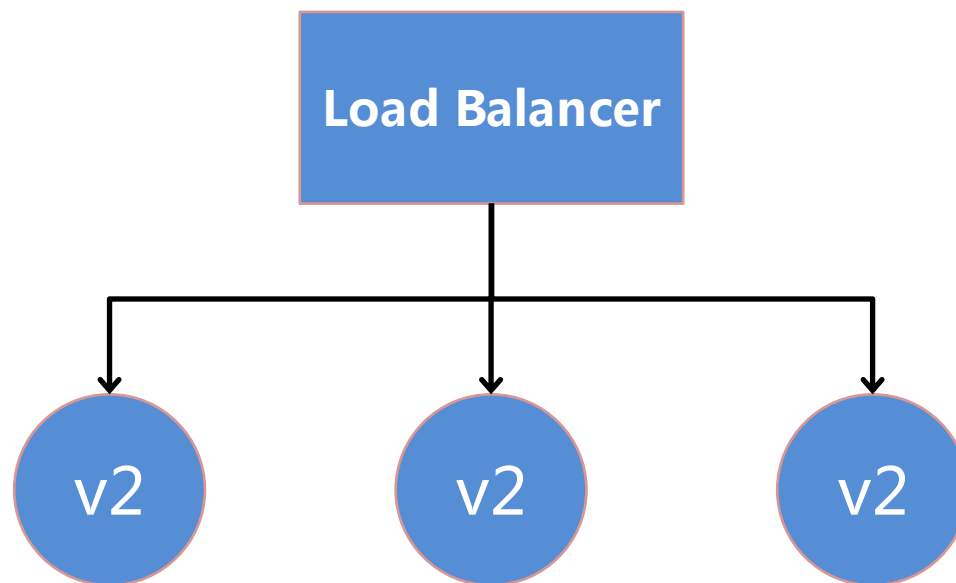
# 滚动升级—不下线部署



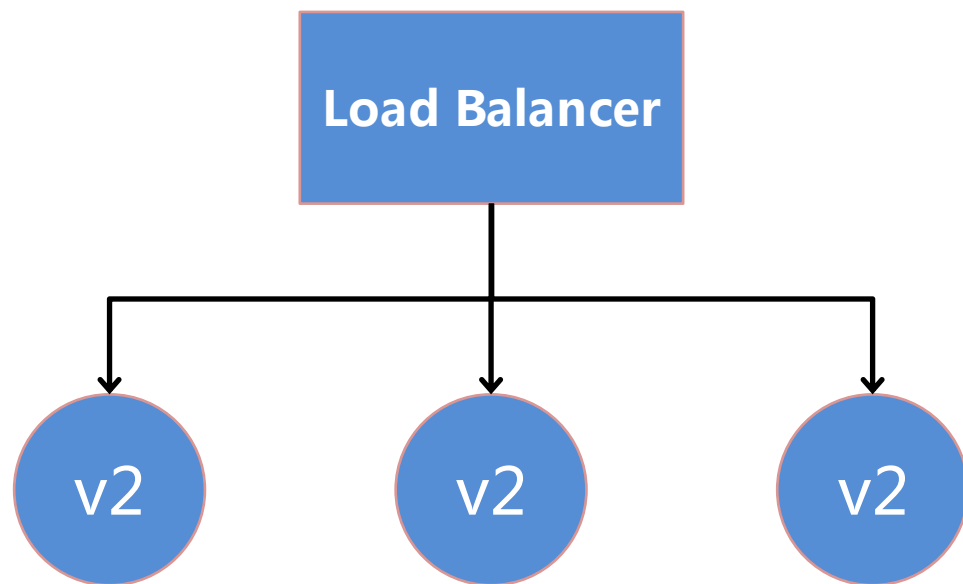
# 滚动升级—不下线部署



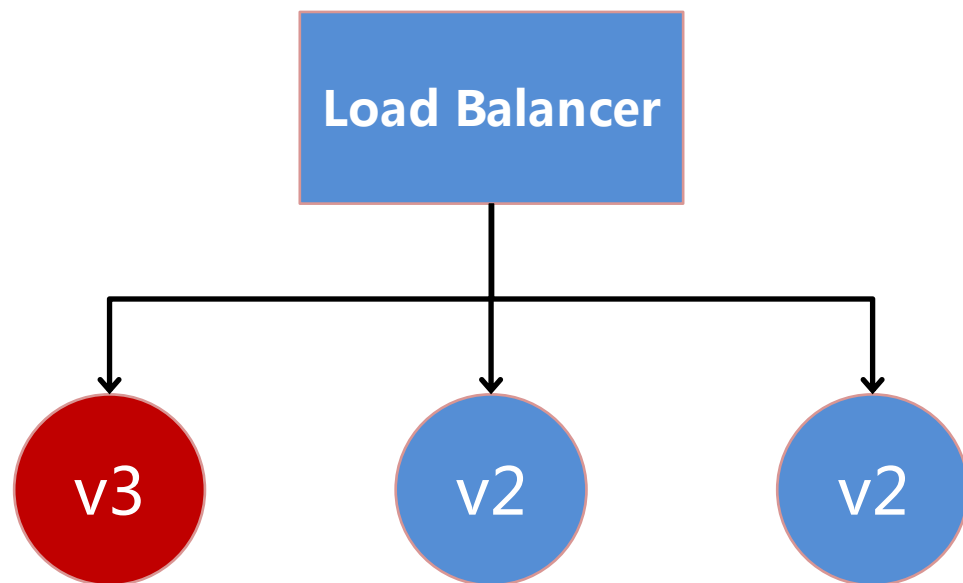
# 滚动升级—不下线部署



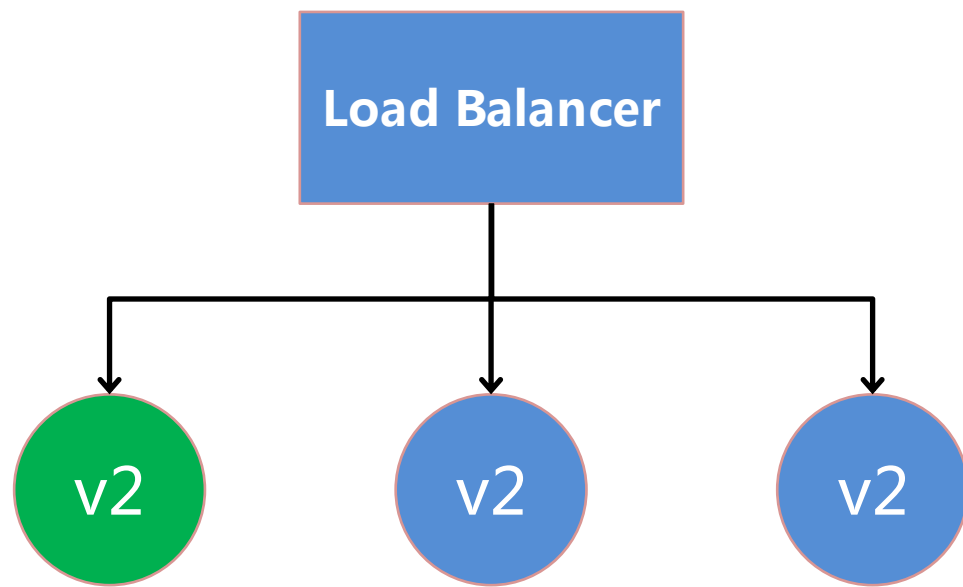
# 健康检查—捕获任何部署问题



# 健康检查—捕获任何部署问题

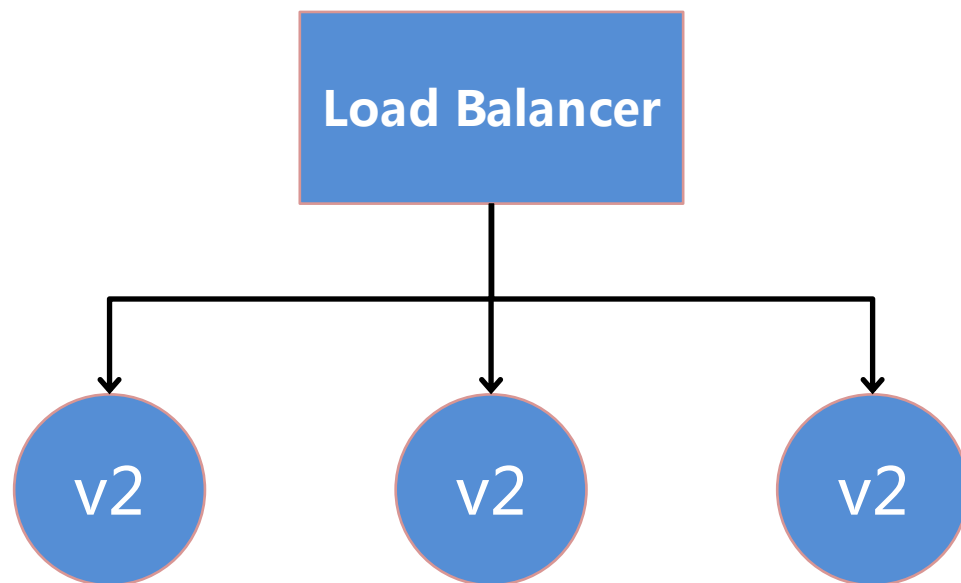


# 健康检查—捕获任何部署问题



混滚

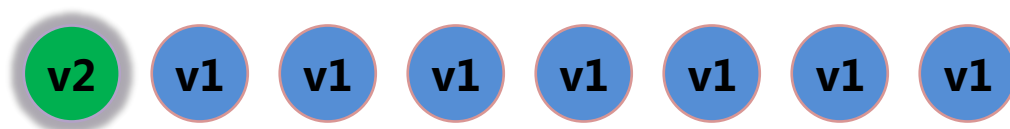
# 健康检查—捕获任何部署问题





# 不同的部署模式—选择速度

一次一个



一次一半



全部一起



# 代码服务与现有合作伙伴方案集成整合











# Demo

## Amazon Web Services





### Compute

-  **EC2**  
Virtual Servers in the Cloud
-  **Lambda**  
Run Code In Response to Events
-  **EC2 Container Service**  
Run and Manage Docker Containers




### Storage & Content Delivery

-  **S3**  
Scalable Storage In the Cloud
-  **Elastic File System** PREVIEW  
Fully Managed File System for EC2
-  **Storage Gateway**  
Integrates On-Premises IT Environments with Cloud Storage
-  **Glacier**  
Archive Storage In the Cloud
-  **CloudFront**  
Global Content Delivery Network








### Database

-  **RDS**  
MySQL, Postgres, Oracle, SQL Server, and Amazon Aurora
-  **DynamoDB**  
Predictable and Scalable NoSQL Data Store
-  **ElastiCache**  
In-Memory Cache
-  **Redshift**  
Managed Petabyte-Scale Data Warehouse Service







### Networking

-  **VPC**  
Isolated Cloud Resources
-  **Direct Connect**  
Dedicated Network Connection to AWS
-  **Route 53**  
Scalable DNS and Domain Name Registration





### Administration & Security

-  **Directory Service**  
Managed Directories In the Cloud
-  **Identity & Access Management**  
Access Control and Key Management
-  **Trusted Advisor**  
AWS Cloud Optimization Expert
-  **CloudTrail**  
User Activity and Change Tracking
-  **Config**  
Resource Configurations and Inventory
-  **CloudWatch**  
Resource and Application Monitoring
-  **Service Catalog**  
Personalized Catalog of AWS Resources








### Deployment & Management

-  **Elastic Beanstalk**  
AWS Application Container
-  **OpsWorks**  
DevOps Application Management Service
-  **CloudFormation**  
Templated AWS Resource Creation
-  **CodeDeploy**  
Automated Deployments
-  **CodeCommit**  
Managed Git Repositories
-  **CodePipeline**  
Continuous Delivery





### Analytics

-  **EMR**  
Managed Hadoop Framework
-  **Kinesis**  
Real-time Processing of Streaming Big Data
-  **Data Pipeline**  
Orchestration for Data-Driven Workflows
-  **Machine Learning**  
Build Smart Applications Quickly and Easily




### Application Services

-  **SQS**  
Message Queue Service
-  **SWF**  
Workflow Service for Coordinating Application Components
-  **AppStream**  
Low Latency Application Streaming
-  **Elastic Transcoder**  
Easy-to-use Scalable Media Transcoding
-  **SES**  
Email Sending Service
-  **CloudSearch**  
Managed Search Service
-  **API Gateway**  
Build, Deploy and Manage APIs

### Mobile Services

-  **Cognito**  
User Identity and App Data Synchronization
-  **Device Farm**  
Test Android and Fire OS apps on real devices In the Cloud
-  **Mobile Analytics**  
Understand App Usage Data at Scale
-  **SNS**  
Push Notification Service

### Enterprise Applications

-  **WorkSpaces**  
Desktops In the Cloud
-  **WorkDocs**  
Secure Enterprise Storage and Sharing Service
-  **WorkMail** PREVIEW  
Secure Email and Calendaring Service

# One More Thing...



# 推出 Amazon API Gateway



- ✓ 管理多版本多阶段的 APIs
- ✓ 向开发者创建分发API keys
- ✓ 使用Signature Version 4 来认证以接入调用APIs
- ✓ 控制监控API请求以保护后端
- ✓ 利用事件触发服务 AWS Lambda



# One More Thing...



# 总结

- 开发运维DevOps和微服务Microservices的概念
- 亚马逊DevOps和微服务的实践
- AWS提供的DevOps的框架和服务

一切皆软件，一切皆API！



# Thanks!

关于亚马逊AWS，更多请访问

网站 <http://www.amazonaws.cn>

博客 [blog.csdn.net/awschina](http://blog.csdn.net/awschina)

微博 [www.weibo.com/amazonaws](http://www.weibo.com/amazonaws)

微信 AWS China

