



# Architecture Based Cloud

## ——云计算时代的ABC



- 云计算和开发
- 产品需求和云服务
- 基于云架构的成功实例
- 云架构的优势和使用技巧

# 云计算技术



## 云计算的优势：

- 节省成本
- 提高效能
- 海量扩展
- 安全稳定

## 商业模式的改变：

- 云计算将自给自足的“IT自然经济”转换为“IT商品经济”

## ■ IaaS

- 层次最低，提供基础资源，基本不改变开发习惯，架构设计自由度高

## ■ PaaS

- 层次居中，提供开发平台，改变开发习惯，架构设计自由度中

## ■ SaaS

- 层次最高，提供应用软件，改变开发习惯，架构设计自由度低



## 盛大云帮您

降低成本  
提高效率  
更加安全可靠  
[>> 了解更多](#)



利用虚拟化技术，将最大的计算资源化为资源池，提供云主机和应用服务。

[了解云主机](#)

为云主机提供的数据高可靠独立可扩展的硬盘。

[了解云硬盘](#)

消除不同网络运营商之间互通瓶颈，降低服务器带宽压力，加速资源访问速度。

[了解云分发](#)

弹性、安全、海量的Key-value存储服务。

[了解云存储](#)

云监控能7x24小时监控网站故障，实时故障报警，分析网站访问速度。

[了解云监控](#)

MongoIC：易用、安全、便捷的MongoDB云服务。

[了解MongoIC](#)

网站、论坛、百科、黄页、云主机，一键安装，瞬间开始您的互联网成功之路。

[COMING SOON](#)

提供各种关系型数据库以及各种NoSQL数据库的云服务。

[COMING SOON](#)[新闻公告](#)[媒体报道](#)[近期活动](#)[...](#)

- 盛大云全新官网上线 2011-12-06
- 充值200元（及以上）立即激活账户 2011-11-23
- 11月19日（周六）部分云主机受影响以及补偿方案通告 2011-11-21
- 关于禁售IP资源公告 2011-11-17
- 安全组件化使用上线公告！ 2011-11-16
- 盛大云-RHEL5.4资源上线公告 2011-11-15

[经典案例](#)[+ MORE](#)

- 云计算和开发
- 产品需求和云服务
- 基于云架构的成功实例
- 云架构的优势和使用技巧

## ■ 硬件（基础）

- 服务器：CPU, MEM, DISK, I/O

- 网络：带宽, IP, 线路

- 存储：盘柜等

## ■ 软件（提供服务）

- Lamp, Inmp, radis, squid等等

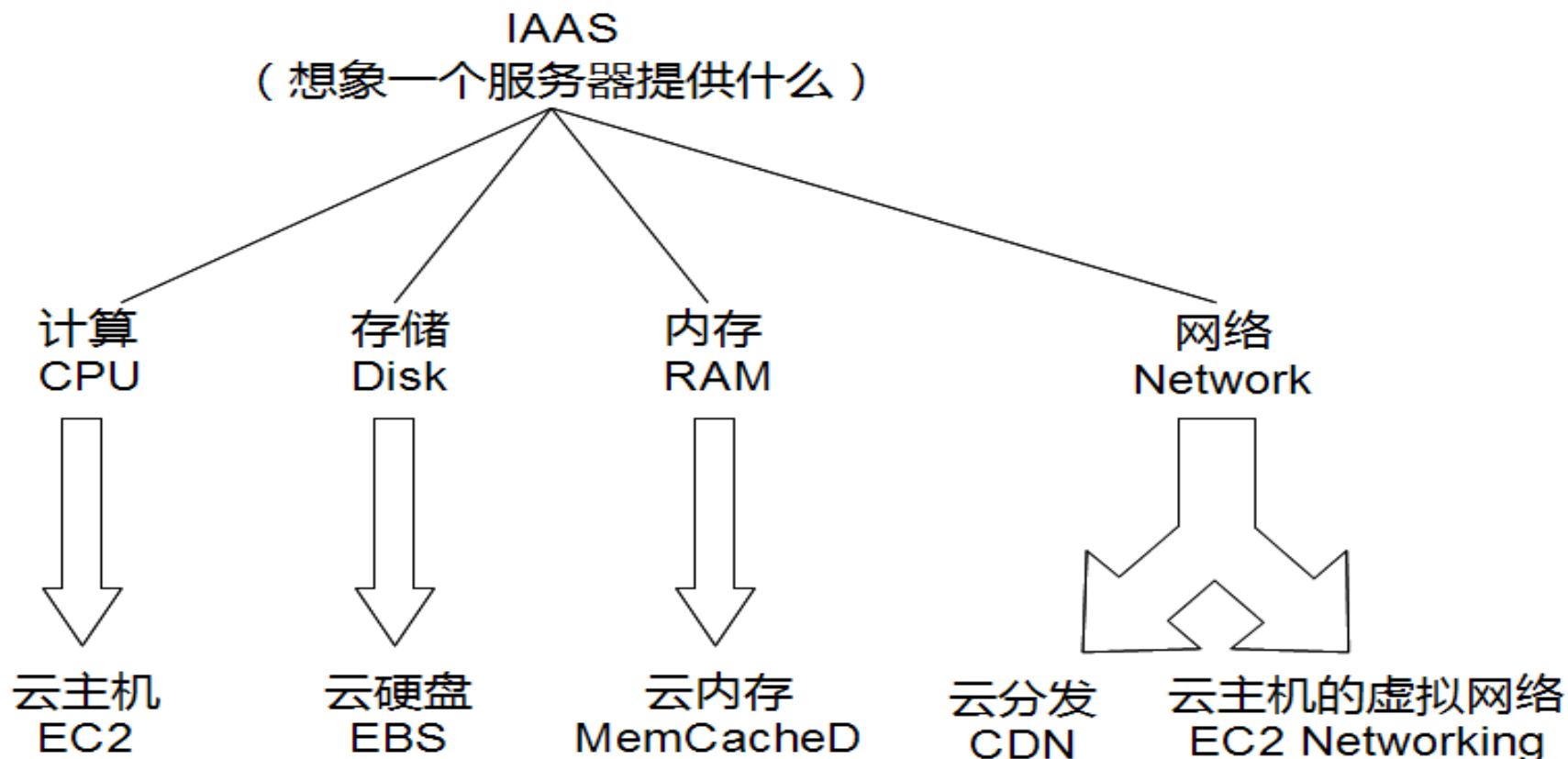
## ■ 质量（更好的提供服务）

- 负载均衡
- 良好覆盖面的网络
- 更快的访问速度（网络延迟，缓存，db读写，业务逻辑等）
- 灾难恢复，多点备份
- 可扩展的架构

## ■ 安全（可靠的提供服务）

- 防攻击
- 防漏洞

# 云的硬件(服务的基础)



# 云主机本地硬盘，云硬盘，云存储对比

- 云主机本地硬盘（类比：普通硬盘）
  - 空间固定，不具备弹性扩展能力
  - 裸设备，不可动态插拔
  - 保存一份数据，若云主机故障则不可用
- 云硬盘（类比：一块高可靠的移动硬盘）
  - 空间可按需申请，具备弹性扩展能力
  - 裸设备，支持动态插拔
  - 保存多份数据，数据持久保存，云主机故障不影响数据
- 云存储（类比：一个高可靠的，容量不限的文件夹）
  - 通过简单的API存取文件和数据
  - 保存多份数据，数据持久保存，硬件故障不影响数据安全

## ■ Lamp,Lnmp等

- 网站云 (ec2+phpcms+discuz! 等)

- 数据库云

- 移动云 (直接提供移动类服务)

- 等等

# 云提供的服务质量（更好的提供服务）

- 负载均衡
  - ELB（弹性负载均衡）
- 数据备份
  - 云硬盘（EBS），云存储（S3）
  - 云主机镜像（image）
- 网络访问质量
  - 云分发服务
- 弹性扩展
  - Autoscaling
- 监控
  - 云监控，cloudwatch

- 平台入口的入侵检测

- 防arp欺骗技术

- 广播风暴控制

- 虚拟以太网技术

- 云计算和开发
- 产品需求和云服务
- 基于云架构的成功实例
- 云架构的优势和使用技巧

## ■ Instagram

- 拍照分享应用Instagram
- 2年
- 0成本起家
- 13名员工（3名工程师）
- 10亿美元被Facebook收购



## ■ 经验

- Keep it very simple
- Don't re-invent the wheel
- Go with proven and solid technologies when you can

## ■ 负载均衡

- Amazon's Elastic Load Balancer

## ■ OS

- Amazon EC2 Ubuntu Linux 11.04 (Natty Narwhal)

## ■ 主机

- 100+ EC2 instances

- Nginx: 3个主机放在ELB后面

- 25+ Django application servers: High-CPU Extra-Large

- PostgreSQL: 12 Quadruple Extra-Large memory

- 6 memcached instances

- 硬盘存储
  - Amazon's EBS (用于快照, 备份, 组建软raid等)
- 数据存储
  - Amazon's S3 (大量照片)
- CDN
  - Amazon's CloudFront
- DNS
  - Amazon's Route53
- Cache
  - 6 memcached instances
  - Amazon Elastic Cache service

- 云计算和开发
- 产品需求和云服务
- 基于云架构的成功实例
- 云架构的优势和使用技巧

# 云主机

**超微主机**

CPU: 1  
内存: 512M  
硬盘: 8G  
**0.05元起 按需**  
**34.75元起 包月**  
**417元起 包年**



选择线路：

独享带宽：



**确认并递交申请**

远程登录用户名：

\* 远程登录密码：

申请主机的数量：

## ■ 弹性扩展

- 几分钟内完成创建或删除，提升或降低计算能力
- 同时创建数十个云主机

## ■ 按需使用，灵活多样

- 计费：按需、包月、包年
- 接入：Web界面；API；
- 配置：允许选择一组处理器个数、内存和磁盘大小不同的配置
- 系统：可以选择使用多种操作系统，并可搭配不同的应用程序
- Windows, CentOS, RHEL, Ubuntu等等

## ■ 镜像制作和主机克隆

- 支持把已经部署好各种软件环境的云主机做成镜像，保存起来
- 通过镜像可以快速生成大量环境相同的云主机，节省部署成本，提升部署速度

## ■ 弹性IP

- IP跟云主机分离；可动态绑定不同的IP到不同的云主机

# 云主机优势

## ■ 稳定、高质量的性能

- 计算能力划分：为云主机cpu提供稳定的计算能力，不受底层多样化物理cpu的影响
- 带宽控制：用户使用高质量独享带宽，彼此不受影响
- 虚拟以太网：每个用户形成独立的二层网络，彼此之间无影响；完全解决广播风暴问题

## ■ 高安全性

- 提供安全组功能，可自定义防火墙策略，通过网络层进行安全控制
- 绑定云主机IP，MAC对，禁止恶意网络包传播
- 云主机间资源的完全隔离

- 基本：全能型，直接替代各类服务器
- 进阶：根据应用的不同选择适宜的类型，部署不同的应用
  - 计算型：高CPU能力主机
  - 缓存型：大内存主机
  - DB型：高IO性能主机
  - 存储型：大磁盘主机

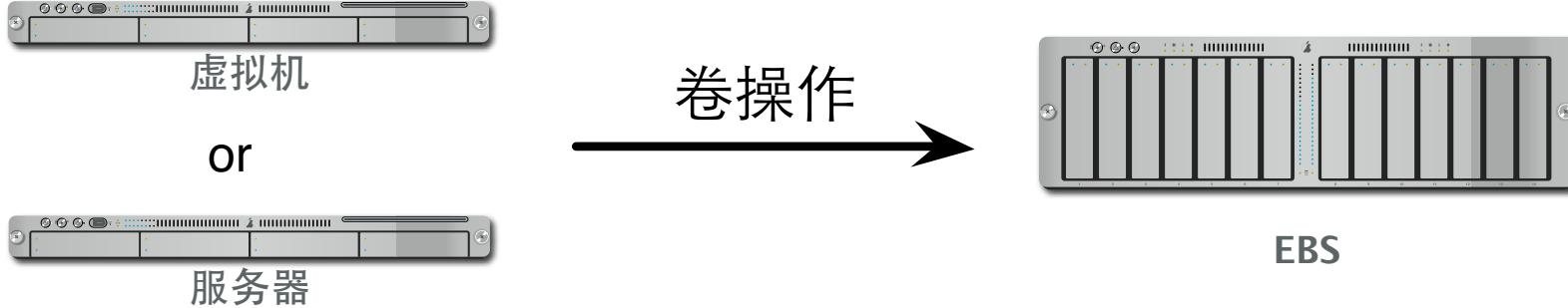
- 优化：全面使用已有的云产品，而不是使用“全能”的云主机
  - DB型：使用数据库云（mongoic, cds）
  - 存储型：云主机+云硬盘+云存储
  - 缓存：Amazon Elastic Cache service
  - 负载均衡：ELB

- 全盘的数据备份
- 使用步骤
  - 配置环境
  - 部署程序
  - 保存镜像
- 使用镜像生成各种云主机
- 勤于制作不同种类的镜像

# 云硬盘

## ■ 原理

- 基于TCP网络服务
- 定制的虚拟卷操作



## ■ 场景

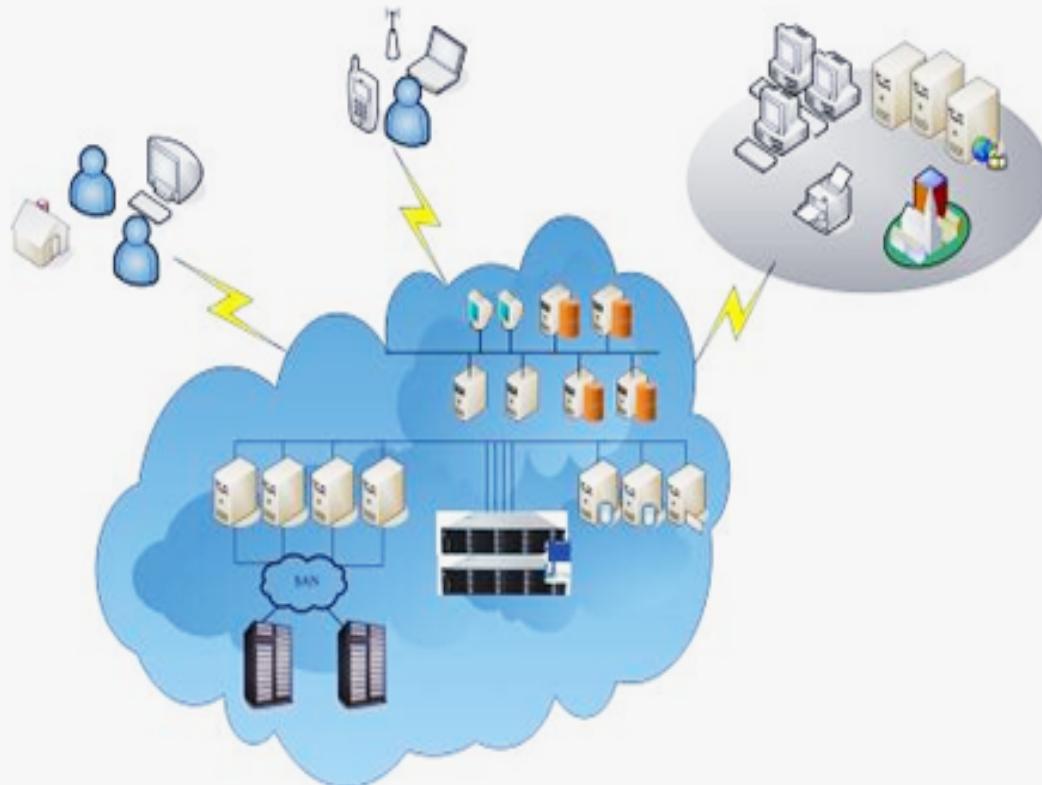
- 虚拟机的存储层
- 物理服务器也可以用

# 云硬盘的特点

- 内部数据主备两份保存，保证数据可靠性
- 弹性动态挂载，无需关机，即插即用
- 裸设备

- 数据备份
- 数据持久化
- 组软raid进一步提升性能和可靠性
- 数据迁移

- 提供基于Key/Value的海量数据存储服务，实现**高可用性、安全性、可靠性和实时性**。



- 各种媒体文件，图片，音乐，视频
- 用户文档，表格，数据文件
- 数据备份和存档，操作日志等
- 快速建站

- 云存储也能建站？？
- [www.hao123.com](http://www.hao123.com)
  - 2004年，百度以1190万人民币和4万股股票（现在约等于3.9亿人民币）收购
  - 技术上：静态页面

# 1分钟复制一个hao123

- 申请域名，比如hao1234.com
- 在云存储服务中创建名为hao1234.com的bucket
- 将该bucket的访问权限设为public
- 上传网站内容文件到该bucket下，比如index.html
- 生成云存储文件访问URL：  
hao1234.com.storage.grandcloud.cn
- 把自己的[www.hao1234.com](http://www.hao1234.com) 域名映射（CNAME）到  
hao1234.com.storage.grandcloud.cn上
- 可以使用了！！

共享流

- Google CloudSQL
- MS SQL Azure
- 盛大 MongolC

独享流

- Amazon RDS
- 盛大 CDS

## 申请

- 6种类型云主机
- 10G-1T的数据空间
- MySQL

## 安全

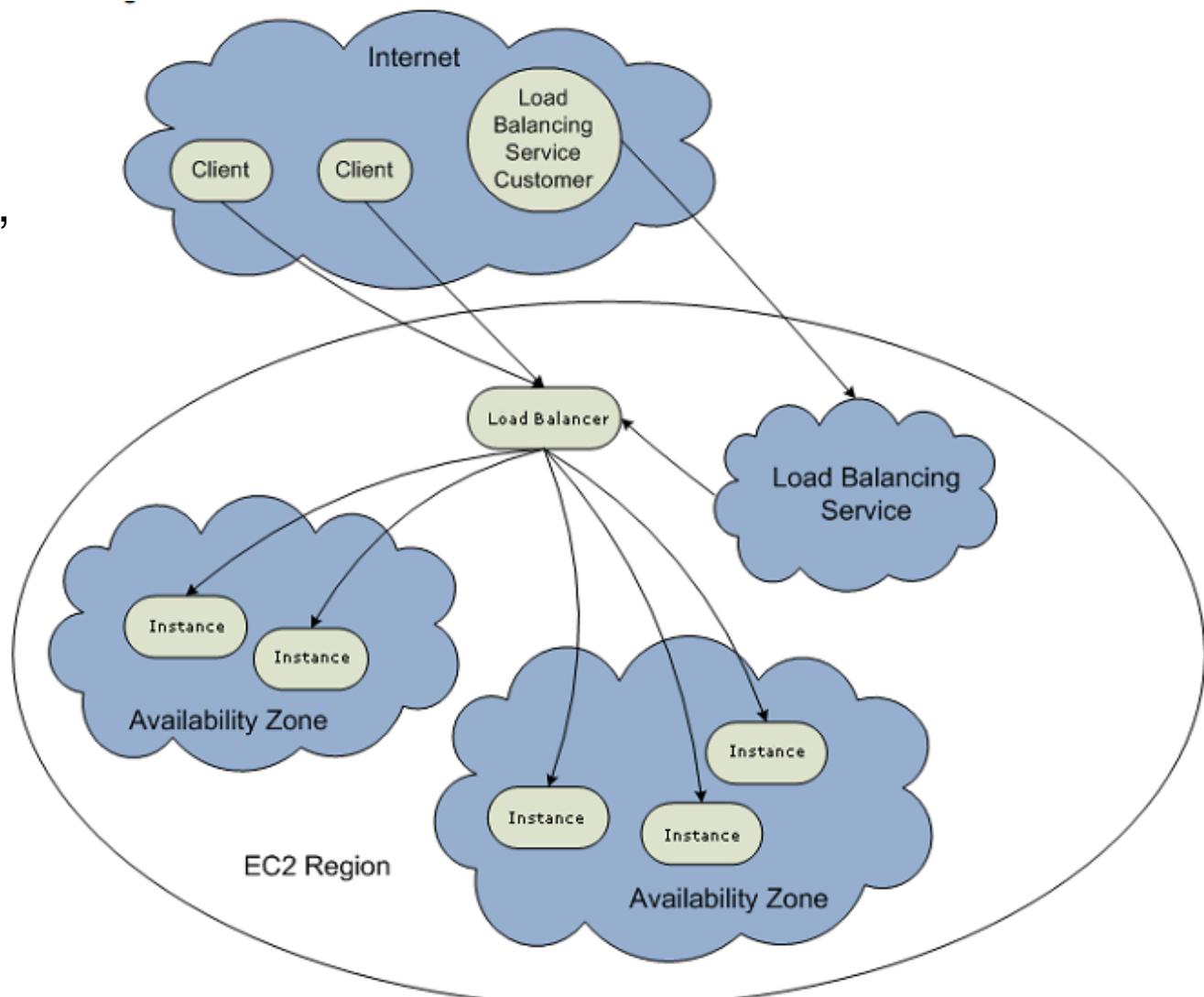
- 定时备份
- 立即备份
- 镜像查看
- 数据恢复
- 日志查询

## 管理

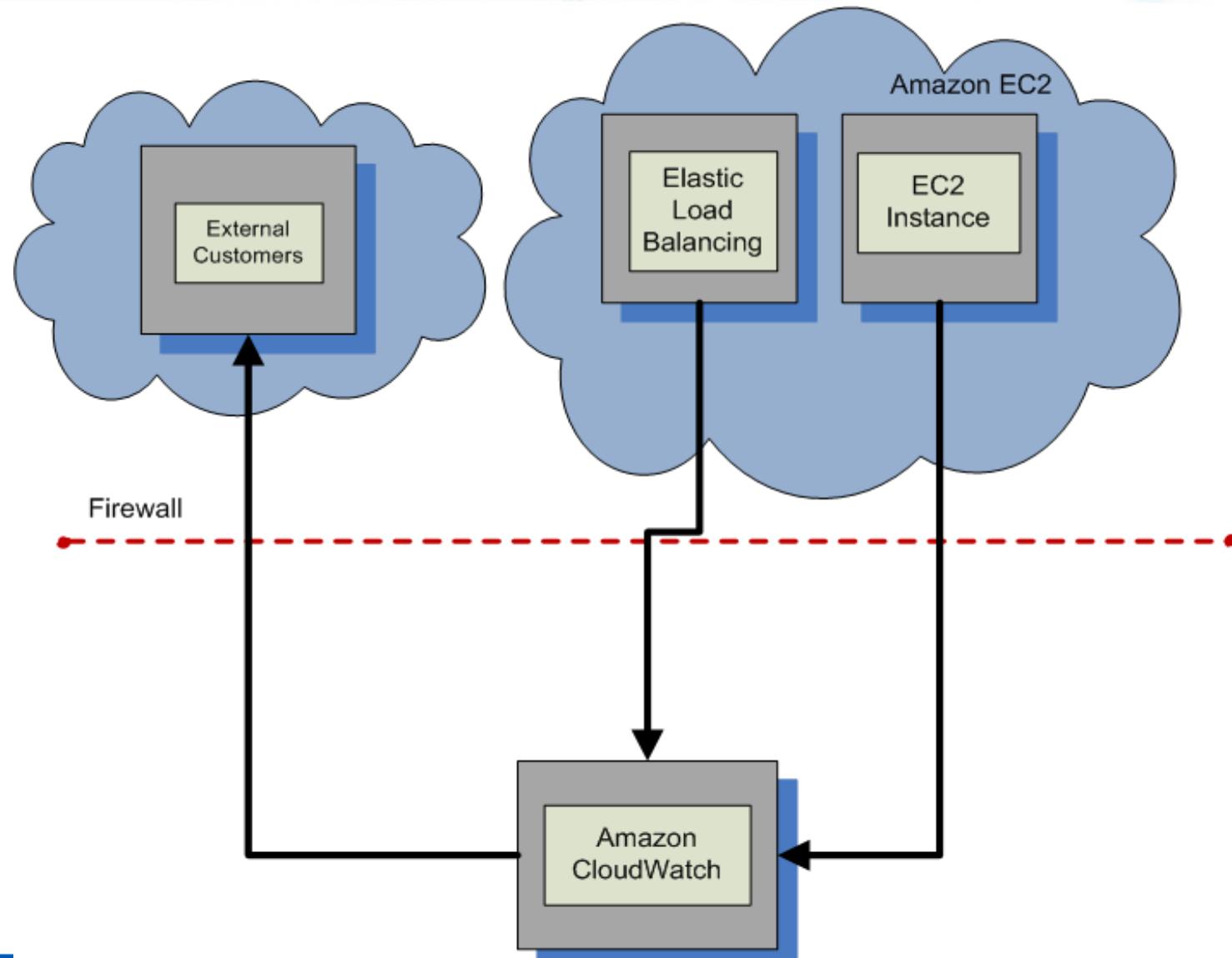
- 数据库用户管理
- 从节点管理
- 监控报表
- 数据库扩容

# 云的质量服务 --- 负载均衡 (AWS ELB)

- elb-create-lb
  - 输入: lb名, lb端口, instance端口, 协议 (HTTP, TCP) , zone
  - 输出: 创建的lb的dns
- elb-configure-healthcheck
  - 输入: interval, timeout, healthythreshold, unhealthythreshold
- elb-register-instance-with-lb
  - 输入: instance



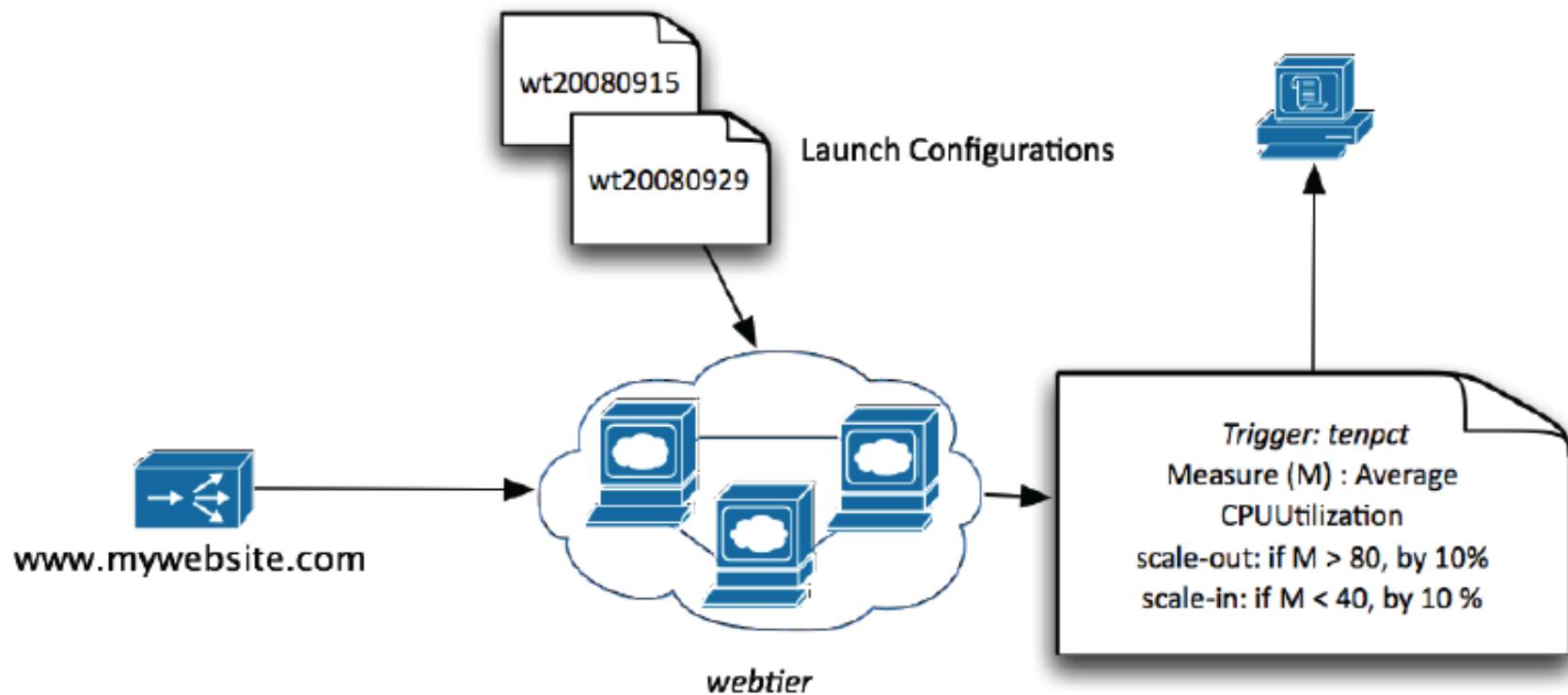
# 云的质量服务 --- aws cloudwatch



# Cloudwatch 监测内容

Amazon EC2	unit	Elastic Load Balancing	unit
CPUUtilization	Percent	Latency	Seconds
NetworkIn	Bytes	RequestCount	Count/Second
NetworkOut	Bytes	HealthyHostCount	Count
DiskWriteOps	Count	UnHealthyHostCount	Count
DiskReadBytes	Bytes		
DiskReadOps	Count		
DiskWriteBytes	Bytes		

# 云的质量 --- AWS Autoscaling



# 应用举例

- 目标：创建一个负载均衡的，可自动调节资源的EC2应用，共4步
- 1. 调用 `CreateLoadBalancer`，使用如下参数：
  - *AvailabilityZones* = us-east-1a
  - *LoadBalancerName* = MyLoadBalancer
  - *Listeners* = lb-port=80,instance-port=8080,protocol=HTTP
- 2. 调用 `CreateLaunchConfiguration` 使用如下参数：
  - *ImageId* = myAMI
  - *LaunchConfigurationName* = MyLaunchConfiguration
  - *InstanceType* = m1.small

### ■ 3. 调用 CreateAutoScalingGroup，使用如下参数：

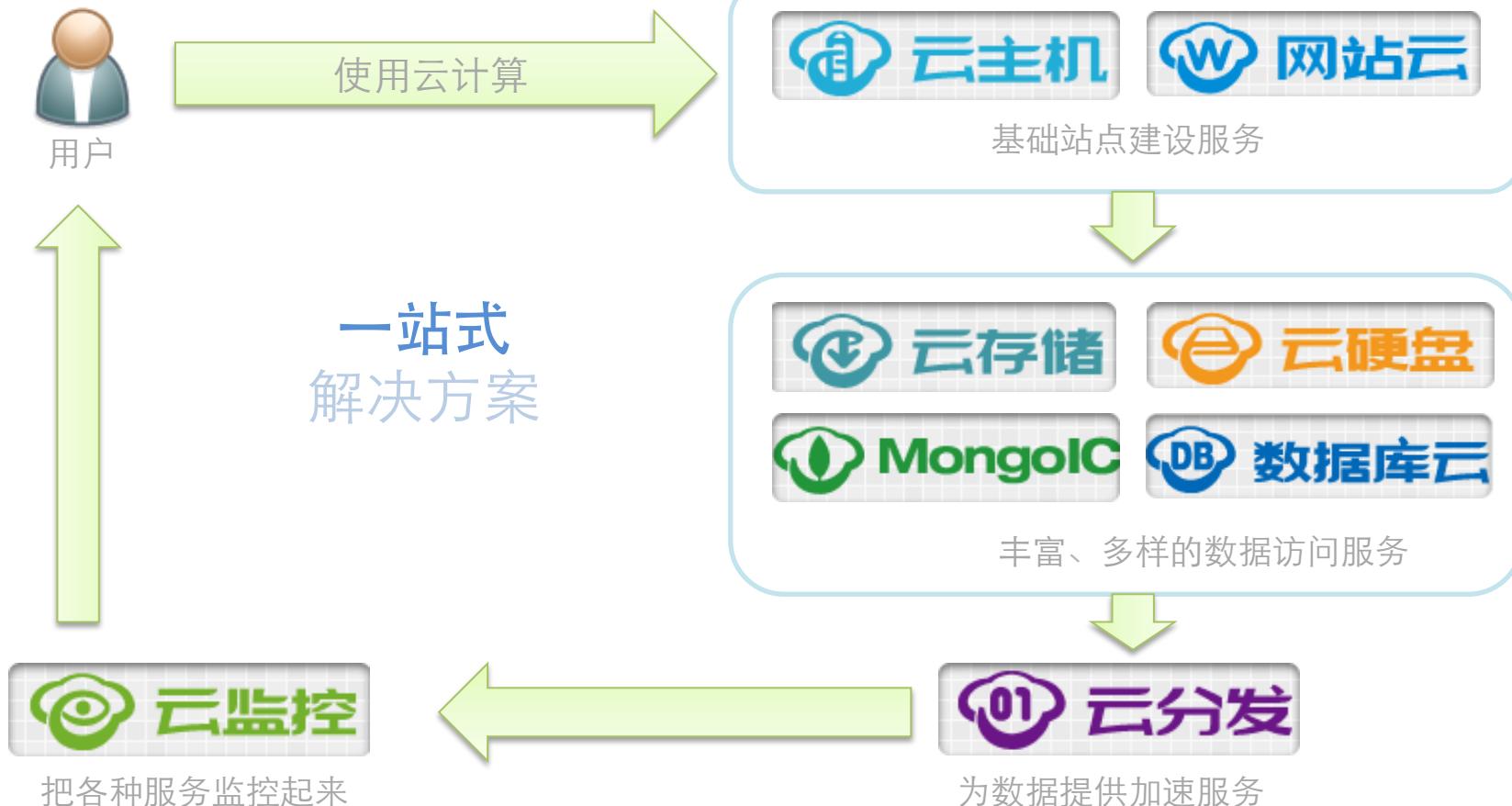
- *AutoScalingGroupName* = MyAutoScalingGroup
- *AvailabilityZones* = us-east-1a
- *LaunchConfigurationName* = MyLaunchConfiguration
- *LoadBalancerNames* = MyLoadBalancer
- *MaxSize* = 20      最多20个instance
- *MinSize* = 2 最少2个instance

■ 4. 调用CreateOrUpdateScalingTrigger，使用如下参数：

- *AutoScalingGroupName* = MyAutoScalingGroup
- *Dimension*: *Name*= AutoScalingGroupName *Value*= MyAutoScalingGroup
- *MeasureName* = CPUUtilization
- *Statistic* = Average
- *TriggerName* = MyTrigger1a
- *Namespace* = AWS/EC2
- *Period* = 60
- *LowerThreshold* = 40
- *LowerBreachScaleIncrement* = -1
- *UpperThreshold* = 80
- *UpperBreachScaleIncrement* = 1
- *BreachDuration* = 600

■ 5. 完成！！

# 云计算服务架构



# Thank You!