

WOTA

51CTO

World Of Tech 2017

全球架构与运维技术峰会

2017年4月14日-15日 北京富力万丽酒店

ARCHITECTURE

云时代架构和运维的新趋势

张侠 博士
AWS首席云计算企业顾问

北京
2017年4月14日

日程

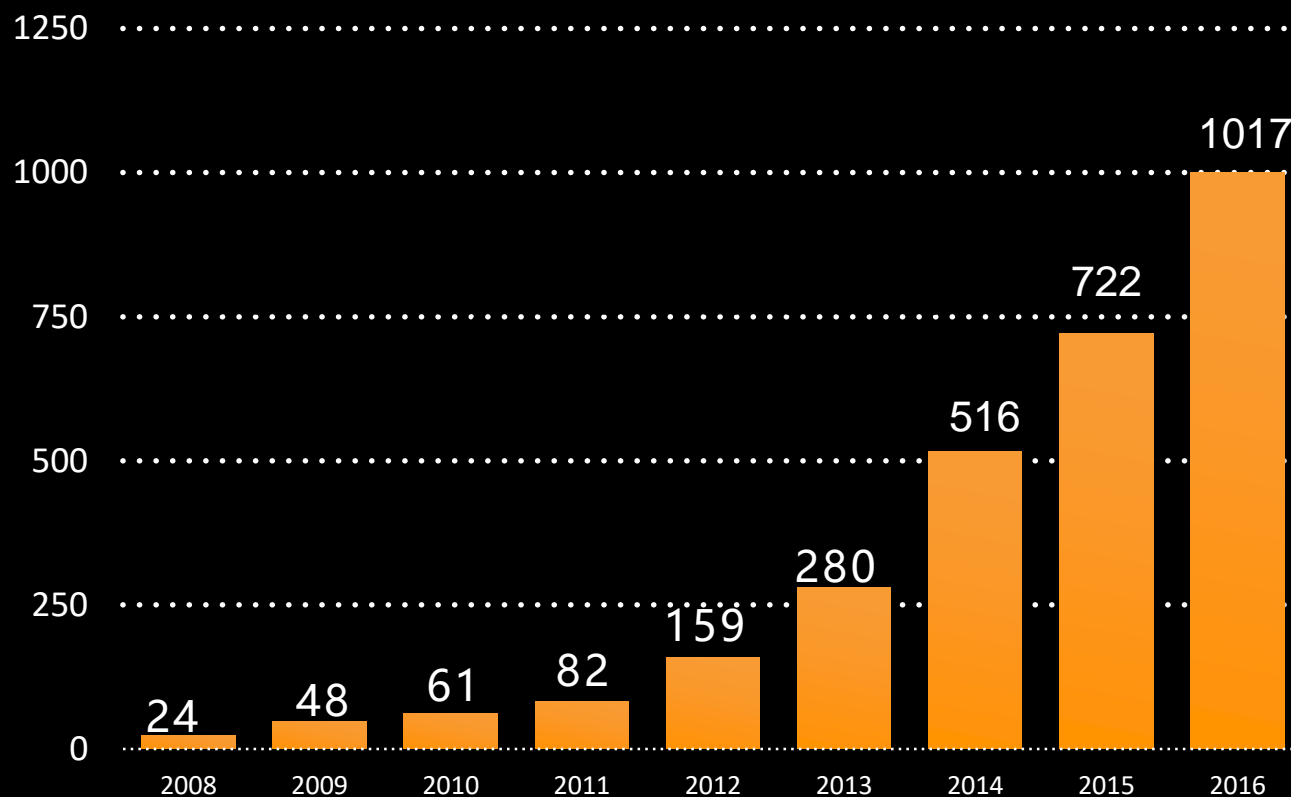
- 云计算与 AWS 的进展
- 从虚拟机到容器和函数
- AWS 支持 DevOps

AWS 云计算平台

- 客户支持
- 支持
- 管理服务
- 专业服务
- 合作伙伴生态
- 培训 & 认证
- 解决方案架构师
- 客户经理
- 安全 & 价格报表
- 客户技术经理

Marketplace	管理工具	数据分析	开发工具	人工智能	物联网	移动	企业应用	游戏开发
Business Applications	Monitoring	Query Large Data Sets						
DevOps Tools	Auditing	Elasticsearch						
Business Intelligence	Service Catalog	Business Analytics						
Security	Server Management	Hadoop/Spark			Rules Engine	Build, Test, Monitor Apps	Document Sharing	
Networking	Configuration Tracking	Real-time Data Streaming	Private Git Repositories	Voice & Text Chatbots	Local Compute and Sync	Push Notifications	Email & Calendaring	
Database & Storage	Optimization	Orchestration Workflows	Continuous Delivery	Machine Learning	Device Shadows	Build, Deploy, Manage APIs	Hosted Desktops	
SaaS Subscriptions	Resource Templates	Managed Search	Build, Test, and Debug	Text-to-Speech	Device Gateway	Device Testing	Application Streaming	3D Game Engine
Operation System	Automation	Managed ETL	Deployment	Image Analysis	Registry	Identity	Backup	Multi-player Backends
迁移	Application Discovery	Application Migration	Data Migration	Database Migration	Server Migration			
混合架构	Data Integration	Integrated Networking	Identity Federation	Resource Management	VMware on AWS	Devices & Edge Systems		
应用服务	Transcoding	Step Functions	Messaging					
安全	Identity & Access	Key Storage & Management	Active Directory	DDoS Protection	Application Analysis	Certificate Management	Web App. Firewall	
数据库	Aurora	MySQL	PostgreSQL	Oracle	SQL Server	MariaDB	Data Warehousing	NoSQL
存储	Object Storage	Archive	Exabyte-scale Data Transport	Block Storage	Managed File Storage			
计算	Virtual Machines	Simple Servers	Web Applications	Auto Scaling	Batch	Containers	Event-driven Computing	
网络	Isolated Resources	Dedicated Connections	Global CDN	Load Balancing	Scalable DNS			
基础设施	Regions	Availability Zones	Points of Presence					

AWS每年推出的新服务和功能



截止到4月1日，在2017年推出了236项新的功能



首选的技术平台

- 功能的深度和广度，持续的快速创新
- 最多用户和合作伙伴组成的生态系统
- 最长的运维经验带来稳定的云服务

云消除了对快速和敏捷的束缚

快速发展的开发趋势



降低风险



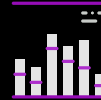
专注的小应用



快速交付



响应客户需求



快速尝试

开发和测试是体现敏捷性的环节



不受限制
地访问资源



测试真实
反映实际情况



更快的
上市速度



生产效率
大幅改善



成本显著
降低

最佳架构框架



安全性



可靠性



性能高效



成本优化



运营卓越

从虚拟机到容器到函数

云计算时代, 计算在不断创新和发展

云计算的发展阶段



虚拟机



容器

$f(x)$

函数

三个发展阶段的对比

虚拟机的世界



AMI



数小时到数月



打补丁



多线程
多任务



每台虚拟机 每小时

容器的世界



容器文件



数分钟到数天



版本控制



多线程
单任务



每个虚拟机 每小时

Lambda 函数的世界



代码



几微秒到几秒



版本控制



单线程
单任务



每内存/秒
每个请求
免费套餐

完善的虚拟机生态系统



VPC



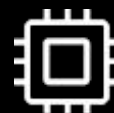
Auto Scaling



RDS

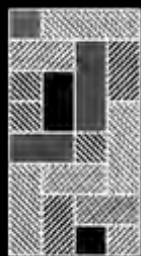


EBS



ELB

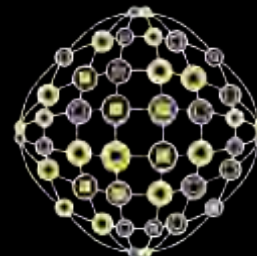
Amazon.com 的案例



单一应用程序



SOA 服务



微服务

容器的兴起和向微服务的迁移

- ▶ 开发过程的输出是小的功能性的模块
- ▶ 最大程度地减小关联性并降低关联性管理的复杂度



基于容器的操作的挑战



容器管理



管道集成



基础设施管理



资源访问



安全性



任务安排



服务发现



高可用性

ECS 是在生产环境中运行容器的最佳方式



容器管理



容器注册表



无需管理
基础设施



访问 EBS、ELB
和 CloudWatch



与 IAM 的集成



批处理和长时间
运行任务安排

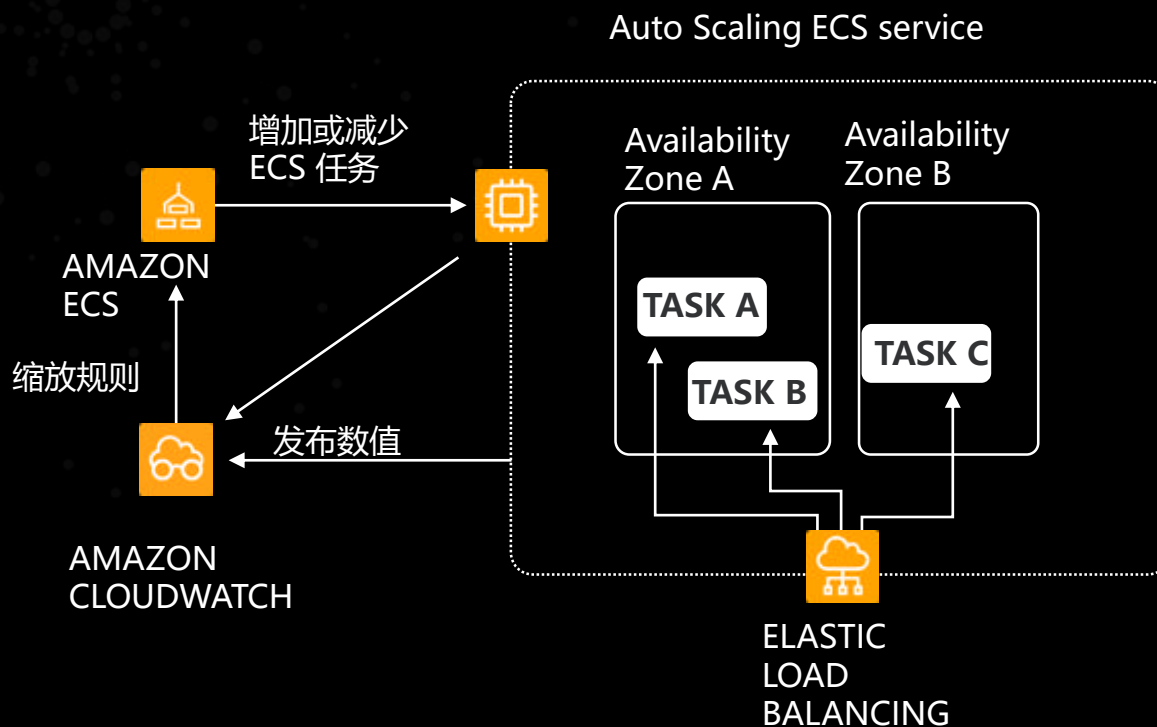


利用多可用区

ECS 自动缩放服务



Amazon ECS



针对任务的 IAM 角色

```
{  
  "family": "signup-app",  
  "taskRoleArn": "arn:aws:iam::123456789012:role/DynamoDBRoleForTask", "volumes": [],  
  "containerDefinitions": [{  
    "environment": [ ... ],  
    "name": "signup-web",  
    "mountPoints": [],  
    "image": "amazon/signup-web",  
    "cpu": 25,  
    "portMappings": [ ... ],  
    "entryPoint": [ ... ],  
    "memory": 100,  
    "essential": true,  
    "volumesFrom": []  
  }]  
}
```



无服务器计算

管理任何服务器都不如
无需管理服务器更加简单



Lambda 的现状



Amazon
Lambda

语言



Node.js
(JavaScript)



Python



Java
(与 Java 8 兼容)

触发器



S3 存储桶



DynamoDB 表



Kinesis 流



SNS 通知

独木不成林，独兽不成群



Lambda



API Gateway



S3



CloudFront



DynamoDB



Kinesis



ElastiCache



CloudSearch



Elasticsearch



SES



SNS



SQS

Amazon API Gateway

在数分钟内创建强健、可扩展且安全的 API



版本控制



计量和限制



缓存



签名和授权

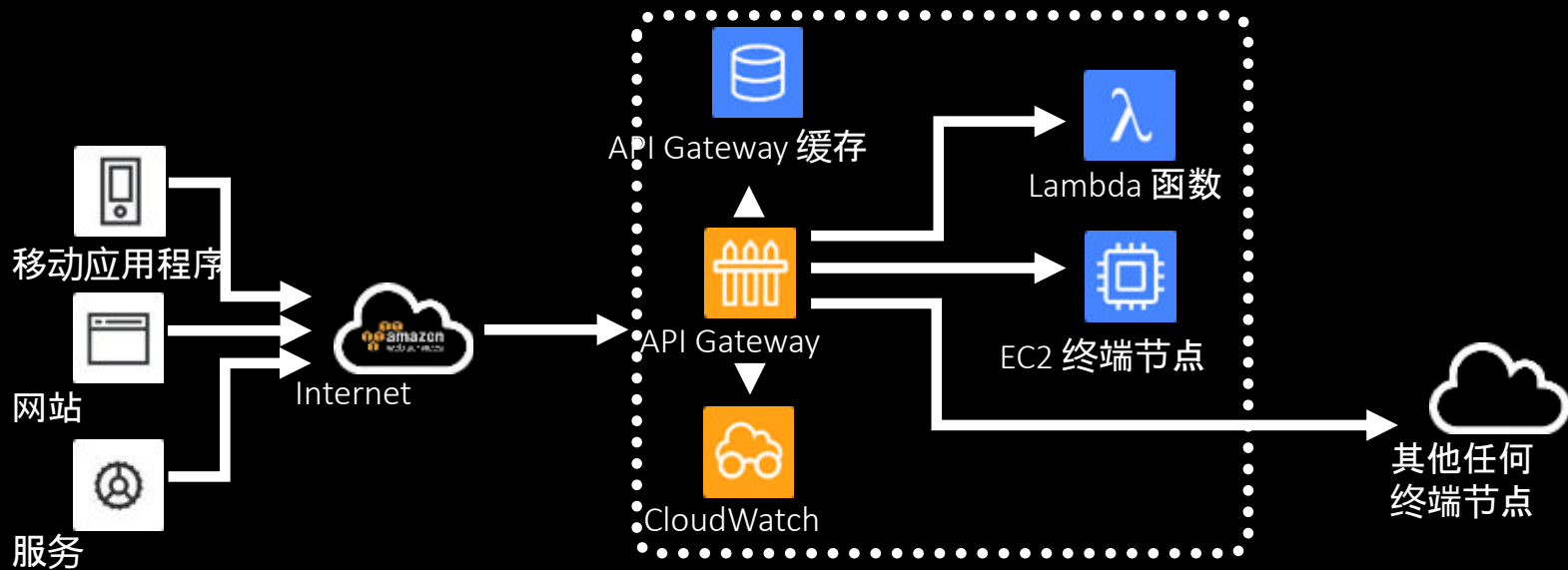


SDK 生成

Amazon API Gateway



Amazon API
Gateway

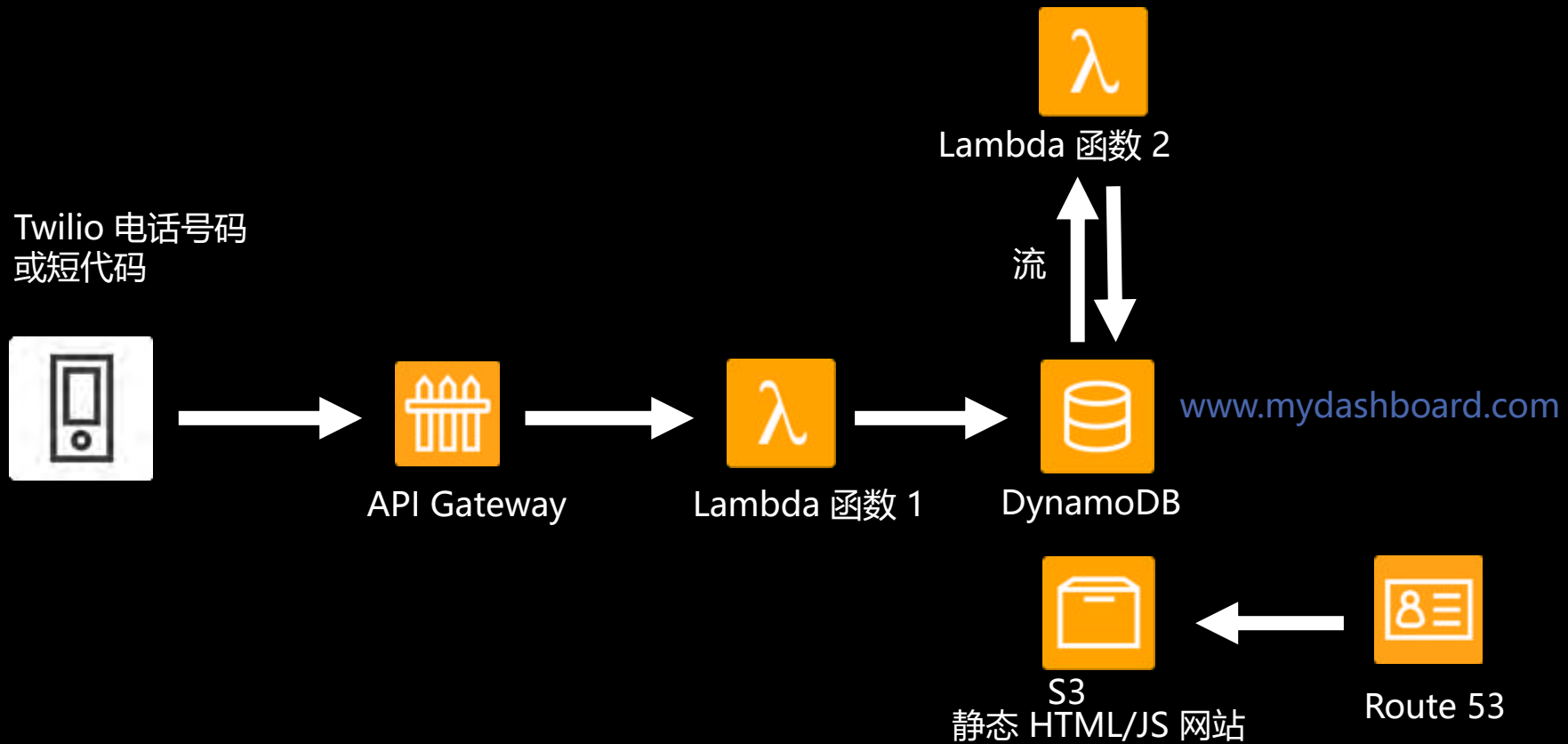


可执行的无服务器应用程序的参考架构

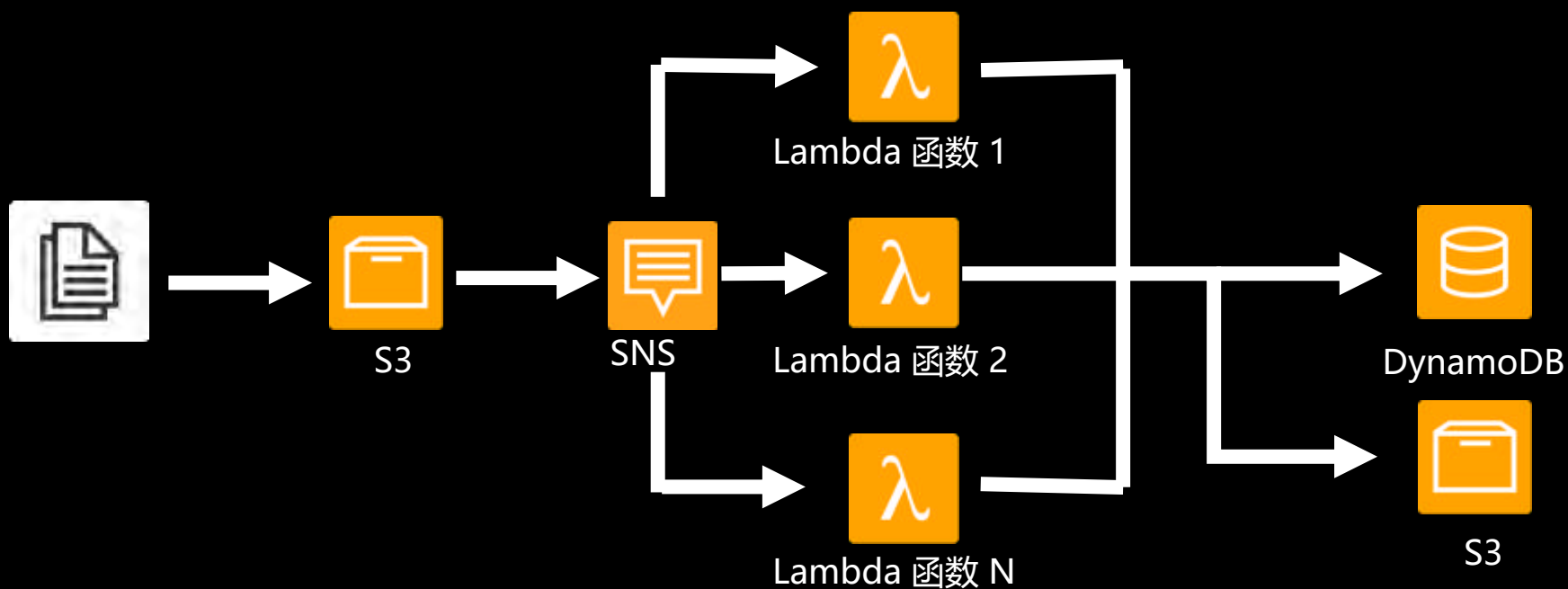


github.com/awslabs

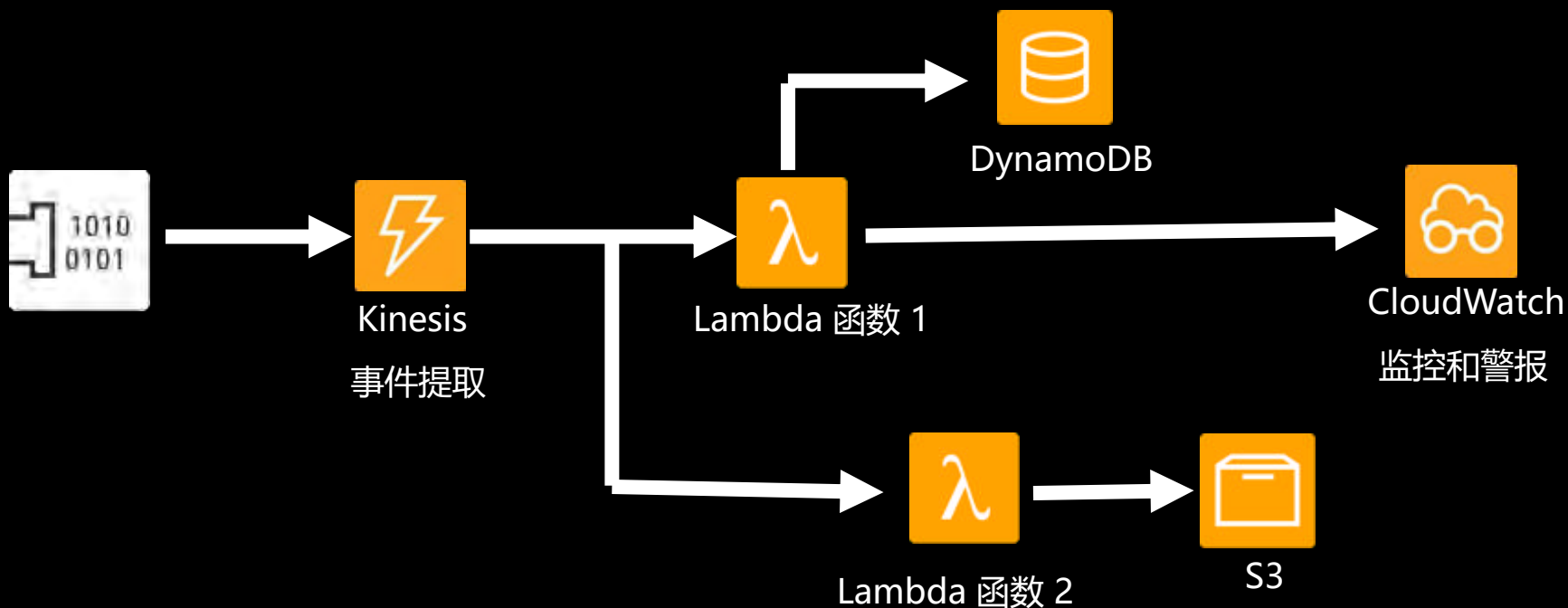
无服务器 Web 应用程序参考架构



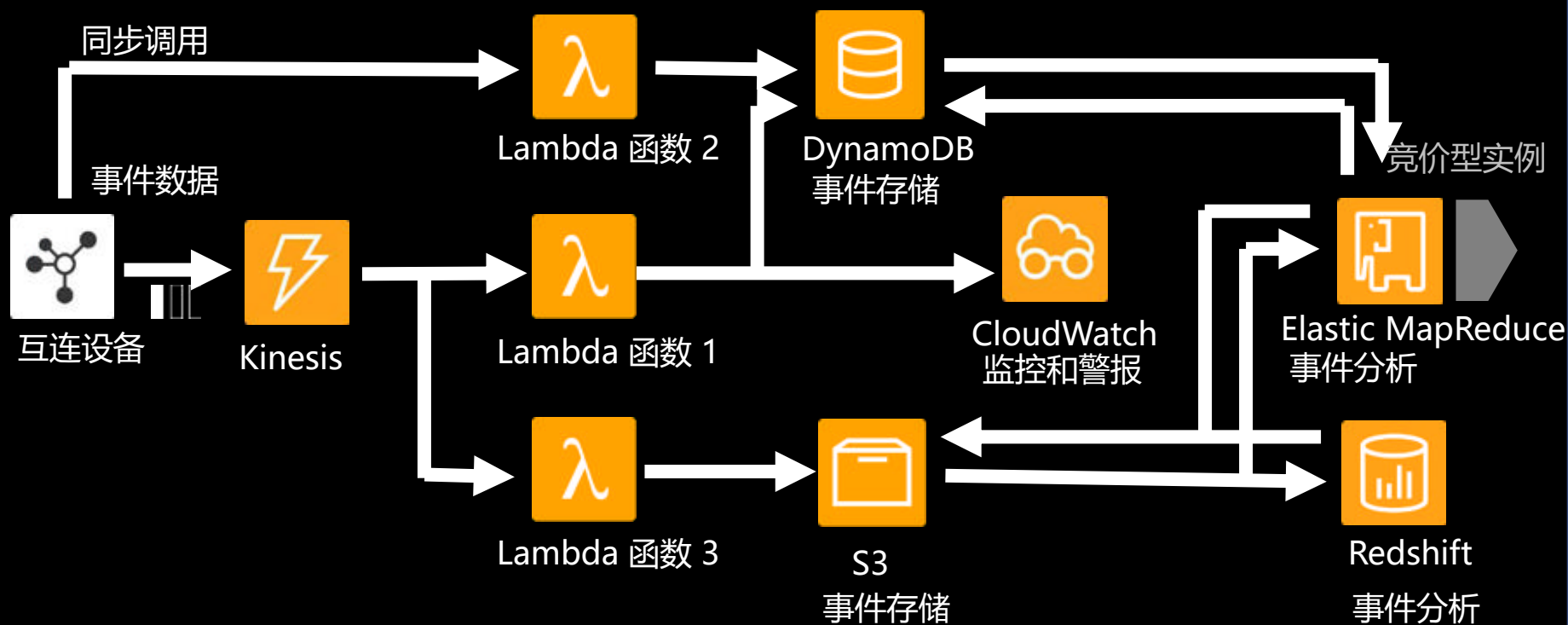
无服务器文件处理参考架构



无服务器流数据处理参考架构



无服务器 IoT 后端参考架构



AWS 全面支持 DevOps

软件交付模式已完全改变

旧软件交付

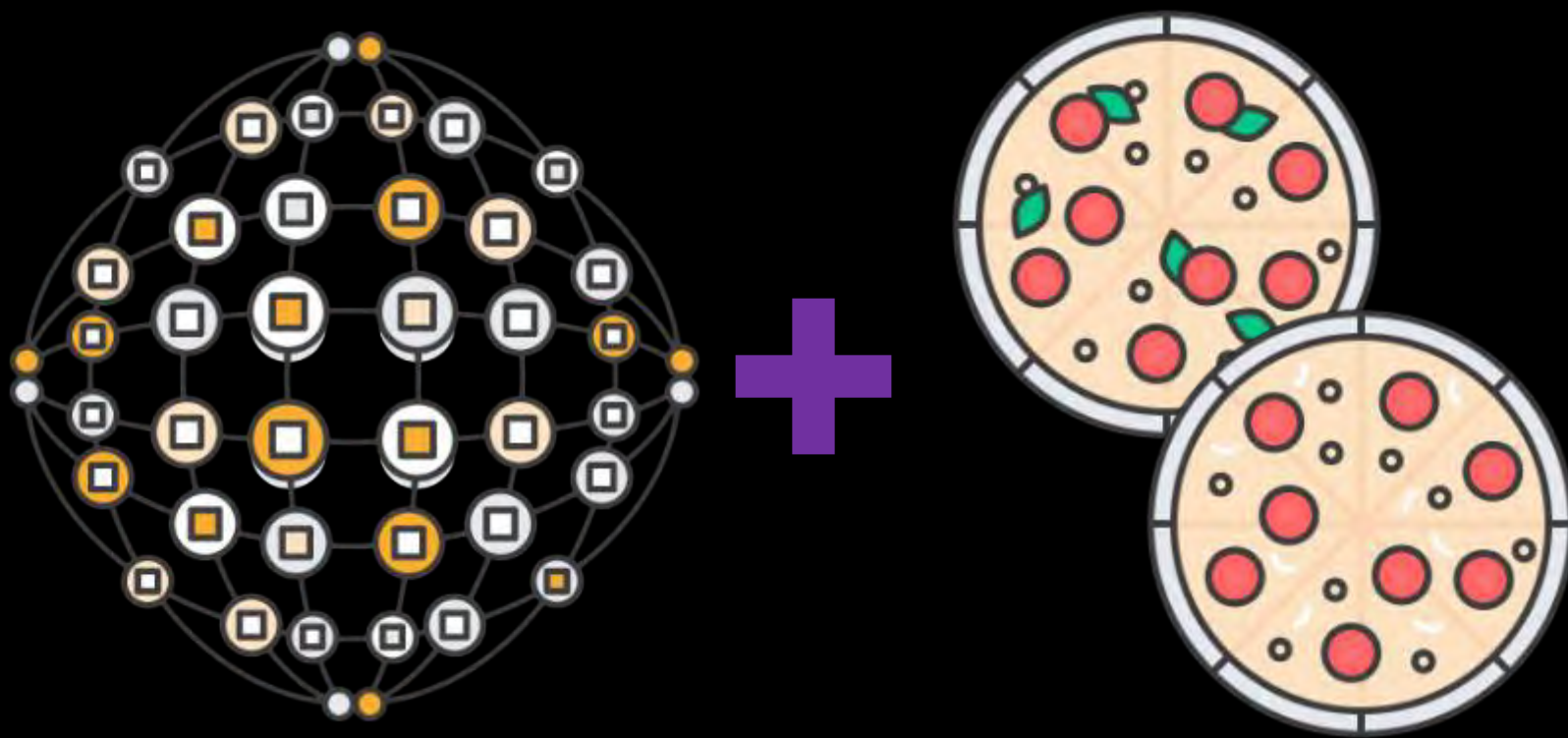


新软件交付



软件快速生成，发布和交付

- 创业公司不需太多投资即可挑战软件巨头
- 把软件发布给上百万用户只需一个点击下载
- 快速对于保持领先防止颠覆至关重要



微服务 + 双比萨饼团队

DevOps: 文化 + 实践+ 工具

每个双比萨饼团队掌控自己的产品

- 自主定制产品
- 管理产品的质量
- 处理相关问题
- 谁建设谁运维
- 提供支持，跟踪业务和技术指标以实现目标



DevOps: 文化 + 实践 + 工具

每个双比萨饼团队在符合标准的情况下自主确定有关实践

- Agile? Scrum? 每天还是每周发布？
只要你的团队觉得适用
- 没有统一审批变化管理的委员会或团队，但通用工具的选择需要一定的审批流程



DevOps: 文化 + 实践+ 工具

提供给每个双比萨饼团队下列工具箱，
使用其中的工具或适当改进：

- 任何花在运维的时间将减少花在开发上的时间
- 减少花在开发上的时间将增加项目不达目标的风险
- 工具本身帮助规范实现最佳实践
- 工具由其它双比萨饼团管理维护



软件管理流程几个主要阶段



- 源代码检入
，例如.java
文件
检查评估代
码
- 比较代码
• 单元测试
• 风格检查
• 代码度量
• 生成镜像
- 集成车市
• 压力测试
• UI测试
• 渗透测试
- 配置架构
- 部署生产
环境
- 监控，反馈，
改进

持续集成

持续交付，部署，管理，监控，改进

AWS支持DevOps的开发管理和应用部署服务

Source
源码

Build
生成

Test
测试

Provision
配置

Deploy
部署

Monitor
监控

持续集成



AWS
CodeCommit



AWS
CodeBuild



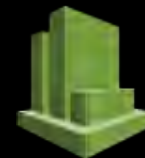
AWS
CodePipeline



AWS
CodeDeploy



AWS
CloudFormation



Amazon
CloudWatch

持续交付，部署，管理，监控，改进

GitHub



Jenkins



AWS

CloudFormation



AWS

OpsWorks



AWS Elastic
Beanstalk



Amazon ECS

AWS CodePipeline



AWS
CodePipeline



通过持续交付快速可靠的实现应用升级

建模并可视化软件交付流程

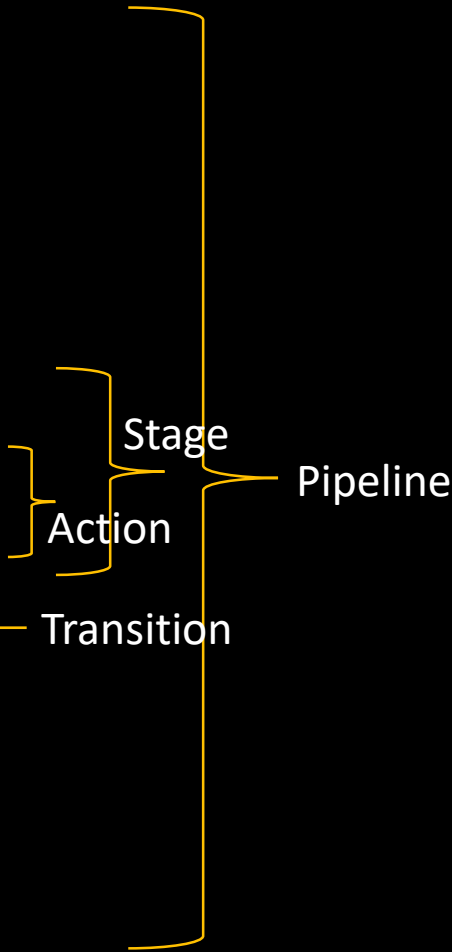
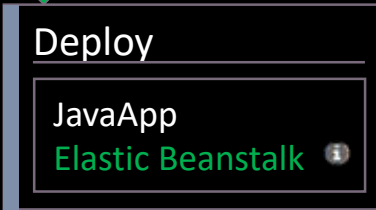
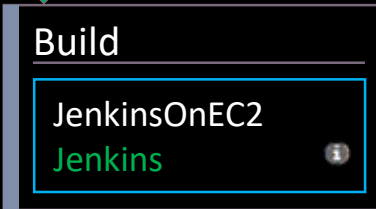
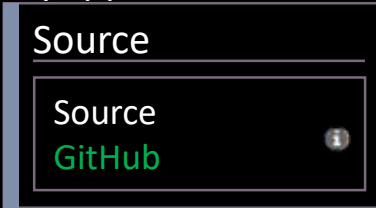
每次源代码有变化时，建立，测试和部署代码

实现AWS与第三方工具的整合



CodePipeline

MyApplication





CodePipeline

MyApplication

Source

Source
GitHub



Build

JenkinsOnEC2
Jenkins

NotifyDevelopers
Lambda

Parallel actions



Deploy

JavaApp
Elastic Beanstalk



CodePipeline

MyApplication

Source

Source
GitHub



Build

JenkinsOnEC2
Jenkins

NotifyDevelopers
Lambda

TestAPI
Runscope

Sequential actions



Deploy

JavaApp
Elastic Beanstalk

CodePipeline

MyApplication

Build

JenkinsOnEC2
Jenkins ⓘ



Staging-Deploy

JavaApp
Elastic Beanstalk ⓘ

QATeamReview
Manual Approval ⓘ

Review

Manual approvals



Prod-Deploy

JavaApp
Elastic Beanstalk ⓘ

CodePipeline

MyApplication

Source

Source
GitHub

Build

JenkinsOnEC2
Jenkins

Deploy

JavaApp
Elastic Beanstalk



Java App

Elastic Beanstalk
Web container



1. Get changes

2. Store source artifact

3. Poll for job

4. Acknowledge job

7. Put success

5. Get source artifact

6. Store build artifact

8. Retrieve build artifact

9. Deploy build artifact

Source Artifact

Build Artifact

AWS CodeDeploy



AWS
CodeDeploy

自动将源代码部署到任何实例上

处理应用升级的复杂性

避免应用部署时的下线时间

如故部署失败自动回滚

用任何语言在任何操作系统上部署到云上或机房里的虚机上

整合AWS与第三方工具



appspec.yml 实例

```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/html
permissions:
  - object: /var/www/html
    pattern: "*.html"
    owner: root
    group: root
    mode: 755
hooks:
  ApplicationStop:
    - location: scripts/deregister_from_elb.sh
  BeforeInstall:
    - location: scripts/install_dependencies.sh
  ApplicationStart:
    - location: scripts/start_httpd.sh
  ValidateService:
    - location: scripts/test_site.sh
    - location: scripts/register_with_elb.sh
```

appspec.yml 实例

```
version: 0.0
os: linux
files:
  - source: /
    destination: /var/www/html
permissions:
  - object: /var/www/html
    pattern: "*.html"
    owner: root
    group: root
    mode: 755
hooks:
  ApplicationStop:
    - location: scripts/deregister_from_elb.sh
  BeforeInstall:
    - location: scripts/install_dependencies.sh
  ApplicationStart:
    - location: scripts/start_httpd.sh
  ValidateService:
    - location: scripts/test_site.sh
    - location: scripts/register_with_elb.sh
```

- 将应用文件和配置文件份送到相应的文件夹
- 设置文件夹授权属性
- 增加/减少虚机到负载均衡
- 安装相关软件包
- 启动Apache
- 确认部署成功
- 更多！

灵活选择部署方式

一次一个



一次一半



全部一起



Dev Deployment group



OR

Prod Deployment group



AWS支持DevOps的开发管理和应用部署服务

**NEW!**

AWS OpsWorks For Chef Automate

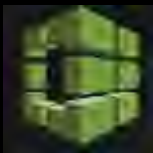
全托管的Chef服务器

**NEW!**

BLOX

开源的容器
管理协同工具

开发工具



AWS

CodeCommit

AWS

CodeBuild

AWS

CodePipeline

AWS

CodeDeploy

管理工具



AWS

CloudFormation

AWS

OpsWorks

计算



AWS Elastic

Beanstalk

Amazon ECS

亚马逊的DevOps的最佳实践

- CI/CD 是必须的!
 - 随时Commit，每次commit后Build，
每次执行build一次，部署到运行环境中继续测试
- 所有代码(应用，架构，文件) 都要进仓储repository
 - 不进Repository就不能进生产环境
- 从持续集成CI入手，一旦测试证明了已掌握DevOps，开始持续部署
- 部署到原型机，测试；部署到AZ可用区，测试；部署到Region大区，测试
- 代码审议是改进代码最好的办法
 - 是否简洁，易懂，达到目的
- 检查风格，让其他人能使用修改
- 自动混滚是部署失败后退回的最佳方式
- 使用仪表盘掌控开发部署全流程



使用 CloudFormation，
IAM 和 Lambda

云计算新趋势

- 云计算从服务器，到容器，到函数深入发展
- 云计算帮助DevOps成为最佳运维实践
- 数据中心运维人员在云时代仍是娇宠

打造架构合理的原生云应用程序的方法

使用最佳应用架构框架

基于核心云功能的五个支柱而构建

通盘考虑：设计、实施、部署和运维

不光关乎目的地，更关乎路上每一步如何走

您在路径中的位置决定了您要使用的服务



Thank you !