

# AWS Summit

AWS 技术峰会 · 北京 2014

AWS产品最新动态

庄富任

2014年12月12日





云已成为新常态

# AWS 快速创新的步伐

从一开始，AWS已经

- 发布**1111**种新服务和功能
- 引进**40+**个主要的新服务
- **47**次宣布降价

**+24**

Amazon EBS  
Amazon CloudFront

2008

**+48**

Elastic Load Balancing  
Auto Scaling  
Amazon VPC  
Amazon RDS

2009

**+61**

Amazon SNS  
AWS Identity & Access Management  
Amazon Route 53

2010

**+82**

Amazon SES  
AWS Elastic Beanstalk  
AWS CloudFormation  
Amazon ElastiCache  
AWS Direct Connect  
GovCloud

2011

**+159**

AWS Storage Gateway  
Amazon Dynamo DB  
Amazon CloudSearch  
Amazon SWF  
Amazon Glacier  
Amazon Redshift  
AWS Data Pipeline

2012

**+280**

Amazon Elastic Transcoder  
AWS OpsWorks  
Amazon CloudHSM  
Amazon AppStream  
Amazon CloudTrail  
Amazon WorkSpaces  
Amazon Kinesis

2013

**+454**

Amazon EC2 Container Service  
AWS Lambda  
AWS Service Catalog  
AWS Config  
AWS CodeDeploy  
AWS CodeCommit  
AWS CodePipeline  
AWS Key Management Service  
Amazon RDS for Aurora  
Amazon Cognito  
Amazon Mobile Analytics  
Amazon Zocalo  
AWS Directory Service  
2014

\*as of Nov 13, 2014

# 不只是广泛的服务..... 更有深层次的特性



## 计算

通用 (M3)  
计算优化 (C3)  
内存优化 (R3)  
图形处理器优化 (G2)  
存储优化 (HS1)  
IO优化 (I2)  
低成本、超频性能 (T2)



## 存储

Glacier  
EBS  
磁性磁盘  
通用SSD  
PIOPS SSD



## 关系型数据库

MySQL远程数据服务  
SQL服务器远程数据服务  
Oracle远程数据服务  
PostgreSQL远程数据服务



## 数据仓库

客户提供的密钥  
快照共享  
PCI和FedRAMP合规  
跨区域备份

# AWS re:Invent

OLD WORLD

旧世界

# 旧世界的关系型数据库

专利的

昂贵的

灵活性有限

锁定

这就是为什么客户转向使用更开放、友好度更高、更便宜的数据库引擎





.....但是，得到可与专有数据库媲美的性能较为困难

# 新产品介绍

## Amazon Aurora

开源成本的商用级数据库引擎



## MySQL兼容

成本仅有当前领先  
商务数据库解决方案的十分之一

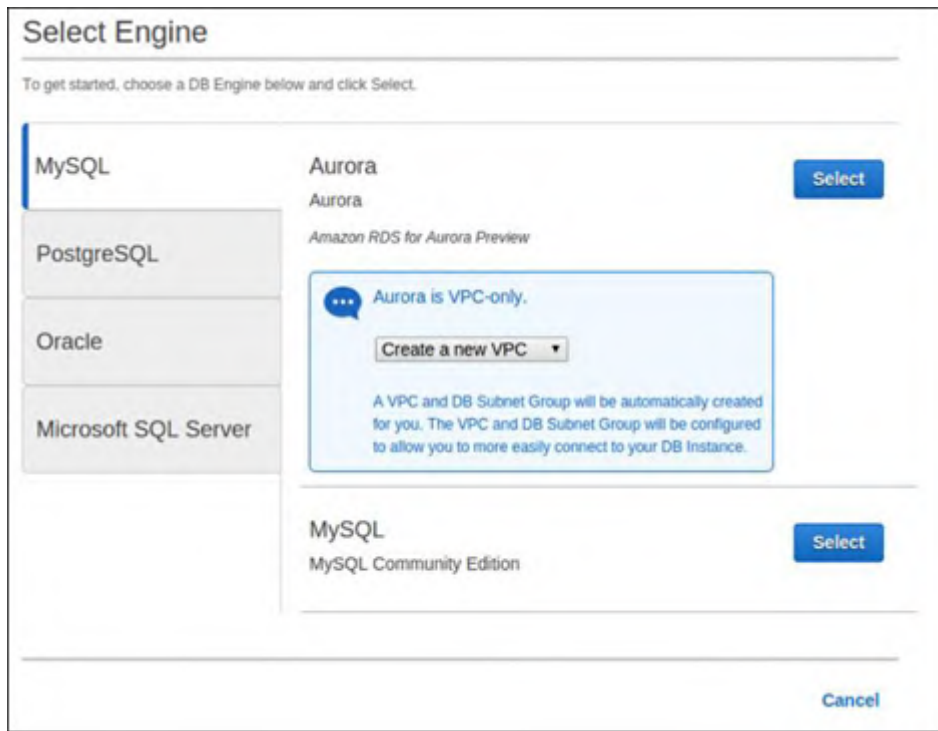
# 性能和可扩展性

- 更快
  - 优于标准MySQL 5倍的性能
- 计算资源扩展性
  - 向下或向上扩展
  - 最高 32 vCPUs 内核 和 244 GiB 内存
  - 数分钟完成
- 存储资源自动扩展
  - 存储根据需求自动扩展 (从 10GB 到 64TB)
  - 不需担心数据增长和管理存储容量
- Amazon Aurora 数据库副本
  - 最高 15 数据库副本
  - 非常低的复制延迟时间 – 数毫秒级别

# 可靠性

- 容错和自我修复的存储
  - 复制6份数据在3个可用区
  - 连续扫描数据块和磁盘错误并自动更换
  - 主数据库故障, 分钟内自动转移到辅助数据库
- 持久性备份
  - 数据备份到Amazon S3
  - 99.999999999% 持久性
  - 对数据库的性能没有影响

# 使用 AWS RDS for Aurora



- 选择MySQL数据库，再选择Aurora引擎

# 使用 AWS RDS for Aurora

Specify DB Details

---

Instance Specifications

DB Engine Aurora - compatible with MySQL 5.6.10

DB Instance Class db.r3.large – 2 vCPU, 15 GiB RAM

Settings

DB Instance Identifier\* mydata

Master Username\* dbadmin

Master Password\* ●●●●●●

Confirm Password\* ●●●●●●

Retype the value you specified for Master Password.

\* Required

Cancel Previous Next Step

- 选择机型，命名数据库并创建一个DBA账户

# 使用 AWS RDS for Aurora

Configure Advanced Settings

Network & Security

VPC\* [VPC Selection is Required]

Subnet Group dbsub

Publicly Accessible No

Availability Zone No Preference

VPC Security Group(s) **default (VPC)**  
d-906737db3a\_controllers (VPC)  
d-906737db3a\_workspacesMembers  
default (VPC)

Database Options

Database Name

Database Port 3306

Parameter Group default:aurora5.6

Option Group default:aurora-5-6

Backup

Backup Retention Period 7 days

Backup Window No Preference

Maintenance

Auto Minor Version Upgrade Yes

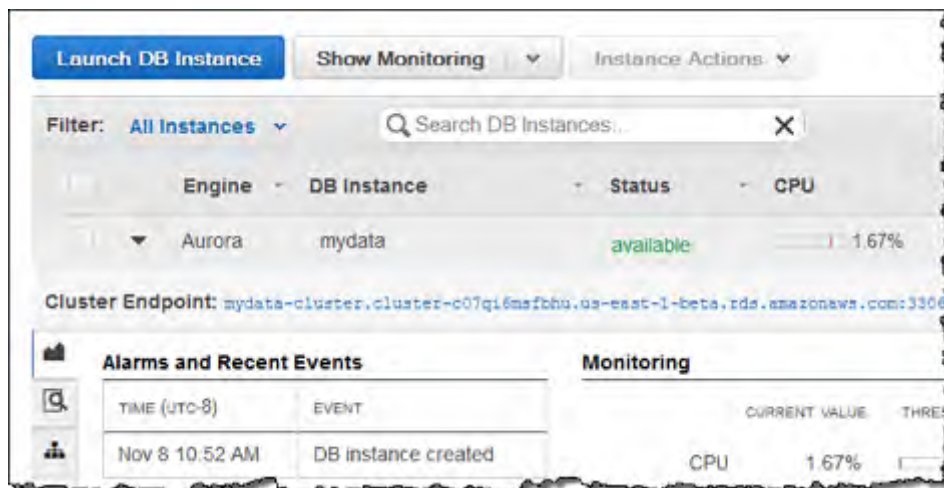
Maintenance Window No Preference

\* Required

Cancel Previous **Launch DB Instance**

- 设定网络与安全，备份及维护选项，以及数据库的特定选项
- 您不需要设定数据库容量，在您使用数据库时，会自动分配**10GB**的存储增量

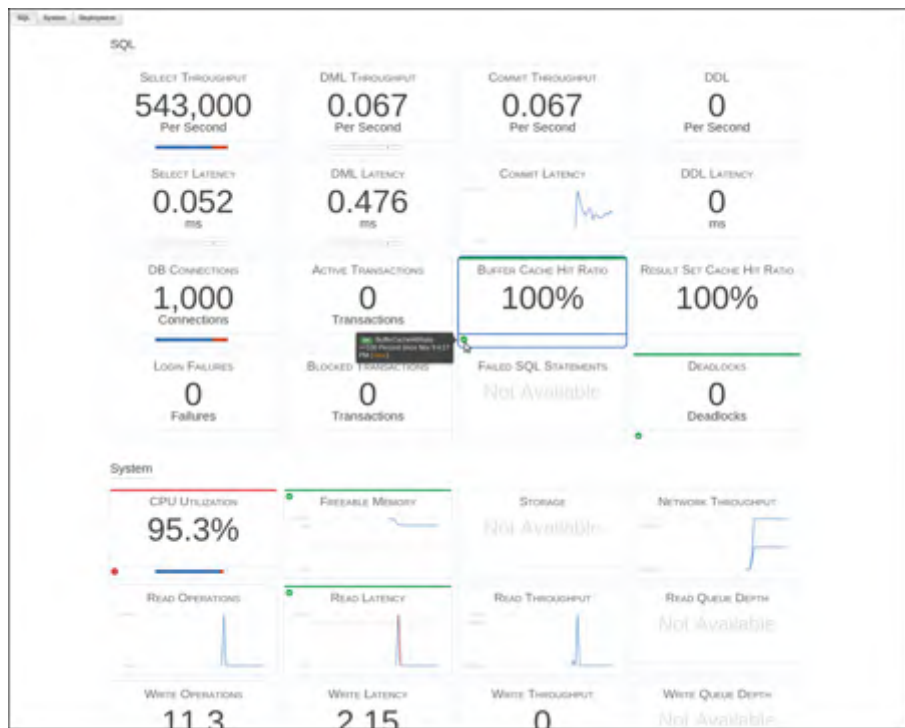
# 使用 AWS RDS for Aurora



- 大约十分钟后，数据库生成



# 使用 AWS RDS for Aurora



- 一旦运行，所有指标都会在监测界面显示

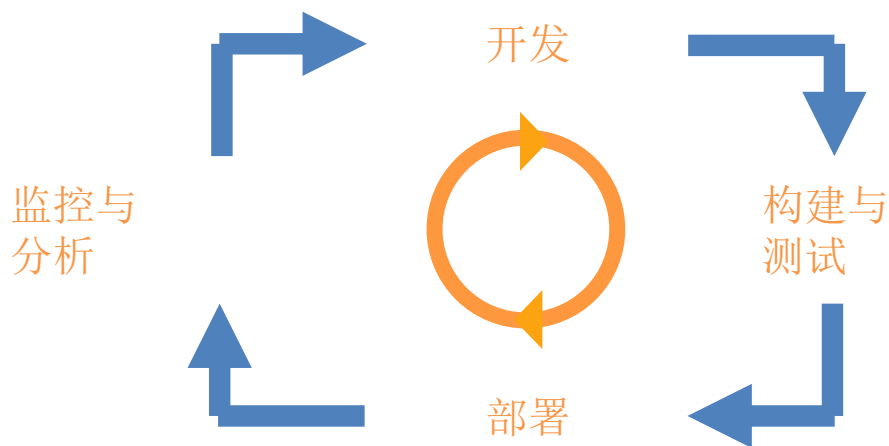
# 后续步骤

- 在美国东部（弗吉尼亚州北部）区域的预览形式中可用
- 更多详情，请登录我们的网站：  
<http://aws.amazon.com/rds/aurora/>

# 新产品介绍

## AWS CodeDeploy

一个全面管理的大规模代码部署服务



# 自动部署



将代码推送至**EC2**的任一运行环境  
(test, staging and production)

自动、可重复、全面管理

降低由部署失败或应用程序错误导致的故障风险

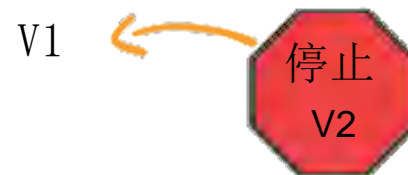
# 最小化停机时间



滚动更新



健康度跟踪



停止及回滚

# 集中控制

集中监控与控制



部署历史记录



# 容易使用

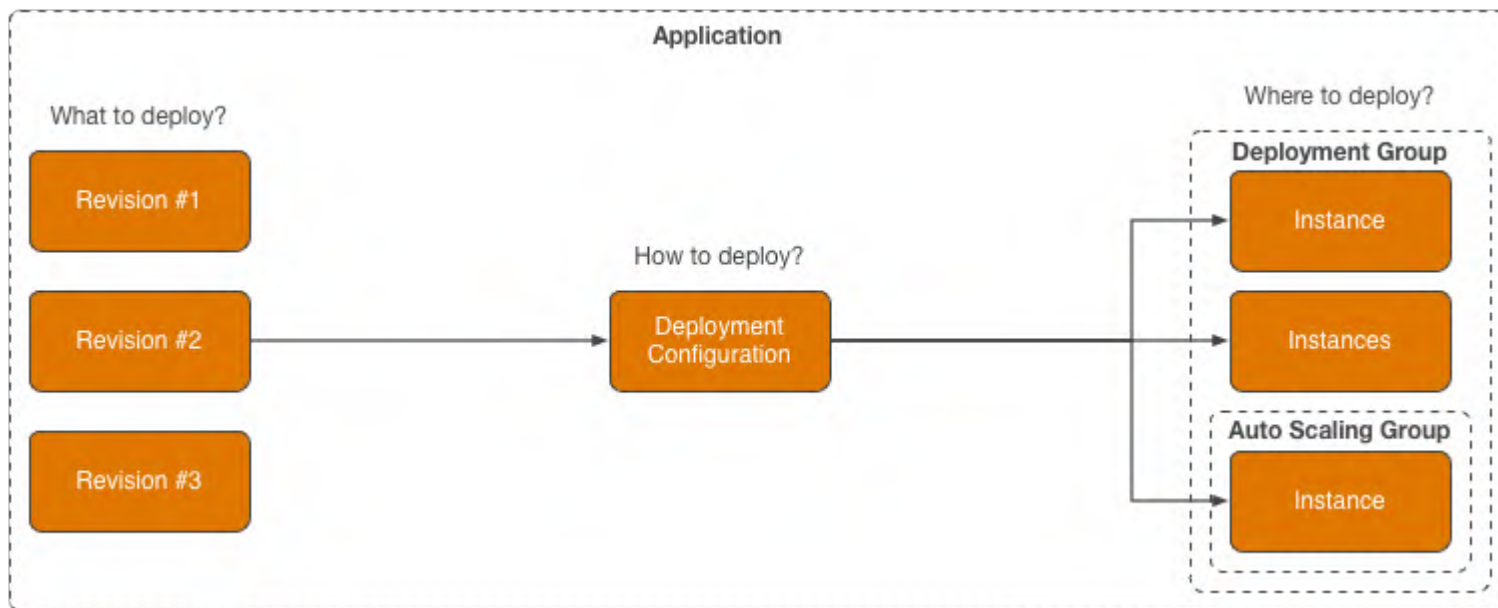
## 语言及架构



## 工具链整合



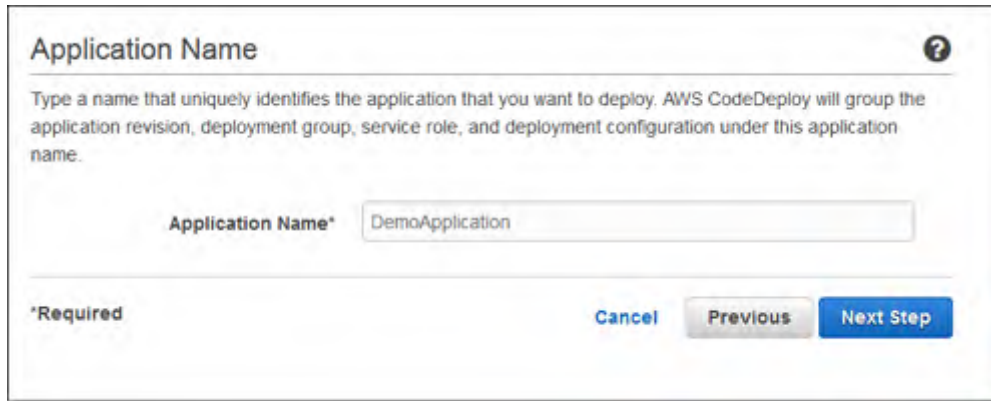
# 术语



**Revision** 修订版本  
Amazon S3桶或  
GitHub库



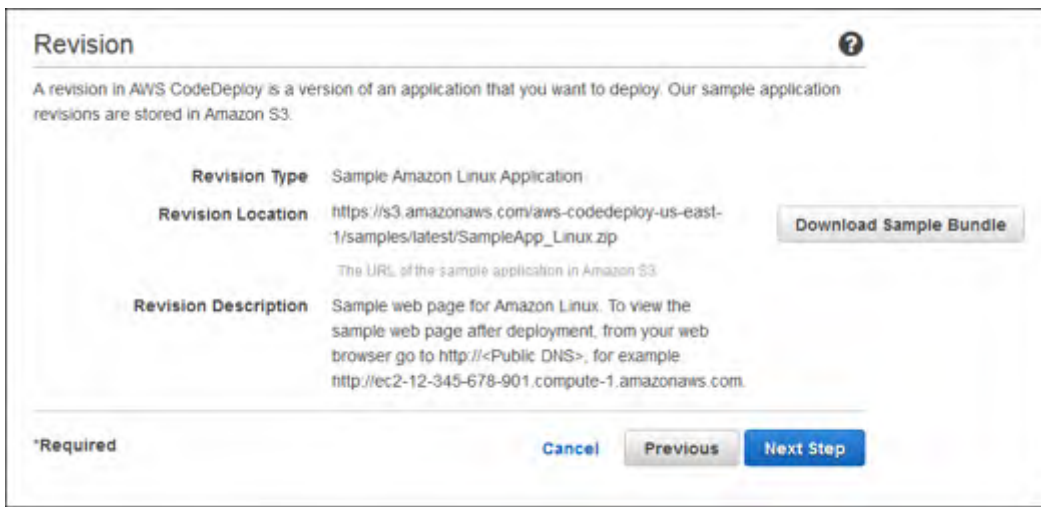
# 使用AWS CodeDeploy



The screenshot shows the 'Application Name' step in the AWS CodeDeploy console. The title 'Application Name' is at the top left, with a help icon (question mark) at the top right. Below the title is a descriptive text: 'Type a name that uniquely identifies the application that you want to deploy. AWS CodeDeploy will group the application revision, deployment group, service role, and deployment configuration under this application name.' A text input field labeled 'Application Name\*' contains the text 'DemoApplication'. At the bottom left, there is a '\*Required' label. At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Next Step'.

- 从定义应用程序开始

# 使用AWS CodeDeploy



- 创建一个修订版本进行部署

在这个例子中，修订版本存储在S3中，但它也可来自于CodeCommit或GitHub

# 使用AWS CodeDeploy

### Service Role ?

Select an existing service role that allows AWS CodeDeploy to work with other dependent AWS services on your behalf during a deployment. If you're not sure if you already have a service role with the correct permissions, in the Service Role drop-down list select Create A New Service Role, and we will create one for you.

**Service Role\*** Use an existing service role

**Role Name\*** CodeDeploySampleStack-9eatwh-Co

\*Required

[Cancel](#) [Previous](#) [Next Step](#)

- 定义IAM角色，用于与其他AWS服务如EC2 或Auto Scaling进行交互

# 使用AWS CodeDeploy

**Deployment Configuration**

Choose from a list of default deployment configurations, or create a custom configuration.

Default Deployment Configurations

Create Custom Deployment Configuration

---

**One at a Time**

**The deployment will:**  
Deploy to one instance at a time. Succeed if all instances succeed. Fail after the very first failure. Allow the deployment to succeed for some instances, even if the overall deployment fails.

**Example:**  
If you deploy your application to 3 instances, this configuration will deploy to one instance at a time.

- ✔ Succeeds if all 3 instances succeed.
- ✖ Fails after any instance fails.

Select

---

**Half at a Time**

**The deployment will:**  
Deploy to up to half of the instances at a time, with fractions rounded down. Succeed if at least half of the instances succeed, otherwise it will fail. The deployment may succeed for some instances, even if the overall deployment fails.

**Example:**  
If you deploy your application to 3 instances, this configuration will deploy to one instance at a time.

- ✔ Succeeds if 2 or more instances succeed.
- ✖ Fails if 2 or more instances fail.

Select

---

**All at Once**

**The deployment will:**  
Deploy to all instances at once. Succeed if at least one instance succeeds. Fail after all instances fail.

**Example:**  
If you deploy your application to 3 instances, this configuration will deploy to all 3 instances at once.

- ✔ Succeeds if any instance succeeds.
- ✖ Fails if all instances fail.

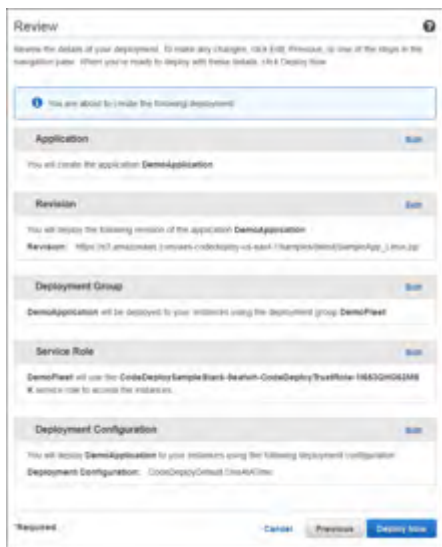
Select

\*Required

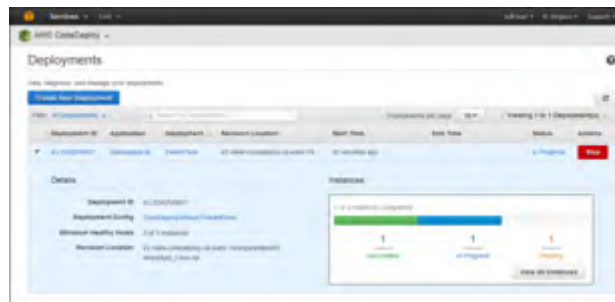
Cancel Previous **Next Step**

- 创建新的配置或者从默认中选择

# 使用AWS CodeDeploy



- 检查设置与配置



- 部署过程会在AWS Management Console上显示。

# 与您现有的开发工具集成



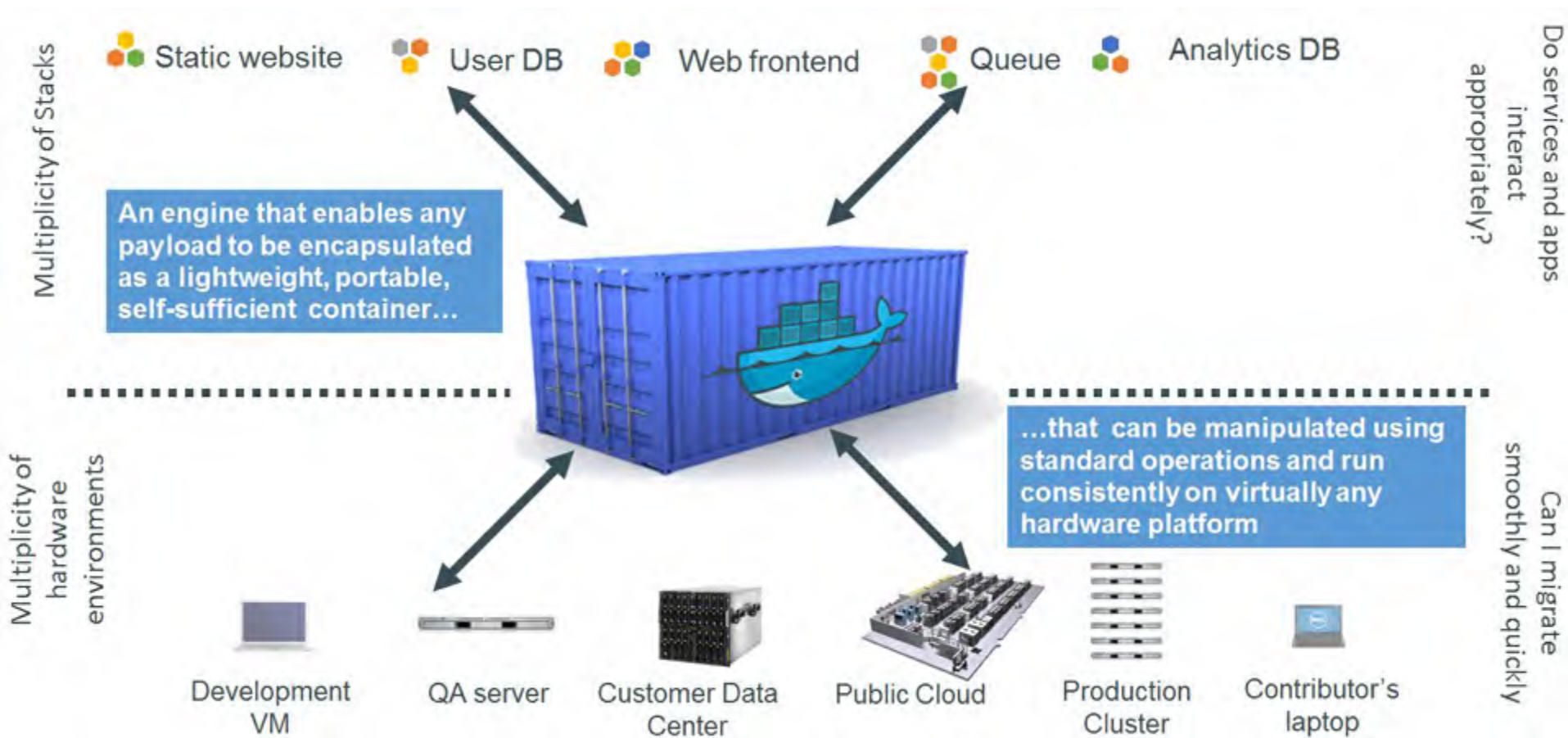
# 后续步骤

- 现在AWS CodeDeploy在美国东部（弗吉尼亚州北部）以及美国西部（俄勒冈州）区域可用。
- AWS CodeDeploy没有额外费用。您只需为您创建用于存储及运行应用程序的AWS资源付费（如EC2或S3 储存）
- 更多详情，请登录我们的网站：  
<https://aws.amazon.com/codedeploy/>

# 配置、管理、扩展 **Docker** 容器最简单的方法



# 构建一次, 随处可用



# 亚马逊EC2容器服务 (ECS)

- 支持 Docker 容器 轻松地在 Amazon EC2 实例的托管集群上运行分布式应用程序
- 能够使用简单的 API 调用启动和停止容器启用的应用程序
- 自动调度容器，在您的资源需求和可用性要求之间寻找一个平衡
- 使用EC2容器服务，您可以简单地开始启用和管理上万事例的集群，可以在多个可用性区域调度上千个容器
- 亚马逊EC2容器服务在您自己的EC2事例中启动您的容器，所以您不会与其他客户共享资源，它将您的集群放到一个VPC里，并允许您使用您自己的VPC安全群组和网络ACLs。
- 您的应用程序可以使用的AWS特性包括：弹性IP地址、资源标签以及虚拟子网（VPC）

# 后续步骤

- 预览形式
- **ECS**无额外费用。跟往常一样，您只需为您使用的资源付费。
- 更多详情，请登录我们的网站：  
<http://aws.amazon.com/ecs/>

# 构建高度扩展的响应式应用程序的新方法

# AWS Lambda

- 依响应事件运行您的代码并自动为您管理计算资源。零管理
- Lambda函数设计: Node.js
- 无状态性 (Stateless) — 如果需要, 用S3或DynamoDB来存储状态
- 以下变化会触发Lambda函数:
  - 亚马逊S3 储存
  - 亚马逊DynamoDB 表格
  - Kinesis数据流

```
1 console.log('Loading event');
2 var aws = require('aws-sdk');
3 var s3 = new aws.S3({apiVersion: '2006-03-01'});
4
5 exports.handler = function(event, context) {
6   console.log('Received event:');
7   console.log(event);
8   // Get the object from the event and show its
9   s3.getObject({Bucket:event.Bucket, Key:event.
10-   function(err,data) {
11-     if (err) {
12-       console.log('error getting object');
```

# 减轻管理底层架构的痛

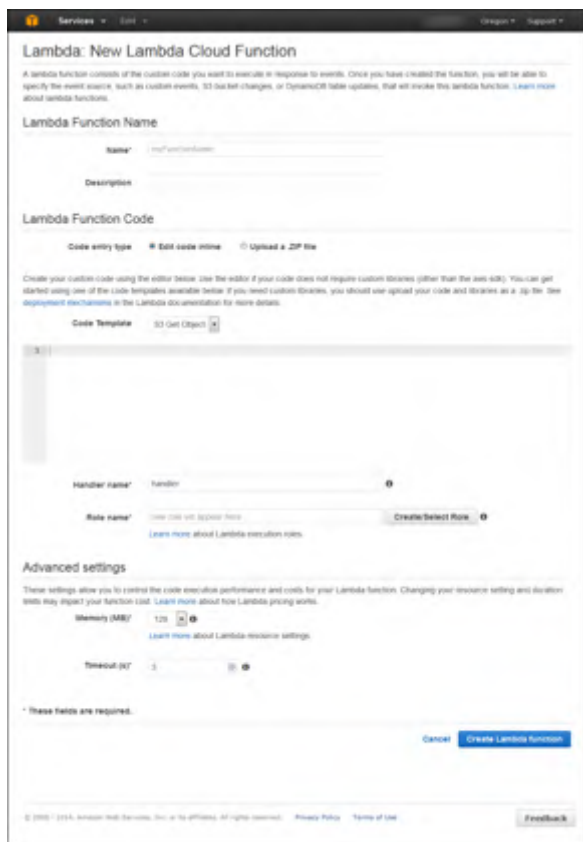
- 好处

- 快速响应新事件
- 经济高效且轻松地从每天几个请求自动扩展到每秒数千个请求
- 降低客户端的负载
- 容错 (99.99% 可靠度)

- 使用案例

- 事件触发
  - 当上传图像到 Amazon S3 自动创建一个缩图或执行面部识别
  - 当上传文件到S3 加上浮水印或进行压缩
  - 当地址更新在 Amazon DynamoDB 表进行验证和触发警报
- 流处理
  - 处理点击流数据 [Amazon Kinesis](#)
- 建立自定义的后端服务
  - 将客户端逻辑移到 Lambda 函数

# 使用AWS Lambda



- 命名并描述新的Lambda函数
- 键入代码或者上传一个zip文件
- 告知Lambda运行哪个函数以及使用哪个IAM角色来执行这个函数
- 设置内存条件并指定执行时间

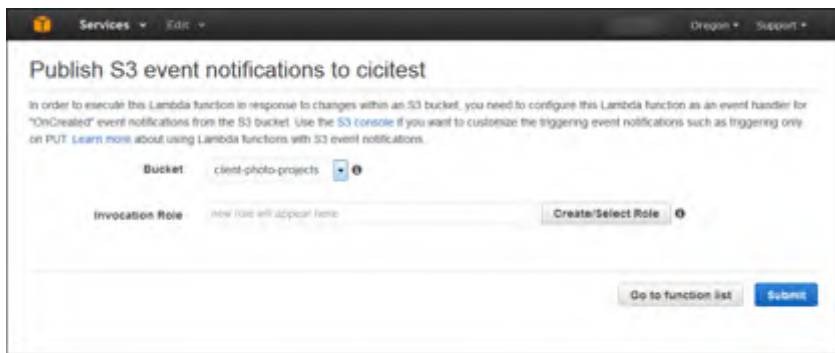
# 使用AWS Lambda



- AWS管理控制台提供了反复编辑和测试功能

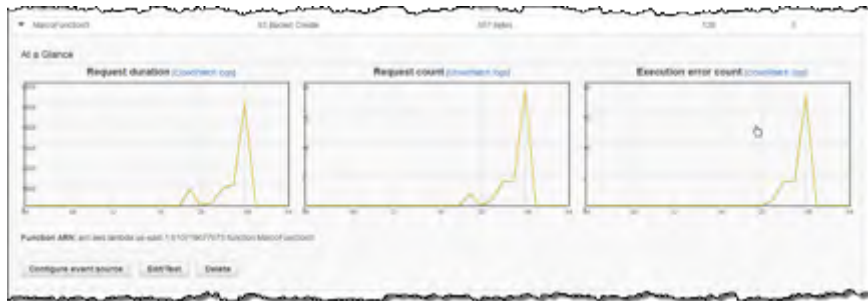


# 使用AWS Lambda



- 一旦测试完成,将Lambda函数附加到事件源中, 如亚马逊S3事件通知
- 指定一个调用角色来为S3提供调用这项功能所需的权限

# 使用AWS Lambda



- Lambda指标被发送到云监控 (CloudWatch) 并在AWS管理控制台显示

# 后续步骤

- 预览形式可用在美国东部(弗吉尼亚部),美国西部(俄勒冈州)和欧洲(爱尔兰)地区。
- 每个请求计费,以100毫秒为单位的计算时间付费。
- 更多详情,请登录我们的网站:  
<http://aws.amazon.com/lambda/>

持续创新核心服务。



# 亚马逊EC2 C4事例



下一代亚马逊EC2计算型优化实例

- 基于Intel Xeon E5 — 2666 v3 (Haswell)处理器
- 2.9 GHz—峰值3.5 GHz，涡轮加速



是运行应用程序、游戏和web服务器，转码服务，高性能计算工作负载的理想选择。

| 实例名称       | vCPU Count | RAM      | 网络性能    |
|------------|------------|----------|---------|
| c4.large   | 2          | 3.75 GiB | 中       |
| c4.xlarge  | 4          | 7.5 GiB  | 中       |
| c4.2xlarge | 8          | 15 GiB   | 高       |
| c4.4xlarge | 16         | 30 GiB   | 高       |
| c4.8xlarge | 36         | 60 GiB   | 10 Gbps |

规格暂定

# 亚马逊EBS

 增加通用(SSD)和预分配IOPS(SSD)大小的性能和容量。

| 实例名称                | 容量              | IOPS                         | 吞吐量        |
|---------------------|-----------------|------------------------------|------------|
| 亚马逊EBS通用 (SSD)      | 16 TB<br>(1TB起) | 10000 IOPS<br>(3000 IOPS起)   | 160 MBps * |
| 亚马逊EBS预分配IOPS (SSD) | 16 TB<br>(1TB起) | 20000<br>IOPS(4000<br>IOPS起) | 320 MBps * |

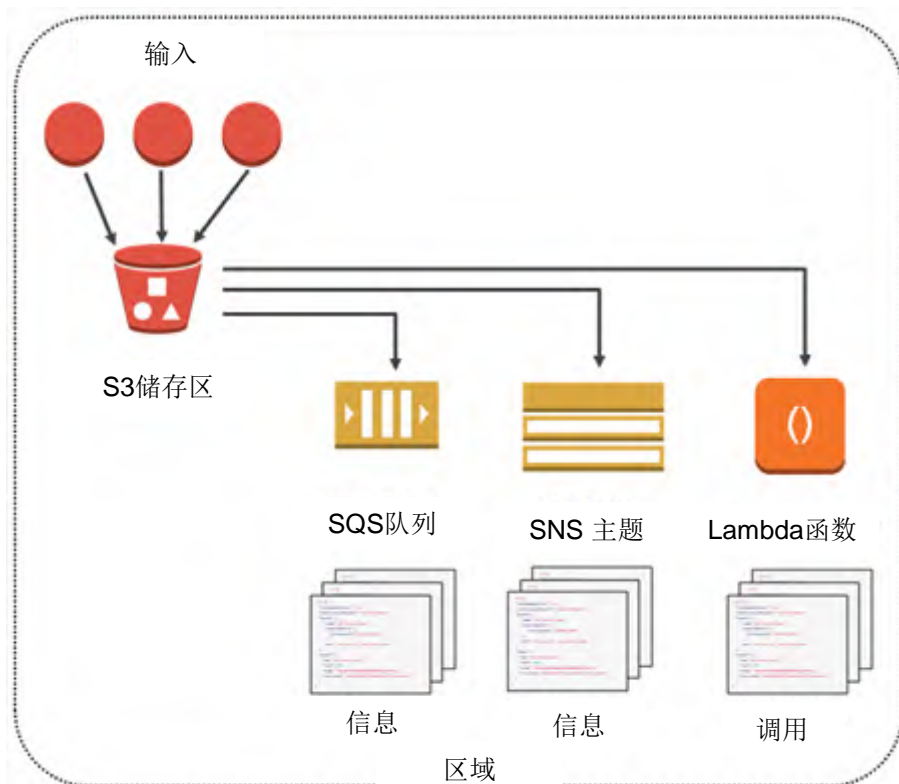
\*当用于EBS优化实例时



# 亚马逊S3事件通知

- 📦 当添加新的对象到Amazon S3 储存或一个现有的对象被重写，会发通知到亚马逊简单队列服务(SQS)或Amazon简单通知服务(SNS)
- 📦 通知也会送到AWS Lambda用Lambda函数进行处理。
- 📦 通知可触发工作流程、警告或者进行其他进程。

# 亚马逊S3事件通知工作流程



- 通知会被发送到亚马逊SQS、亚马逊SNS或者会被用于触发一个亚马逊Lambda函数



# 主要特性

- 事件通知在1秒以内送到目的地。
- 可靠性— 旨在执行高度可靠地配送通知。它包括内置的回退并重试机制，用来处理瞬间的问题
- 更多的事件类型 — 随着时间添加更多的事件类型

# 谢谢！

马上开启您的云旅程

立即线上申请AWS中国云服务

<https://www.amazonaws.cn/sign-up/>

中文网站: [www.amazonaws.cn](http://www.amazonaws.cn)

新浪微博: @亚马逊AWS

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

微信  
AWS 中国

