成本优化支柱

AWS 良好架构框架

2016 年 11 月

通告

本文档所提供的信息仅供参考，且仅代表截至本文件发布之日时 AWS 的当前产品与实践情况，若有变更恕不另行通知。客户有责任利用自身信息独立评估本文档中的内容以及任何对 AWS 产品或服务的使用方式，任何“原文”内容不作为任何形式的担保、声明、合同承诺、条件或者来自 AWS 及其附属公司或供应商的授权保证。AWS 面向客户所履行之责任或者保障遵循 AWS 协议内容，本文件与此类责任或保障无关，亦不影响 AWS 与客户之间签订的任何协议内容。

目录

摘要

本份白皮书主要面向 Amazon Web Services（简称 AWS）良好架构框架当中的成本优化支柱。其提供相关指导以帮助客户遵循最佳实践以立足 AWS 环境在各开发阶段当中实现成本优化，具体包括设计、交付与维护。

内容简介

在 Amazon Web Services，我们理解客户在评估当中所关注的架构最佳实践方案，希望借此指导自身立足云环境设计并运营一套可靠、安全、高效且具有成本效益的系统。作为这项工作的重要组成部分，我们开发出 AWS 良好架构框架，旨在帮助大家了解立足 AWS 进行系统构建时，各项相关决策中的优势与弊端。我们相信，良好架构系统将能够帮助企业客户显著提升成功机会。

此套框架基于五大支柱性因素：

- 安全性
- 可靠性
- 性能效率
- 成本优化
- 卓越运营

本份白皮书着眼于成本优化支柱，旨在阐述如何在保障系统架构高效利用服务与资源的同时，帮助客户享受最低使用成本。

您将借此了解到与成本优化支柱相关的各磺最佳实践。成本优化在传统内部解决方案当中往往极具挑战性，这是因为您需要立足成本归因与采购模式进行容量与复杂度猜测。在本份白皮书内相关实践指导的帮助下，您将能够构建起切实满足以下标准的架构方案：

- 您能够确保您的成本始终与预期相符。
- 您能够利用适当资源类型以尽可能降低成本。
- 您能够对成本进行归因与分析。
- 您能够随时间推移不断实现成本削减。

本份白皮书主要面向各技术类职能角色，具体包括首席技术官（简称 CTO）、架构师、开发人员以及运营团队成员。本份白皮书并不提供任何具体实施细节或者架构模式；但文末将提供与此类信息相关的扩展资源。

成本优化

成本优化属于贯穿整个系统生命周期的持续性提升与改进过程。由最初的概念验证性设计到生产工作负载的持续运营，您可采用本份白皮书当中提及的各项实践以构建并运营起具有成本效益的系统体系，从而在实现业务成果的同时最大程度降低成本水平，确保您的业务拥有最佳投资回报率。

一套成本优化型系统应充分利用全部资源、以最低价格点实现业务成果，同时充分满足您的功能需求。本份白皮书将面向工作负载设计、服务选取、服务配置与运营以及应用优化杠杆等层面为您提供深层指导。

设计原则

在探讨成本优化最佳实践时，大家应牢记以下设计原则：

- **采用消费模式：**仅根据您的业务消费要求（而非通过详细的预测）申请资源量，同时随时添加或减少资源需求，最终为您的实际消费量付费。举例来说，开发及测试环境在每周工作日内往往每天仅运行八小时。您可以在此类环境处于闲置状态时清空其资源分配量，从而实现高达 75% 的潜在成本节约效果（将原本的 168 小时运行时长降低至 40 小时）。
- **规模经济收益：**通过使用云计算，您有机会获得较自有方案更低的成本投入，这是因为庞大的 AWS 客户群体能够带来理想的规模经济效应。数十万家客户集中在 AWS 云当中，这意味着我们能够提供更为低廉的按使用量计费价格。
- **不再承担数据中心运营支出：**AWS 为您承担起服务器机架安装、部署以及电力供应等任务，因此您可以将精力集中在自身客户以及业务项目身上，而不再需要为 IT 基础设施分神。
- **支出分析与归因：**云环境能够帮助大家更轻松且更准确地统计系统使用情况与成本开销，这同时意味着我们能够更为透明地对 IT 成本进行归因。这不仅有助于实现投资回报率（简称 ROI）量化，同时亦可为系统拥有者提供良好的资源优化与成本削减机会。
- **使用托管服务以降低拥有成本：**在云环境当中，**托管服务**能够消除邮件发送或者数据库管理等任务给服务器维护工作带来的负担。另外，由于**托管服务**以云规模方式运行，因此亦能够带来更低的每事务或每服务实现成本。

定义

云环境下的**成本优化领域**共包含四项最佳实践：

1. 成本效率资源

2. 供需匹配
3. 支出认知
4. 随时间推移优化

与其它支柱一样，成本优化工作同样存在多种因素需要权衡。举例来说，您更强调上市速度抑或是成本水平？在某些情况下，速度优化最为重要——即更快将产品推向市场、更快发布新功能或者单纯满足进度表规划，而非单纯立足经济成本进行优化。有时候，设计决策可能较为匆忙而导致未遵照经验数据进行核查，这意味着设计者只能利用过度预估用量来“以防万一”，而无法真正根据时间基准测试制定出最具成本效益的部署方案。在这样的背景之下，大规模过度配置与未经优化的部署方案几乎都会存在。然而，这一切皆为将资源由内部环境“转移并提升”至云端的必要阶段。总体而言，我们的最佳实践要求避免在缺少明确成本优化策略的情况下盲目制定决策。以下各章节将着眼于初步及手续部署与环境层面的成本优化工作提供技术方案与最佳实践指导。

成本效益资源

作为成本节约的核心所在，您应当为自己的工作负载选择适当的服务、资源以及配置方案。在 AWS 当中，我们提供以下不同方案：

- 合理配置
- 正确规模
- 采购选项：按需实例、竞价实例与保留资源
- 地理选择
- 托管服务

您可以借助 AWS 解决方案架构师、AWS 参考架构以及 AWS 合作伙伴的力量结合自身的经验设计架构，同时基于测试数据去进一步优化架构。在敲定您的架构方案之后，您应利用一套数据驱动型流程调整资源类型与配置方案选择。利用基准与负载测试获取这部分数据。欲了解更多与上述议题相关的最佳实践信息，请参阅《AWS **良好架构框架性能效率支柱**》白皮书中的“审查”章节。

合理配置

总体而言，当您使用托管服务时，服务供应商将负责提供并管理相关底层资源。您也许会对某些服务属性进行设置，从而确保拥有充分的容量满足业务需求。您

可能还希望确保对这些属性加以优化调整，从而在控制超量容量处于合理水平的同时为最终用户提供良好性能表现。

AWS 利用 AWS 管理控制台或者 AWS API 及 SDK 为客户提供对托管服务属性的修改能力，意味着您可以借此确保 AWS 服务始终同您的业务需求及变化相匹配。举例来说，Amazon EMR 或者 Amazon RedShift 集群当中的节点数量可随时通过规模伸缩的方式进行增添与削减。

您亦可立足单一 AWS 资源建立多个实例以实现高密度资源利用率。举例来说，您可以在单一 Amazon 关系数据库服务（Amazon Relational Database Service ，简称 RDS）数据库实例当中配置多套小型数据库。在此之后，随着资源使用量的提升，您可以利用快照及恢复流程将其中部分数据库迁移至其它专用 RDS 数据库实例当中。

在对以托管服务为基础的系统进行配置时，您需要了解服务容量的调整要求。这些要求通常会给系统的正常运营带来时间、精力或者其它层面的影响。如果调整时耗较您的预期更长，则应考虑采取合理的过度配置方式以适应业务增长。在另一方面，您可以将 API 与 SDK 同系统以及 Amazon CloudWatch 等监控工具加以结合，从而将服务修改所需要的精力投入降低至几乎可以忽略不计。AWS Trusted Advisor 亦可对 Amazon RedShift 以及 Amazon RDS 等服务的资源使用量以及活跃连接进行监控。另外，您可能需要在系统维护窗口之内配置额外容量。

关键性 AWS 服务

合理配置层面的关键性 AWS 服务为 Amazon CloudWatch，其可帮助大家对各项核心指标进行收集与追踪。

正确规模

正确规模要求我们使用最低成本资源量，但同时亦确保满足特定工作负载提出的技术需求。您可以对各单一资源进行规模调整以实现成本优化。您可以反复调整资源规模以找到最为适合的规模水平。正确规模调整工作涵盖系统当中使用的一切资源类型以及每种资源的全部属性。正确规模调整工作可作为一种迭代式流程，由使用模式以及 AWS 降价或者新型资源发布等外部因素的变更所触发。

AWS 提供的各 API、SDK 及功能允许客户根据需求变化进行资源配置修改。举例来说，您可在 Amazon 弹性计算云（Amazon Elastic Compute Cloud ，简称 EC2）当中通过停止然后启动的方式对实例规模或者实例类型作出变更。或者，您亦可利用来自其它磁盘卷类型的快照对当前 Amazon 弹性块存储（Amazon Elastic Block Store ， 简称 EBS）磁盘卷进行恢复，从而通过增加 IOPS 以及/或者数据吞吐量实现性能改进。

在正确规模调整当中，您可利用资源与警报监控机制获取指导性数据。这类监控方案亦可触发更进一步的执行周期。您可利用 Amazon CloudWatch 以及 CloudWatch 定制化日志进行资源使用情况分析，后者可捕捉来自系统内任意组件的定制化指标并借此执行正确规模分析。举例来说，您可借此对内存使用量或者系统连接进行监控与分析。

在执行正确规模调整时，请务必重视以下注意事项：

- **监控机制必须能够准确反映最终用户的实际体验**。选择正确的时间段粒度标准，同时合理选择大多数或者 99%百分位数体验结论——而非简单取平均值。
- 为分析**时间段**选择正确的粒度水平，确保其切实涵盖各个系统周期。举例来说，如果您决定执行为期两周的分析工作，则可能会忽略掉为期一个月的高资源利用率周期，最终导致资源配置量不足。
- 根据正确规模收益因素对**修改成本**进行评估。举例来说，您应当计算执行变更所带来的业务成本，包括手动系统测试以及验证等。

关键性 AWS 服务

正确规模调整工作当中的核心 AWS 服务为 Amazon CloudWatch，其允许大家建立起监控体系以了解资源的实际利用情况。此外，以下服务同样发挥着重要作用：

- **AWS Trusted Advisor** 负责对资源进行分析，同时报告存在配置不足或者配置过量的资源类别。举例来说，Trusted Advisor 能够在 Amazon EC2 实例上在特定时间段内分析其 CPU 利用率与网络流量。在 Amazon EBS 方面，其则可分析各分卷的每秒输入/输出操作（简称 IOPS）活动，同时提供一套可用于指导正确规模调整工作的资源清单。

采购选项

在 AWS 当中，我们为大家提供多种不同采购模式以保证 Amazon EC2 实例的成本效益与您的业务需求相匹配。以下章节分别阐述了每一种采购模式：

- 按需实例
- 竞价实例
- 保留资源

按需实例

在按需购买 EC2 实例时，您需要以小时为单位支付平均费率，无需作出长期使用承诺且可根据实际应用需求随时增加或者减少实例内的计算资源。按需实例适用于运行周期较短（例如周期为 4 个月的项目）但可能定期出现峰值资源需求或者需求不可预测但无法接受中断的工作负载。按需实例亦适用于开发与测试环境等工作负载类型，即运行周期长于竞价实例但通常又短于保留资源的应用任务。

竞价实例

竞价实例适用于短周期工作负载（最高 6 个小时）或者能够以更为灵活的方式运行的应用程序类型。竞价实例允许大家对闲置 EC2 实例进行竞标，从而显著降低您的 Amazon EC2 资源使用成本。您亦可在必要周期内启动竞价时段（即 Spot blocks），其不会在竞价价格发生变化时引发服务中断。与按需实例费率相比较，Spot blocks 可将成本降低达 45%，而传统竞价实例则最高可实现 90%的成本节约。竞价实例适用于批量处理、科学研究、图像或视频处理、金融分析以及测试等用例类型。除了降低应用程序运行成本之外，您亦可利用竞价实例在不影响成本预算的前提下提升计算规模及数据吞吐能力。

竞价实例根据市场供需情况提供竞拍平台，您可在这里给出您愿为某一实例支付的最高出价——即竞标价。若您的竞标价达到或者超过当前现货价格，则您即中标并可在现货价格超出您的出价（以先超出者为准）之前一直使用对应资源。不同实例类型、规模以及可用区等因素可能使竞价实例的价格与可用性有所区别。

在使用竞价实例时，您应当认真考量以下三大关键性因素：

- 实例类型：如果您能够在实例类型与规模方面接受较为灵活的供应方式，那么竞价实例能够提供几乎最低的使用成本以及最低的工作负载中断可能性。
- 资源的具体位置：各个可用区皆属于独立市场。如果您对于工作负载的运行位置并无特别要求，则可享受由竞价实例带来的最低使用成本与中断可能性。
- 连续性设计：您应在工作负载设计当中充分考虑到频率中断状况可能带来的影响，并保证能够正常处理中断通知工作。

保留资源

AWS 为您提供多种方式以保留或者承诺使用特定资源数量，从而降低使用成本。举例来说，您可以在 Amazon CloudFront 当中选择保留容量计费（Reserved Capacity Pricing ）或者在 Amazon EC2 当中使用预留实例。利用预留实例，您需要承诺一段特定的使用周期（一年或者三年），并可在此期间享受较按小时计费的按需实例低出达 75%的费率优惠。您的预留实例会被分配至某一特定可用区，

其同样提供容量预留（即标准预留实例）功能，允许您仅在必要时启动一定数量的实例。对于需要稳定运行状态或者具备可预测资源利用量的应用程序，预留实例能够提供远优于按需实例的成本节约效果。

Amazon EC2 预留实例的三种类型：

- 标准型
- 区域优惠型
- 可转换型

标准预留实例能够提供计费与容量保留效益，但其具体家族、规模以及可用区皆固定且不可变更。区域优惠型跨越任何可用区为您的预留实例提供计费优惠，要求您指定实例家族与规模且不可变更，但不会为您预留容量。可转换预留实例要求您签订为期三年的使用承诺，允许您根据需求对家族、定价、实例规模、平台或者租户数量等属性进行变更，且提供与区域优惠型相同的计费折扣。

预留实例提供三种支付选项：

- 无预付——无需前期支付，为后续每小时使用量提供波动折扣。
- 部分预付——预付部分资源用量费用，在接下来的使用周期内享受可观的每小时使用量折扣。
- 全部预付——预付全部周期内资源用量费用，可在接下来的使用周期内随意使用资源而无需支付其它成本或者考虑具体使用时长。您可将这三种支付方式进行任意组合以匹配企业内的不同工作负载。

关键性 AWS 服务

采购支付选项层面的关键性 AWS 服务为 Amazon EC2 预留实例。相较于按小时计费的按需实例，利用预留实例可帮助您获得高达 75%的折扣幅度。此外，以下服务亦发挥重要作用：

- **竞价实例**能够较按需实例为您提供高达 90%的成本节约效果。

资源

欲了解更多与采购预留实例与竞价实例相关之 AWS 最佳实践信息，请参阅以下资源。

说明文档

- 竞价实例最佳实践
- Amazon EC2 预留实例
- 预留实例工作原理阐述

相关工具

- 竞价实例竞标顾问（Spot Bid Advisor）
- 竞价实例计费历史
- 如何采购预留实例

地理选择

对于电子商务以及众多其它网站而言，降低延迟对于改善使用体验能够起到关键性作用。因此在设计方案架构时，最佳实践要求尽可能保证计算资源贴近用户，从而提供更低延迟以及更符合要求的数据到达效果。面向全球受众群体，您可能需要选择多个地理位置以满足其使用需求。另外，您亦需要在地理位置选择当中最大程度降低使用成本。

AWS 在全球范围内的多个区域内运营有数据中心。各个 AWS 服务区皆立足当地市场情况进行运营，且各个服务区间的资源价格亦有所区别。您可选择特定服务区以运营您的特定组件或者整体解决方案，从而最大程度降低业务体系的全球运营成本。

利用 AWS 简单月度计费器（AWS simple monthly calculator ），您能够跨越多个服务区建立解决方案模型并比较其成本水平。您亦可使用 Amazon CloudFront 或者 AWS CodeDeploy 等服务在不同服务区内配置起概念验证环境，立足这些环境运行工作负载，并对各服务区内的特定及整体系统成本加以分析。

关键性 AWS 服务

AWS 服务区属于 AWS 云计算体系内的一项基础概念。AWS 基础设施由各服务区与可用区共同构建而成。当您在考查利用某一服务区运营特定组件或者整体解决方案时，您应考虑将解决方案迁部署在同等资源使用价格最低的服务区之内。

以下服务与功能在地理选择层面亦发挥着重要作用：

- **Amazon Route 53:** 当您使用多个 AWS 服务区时，可通过将相关请求与网络延迟最低的 AWS 服务区相对接的方式改善用户体验。Amazon Route 53 延迟路由服务允许您利用域名系统（简称 DNS）将用户请求路由至最佳 AWS 服务区，借以为用户提供最快响应速度。

- **Amazon CloudFront:** 在选择您的 AWS 服务区时，大家可以使用 Amazon CloudFront 以进一步降低延迟水平。Amazon CloudFront 可用于通过全球边缘站点分发您的整体网站，具体包括动态、静态、流式以及交互内容。指向您内容的请求将被自动路由最近边缘位置，以确保内容交付过程符合最佳性能要求。

资源

欲了解更多与地理选择相关的 AWS 最佳实践信息，请参阅以下资源。

说明文档

- AWS 全球基础设施
- 服务区与可用区

相关工具

- Amazon Web Services 简单月度计费器

托管服务

通过使用 AWS 提供的托管服务，您将能够摆脱以往服务交付过程中所必需的资源管理与维护等运营负担。另外，由于各项托管服务以云规模方式运营，因此其可提供更低的每事务或每服务成本开销。

AWS 面向 Amazon RDS 以及 Amazon DynamoDB 等数据库提供托管服务。通过使用这些托管服务，您能够削减或者彻底消除各类管理与运营日常开销，确保您将主要精力集中在增值业务活动身上。

您亦可以选择使用 AWS Lambda、Amazon 简单队列服务（Amazon Simple Queue Service ，简称 SQS）、Amazon 简单通知服务（Amazon Simple Notification Service，简称 SNS）以及 Amazon 简单邮件服务（Amazon Simple Email Service ， 简称 SES）等无服务器或应用级服务。这些服务能够帮助您摆脱代码执行、服务队列以及消息交付等工作带来的虚拟服务器管理负担。

使用托管服务能够有效降低基础设施管理成本，同时亦可帮助您的团队节约下大量时间资源，专注于构建各类增值型功能。如果时间压力较大，您应考虑使用上述托管服务。举例来说，您可能需要将内部环境“转移并提升”至云端，并决定稍后再行加以优化。最后，您会在使用过程中意识到，托管服务亦可显著降低甚至完全抵消掉许可授权类成本。

关键性 AWS 服务

支持托管方案层面的关键性 AWS 服务为各 AWS 数据库服务，例如 Amazon RDS 与 Amazon DynamoDB。这些服务能够降低数据库容量成本，同时帮助开发人员与数据库管理员节约大量时间。以下托管服务与功能亦在此层面发挥有重要作用：

- **AWS Lambda:** 这项无服务器计算服务可在无需底层基础设施成本的前提下实现代码执行。相关成本基于您所消费的具体请求数量和计算时长。
- **Amazon SQS、Amazon SNS 以及 Amazon SES:** 您可使用这些应用级服务且无需承担任何底层基础设施成本。AWS 的计费模式仅要求您按实际需求量使用，并为实际使用量付费。

供需匹配

最优供需匹配应能够为特定系统提供最低成本，同时亦需要具备充足的额外供应以控制供应时间并应对个别资源故障。需求可为固定或是可变形式，且应利用指标及自动化机制确保管理工作不致产生过高成本。

在 AWS 当中，您能够自动进行资源配置以匹配实际需求。以下章节将详尽描述这些方案的具体使用方法：

- 基于需求的方案
- 基于缓冲的方案
- 基于时间的方案

基于需求的方案

利用云计算提供的弹性优势能够在需求变化时始终给予有力支持，同时实现显著的成本节约效果。所谓**弹性**，是指云环境下几乎无限的容量供应能力；服务供应商负责容量管理与资源配置工作。通过利用 API 或服务特性，您可以以编程方

式动态地改变体系结构中云资源的数量。如此一来，您将能够对架构内的各组件进行规模伸缩，同时在需求达到峰值时自动增加资源配额以维持性能表现，并在需求量下降后释放资源以削减使用成本。

在 AWS 当中，我们通常利用 Auto Scaling 实现这种动态调整能力，其可帮助大家根据您预先定义的条件对 Amazon EC2 实例容量进行自动规模伸缩。另外，客户亦通常将 Auto Scaling 同弹性负载均衡（Elastic Load Balancing ，简称 ELB）配合使用，从而将跨越多个 Amazon EC2 实例的应用流量分发至 Auto Scaling 组当中。Auto Scaling 可根据规模伸缩计划中包含的政策及规模调整定义（包括手动、计划以及按需）进行触发，亦可配合 Amazon CloudWatch 当中的监控指标与警报执行响应性操作。各 CloudWatch 指标可用于触发计划内事件。您可使用 CPU 利用率、ELB 观察请求/响应延迟等标准 Amazon EC2 指标，甚至是来自 EC2 实例内应用程序代码的定制化指标。

Auto Scaling 可利用定制化代码或者指标触发规模自动伸缩，从而对业务事件作出响应。

在为基于需求的方案进行架构设计时，需要考虑到两大注意事项：首先，了解您对于新资源配置速度的具体需求。第二，了解供需之间余量的变化情况。您需要时刻准备应对需求变化，同时考虑到资源故障等问题的发生机率。

您可以通过减少启动程序量以及调整实例引导时的配置任务来优化配置速度。在这方面，建议大家使用预配置 Amazon Machine Image（简称 AMI）。需要注意的是，这项优化将对实例可配置能力造成影响。您需要根据需求在速度与可配置性之间找到平衡点。在初始开发时，您可能会发现自身架构的供需之间存在过大余量；请别担心，大家可以通过后续的测试与生产经历利用实验、负载测试等方式逐步削减这部分余量。

关键性 AWS 服务

基于需求的方案中最为关键的 AWS 服务为 Auto Scaling，其允许大家在无需超额支出的前提下根据需求添加或移除各类资源。以下服务亦在这一领域发挥有重要作用：

- **Amazon CloudWatch:** 利用 Amazon CloudWatch 可收集并追踪各项指标。您的架构可利用 CloudWatch 以观察需求及资源利用率、整合客户指标，同时触发并响应阈值与事件。
- **弹性负载均衡:** 立足云端跨越多个 Amazon EC2 实例进行输入应用程序流量的动态分发，从而帮助大家实现横向规模扩展。

基于缓冲的方案

基于缓冲的方案旨在利用队列接收来自生产程序的消息（即工作单元），从而实现供需匹配。出于保障弹性的考量，该队列应使用持久性存储介质。**缓冲区**机制负责保证各应用程序能够在随时间推移而以不同速率运行时，始终有能力彼此进行通信。各消息随后可供消费方读取，这就保证了各消息能够以与消费方业务需求相吻合的速率运行。通过使用基于缓冲的方案，大家可以将规程数据通量速率从消费方处解耦出来。总而言之，您不再需要考虑规程处理数据持久性及“**背压**”（因消费方运行速度缓慢导致生产方受到影响）的具体方式。

如果您的工作负载会产生大量不需要立即处理的写入负载，则可使用缓冲区机制以确保消费方始终拥有顺畅的运行效果。

在 AWS 上，您可以从多种服务当中作出选择以实现这类缓冲方案。Amazon 简单队列服务（简称 SQS）提供的队列可供单一消费方读取单一消息。Amazon Kinesis 则可面向多个消费方提供可供读取的通用消息流。

Amazon SQS **队列**属于一套可靠、可扩展且全面受控的待处理消息库。Amazon SQS 使得各类应用程序能够快速且可靠地将由某一应用程序组件生成并由另一组件消费的消息纳入队列。在这套方案当中，队列作为生产方消息的发送目的地，且该消息被驻留于全局可知的地址当中。

为了降低成本，生产方利用 Amazon SQS 批量 API 操作进行消息发送。消费方利用一套固定大小的 EC2 实例 Auto Scaling 组应对实例故障，并从全局可知队列中读取消息。**长轮询**则允许大家在 Amazon SQS 队列可用后随时从其中检索消息。利用长轮询机制可降低消息检索成本。Amazon SQS 还利用**消息可视性**功能对已经读取完毕的消息进行隐蔽。消息可见性能够降低您重复处理同一消息内容的风险，不过需要注意的是，Amazon SQS 默认执行**至少一次交付**，意味着您可以多次收取同一条消息。

如果需要保证仅进行一次处理，大家则可使用 Amazon SQS 先进/先出（简称 FIFO）队列。消息可见性功能允许各尚未经过处理的消息重新出现（例如在发生实例故障的情况下）。对于长期运行的任务，您可以扩展可见性超时，这时您的应用程序需要在消息被处理完成后对其进行主动删除。

另一种方案在于利用 Amazon Kinesis 提供缓冲区。与 Amazon SQS 的区别在于，Amazon Kinesis 允许多个**消费方**随时读取同一消息。然而，单一将利用 Kinesis 客户端库或者 AWS Lambda 进行读取，而后被交付至一个且惟一一个消费方—

—即“**应用程序**”，此名称在消费方接入流时提供。不同的“**应用程序**”可同时消费同一批消息，相当于建立起一套“发布与订阅”模式。

在利用基于缓冲的方案设计架构时，请注意两大关键性因素：其一，工作生产与工作消费之间的延迟水平接受范围如何？第二，您计划如何处理重复的工作请求？

为了在由多个消费方消费工作条目的场景下优化速度表现，大家可以使用 Amazon 竞价实例。利用竞价实例，您可以竞标备用 Amazon EC2 计算容量。要处理 Amazon SQS 中的重复消息，我们建议您使用 Amazon SQS（FIFO）队列，其能够实现仅一次处理效果。

对于 AWS Kinesis，大家可以考虑在 Amazon DynamoDB 当中配置一套表，用于追踪已经被成功处理的工作条目。另一种重复消息处理方式则为幂等处理，其更适用于对数据吞吐量要求更高的场景。**幂等**机制允许消费方利用应用程序的逻辑模式对消息进行多次处理，且消息在按序处理时不会影响到下游系统或者存储。

关键性 AWS 服务

对于基于缓冲的方案，关键性 AWS 服务为 Amazon SQS 与 Amazon Kinesis——其能够轻松且以符合成本效益的方式对云应用程序中的各个组件进行解耦。以下服务与功能同样在这类场景中发挥重要作用：

- **Amazon EC2 竞价实例：**允许您竞标备用 Amazon EC2 计算容量，并借此以更低成本处理工作条目。
- **AWS Lambda：**提供一套无服务器方案以处理消息，从而消除运行 EC2 实例带来的固定成本支出。

基于时间的方案

基于时间的方案能够为可预测或者按时间进行明确定义的需求分配资源容量。这类方案通常并不依赖于资源利用率。基于时间的方案能够确保各类资源在特定必要时间段内处于可用状态，同时凭借启动规程及系统或者一致性检查实现资源的无延后交付。利用基于时间的方案，您可以在业务繁忙时段提供额外的资源或者增加容量上限。

在 AWS 当中，您可以在 Auto Scaling 当中利用计划内规模伸缩以建立这类基于时间的方案。系统可以根据计划在特定时间段内进行规模扩展或者收缩——例如工作日时段，从而确保各资源在用户需要时提供理想的可用性。

您可以利用 AWS API 与 AWS CloudFormation 自动配置并停用整体环境。这类方案特别适合仅以预定义方式在工作时间或者其它特定时间段内运行的开发或者测试环境。

您可以利用 API 在环境之内对资源进行规模伸缩。举例来说，您可以通过变更实例大小或者类对生产系统进行向上扩展。您亦可通过选择不同的实例大小或者类停止及启动实例。另外，这种方式也适用于其它资源，例如 Amazon 弹性块存储（简称 EBS）分卷。您可以创建快照并纳入更高每秒输入/输出操作（简称 IOPS）或者不同分卷类型以恢复该分卷。

在设计基于时间的方案时，请重视以下两大注意事项：其一，您的使用模式拥有怎样的一致性？第二，模式变更会造成怎样的影响？您可以通过系统监控或者使用商务智能工具等方式提升预测准确度。如果您发现使用模式发生了显著变化，则可调整具体时间以确保整体方案仍能涵盖当前使用模式。

关键性 AWS 服务

在支持基于时间的方案方面，最为关键的 AWS 服务为 Auto Scaling，其允许大家在无需承担日常开支的前提下根据需求添加或者移除各类资源。以下服务在这方面同样发挥着重要作用：

- **AWS CloudFormation:** 为您提供多套模板，可用于创建 AWS 资源并以有序且可预测的方式进行配置。其非常适用于创建各类短周期环境，例如测试环境。

支出认知

了解业务成本驱动因素对于高效管理支出并确定成本削减机会至关重要。大多数企业都由多支团队以及其下的多套系统共同构成。这些团队可能来自不同业务部门，且各自拥有自己的收入来源。将资源成本归因至个别业务或者产品拥有者能够有效驱动资源使用活动，同时有助于降低资源浪费。精准的成本归因亦可帮助大家理解产品的真实利润空间，同时以更为明智的方式作出预算划分决策。

您应考虑采用多因素方法以建立支出认知。您的团队需要收集数据、加以分析并整理报告。以下为这方面的重要注意事项：

- 相关方
- 可见性与控制能力
- 成本归因

- 标签
- 实体生命周期追踪

相关方

企业当中的各相关方需要参与到云计算各个阶段的支出讨论当中。以下为典型的参与相关方及其角色类别：

- 财务：首席财管官/财务总监必须了解云消费模式、采购选项以及月度结算流程与相关数据（详尽的计费与/或使用量文件）。由于云（按使用量计费以及详尽的计费与使用量信息）同内部运营在成本方面存在巨大差异，因此财务团队必须有能力立足这种新型运营与信息模式继续履行自身职能。
- 管理：管理团队必须了解云商业模式，从而确保能够为各业务部门以及企业整体的发展提供方向性指引。这类云知识在增长预测以及系统使用方式层面表现得尤为重要，同时要求管理层及时评估不同采购选项（例如保留资源）。
- 技术领导者：技术领导者必须根据来自财务及管理相关方的外部业务、资源使用情况或者成本因素对系统进行归因或者调整。如此一来，系统方能够切实支持并实现预期业务目标。
- 第三方：如果您的企业拥有第三方合作伙伴，请确保其遵循您的财务目标并通过其参与模式表现出这种遵循能力。一般来讲，第三方可对其负责管理的系统进行报告与分析，同时为其设计的系统提供成本分析结论。

可见性与控制能力

您应有能力了解成本细目、预测未来成本并在实际支出超出预期水平之前作出响应，从而保持对成本基础的控制能力。

AWS 提供一整套报告与工具，供您对相关 AWS 支出进行估算、监控、计划、报告与分析。AWS 计费与成本管理服务（AWS Billing and Cost Management ）能够帮助您实现以下目标：

- 估算并预测您的 AWS 使用成本
- 如果成本水平超出您设置的阈值，则发送警报
- 评估您在 AWS 资源层面的最大投入
- 分析您的支出与使用量数据

考虑利用免费的成本管理器（Cost Explorer）工具显示并分析您的成本。这款工具允许您以图形化方式查看年度成本，同时以高于 80%的可信水平预测未来几个月内的预期支出额度。您可以利用成本管理器按时间推移查看 AWS 资源消费模式，根据需求进行更为细致的深入查询，同时了解成本支出的变化趋势。

AWS 计费与成本管理服务提供的预算结论可供您为自身 AWS 成本支出制定月度预算水平、总体成本水平（即全部成本）或者更为细化的成本核算级别——您可在细化成本中指明特定条目，包括关联帐户、服务、标签或者可用区。您亦可在成本当中纳入邮件通知机制，其会在您的当前或者预期成本超过预设百分比阈值时发送通知邮件。

您可以利用 Amazon CloudWatch 警报与通知以创建计费警报，从而在服务使用量超出所定义的财务阈值时发布通知。您亦可定义其它离散维度，包括服务、关联帐户或者关联帐户加服务。

包含资源与标签的详尽计费报告（简称 DBR-rt）及成本与使用量报告能够提供更为细化的报告级别。其中囊括有每小时资源使用量以及各项可计费 AWS 服务产生的具体费用。这些报告可帮助大家确切了解系统的使用情况与成本水平，且以小时为基本计量单位。为了对此类数据进行分析与报告，大家可以使用 Amazon Redshift 承载计费数据，利用第三方应用程序处理这部分计算数据，并将计费数据导入至现有财务系统当中。

计费与成本管理服务被集成于 AWS 身份与访问管理（简称 IAM）服务当中。您可以利用 IAM 服务配合计费与成本管理通过计费控制台控制指向财务数据以及各 AWS 工具的访问请求。

成本归因

您可以利用成本归因机制为企业内的各个单位分配成本额度，具体包括各部门、业务单位、产品或者内部团队，从而实现成本管理。

为了进行 AWS 成本组织与管理，您可以考虑使用帐户结构与标记方案。单独使用或者将二者结合能够帮助您顺利完成面向业务门类的成本归因工作。

帐户结构。AWS 拥有一套一主多从帐户结构，我们通常将其视为一支付方（主）帐户与多链接（子）帐户体系。您可以采取单一 AWS 帐户或者多 AWS 帐户结构。当然，并不存在能够适应一切 AWS 帐户结构的解决方案。大家应当首先评估自身当前及未来运营成本模式，从而确保您的 AWS 帐户结构能够充分反映企业的实际情况。部分企业可能出于以下几种常见理由而需要建立多个 AWS 帐户：

- 要求各业务单位、成本中心或者特定工作负载之间实现管理与/或财务与计费隔离。
- AWS 服务限制无法满足特定工作负载的资源需求。

- 某些工作负载需要配合特定预留实例（例如出于高可用性及灾难恢复需求）。

合并计费机制用于在一个或者多个关联帐户与支付方帐户之间建立结构。各关联帐户允许您根据预定义分组对对应使用量及计费方式进行隔离与区分。其中一类常见作法在于为各个业务单位分别建立关联帐户（例如财务、营销与销售等），或者面向各环境-生命周期（例如开发、测试与生产）抑或各个项目（例如项目A、B 与 C）分别建立关联帐户，而后将各关联帐户加以合并从而计算总体成本。合并计费允许您立足单一支付方帐户处理多个 AWS 帐户的使用成本，同时仍然保持各个关联帐户的活动可见性。由于成本与使用情况会被汇总至支付方帐户当中，因此您将能够最大程度享受资源费率折扣、预留实例批量折扣以及预留实例优惠。

标签

标签机制允许您将业务与组织信息同计费与使用量数据进行匹配。这种作法能够帮助大家准确分类并追踪成本与有意义业务信息间的关联。您可以利用此类标签表达业务类别（例如成本中心、应用程序名称、项目或者持有方），从而跨越多项服务及多个团队实现成本规划。

当为 AWS 资源（例如 Amazon EC2 实例或者 Amazon 简单存储服务（简称 S3）存储桶）应用标签并激活时，AWS 会将相关信息添加至详尽计费报告与成本配额报告当中。其中成本配备报告囊括配合标记信息的计费摘要内容。

通过为资源分配标签，大家还能够进行更高级别的自动化与易用性管理。您可以通过罗列全部包含特定管理任务执行标签的资源，而后执行对应操作。举例来说，您可以列出一切拥有 environment:test 标签值的资源，而后对各项资源进行删除或者终止。这种方式适用于在每天工作时段结束时自动关闭或者移除测试环境。为已标记及未标记资源运行报告处理功能，则可帮助大家更为严格地满足内部成本管理政策的具体要求。

在您企业当中各个帐户之上以标准化方式执行 AWS 标记操作能够显著提升 AWS 环境管理工作的一致性与统一度。

实体生命周期追踪

在您管理一份包含项目、员工以及随时间推移技术资源变更的清单时，您可以识别出其中不再使用的资源或者不再具备持有方的孤立项目。

AWS 提供多项可用于实体生命周期追踪的对应服务选项，例如 AWS Config——其能够为您提供 AWS 资源详细清单并进行配置，同时持续记录各项配置变更。AWS CloudTrail 与 Amazon CloudWatch 则允许您您为生命周期事件建立资源记

录。IAM 允许您管理各组、用户以及角色，同时利用现有身份供应方提供的企业内工作人员记录源联动信息建立可信保障。利用联动机制可降低在 IAM 内创建大量用户的需求，且可继续利用现有身份、凭证以及已经在企业内建立的角色访问机制。

您可以集成现有项目或者资产管理系统，并将其与企业内的活动项目与产品相匹配。通过将现有系统同 AWS 提供的丰富事件与指标集相整合，您将能够构建起一套重大生命周期事件视图，且主动管理各项资源以降低不必要成本。

关键性 AWS 服务

支持支出认知场景的关键性 AWS 服务为 AWS 计费与成本管理服务。以下服务与功能亦在这方面发挥有重要作用：

- **成本管理器：**可利用此工具实现成本可视化与分析。
- **标记：**利用标签将业务与组织信息同您的计费与使用量数据加以匹配。
- **Amazon CloudWatch 警报：**创建结算通知，用于告知您服务使用量何时超出预先定义的财务阈值。

资源

请参阅以下资源以了解更多与支出认知相关的 AWS 最佳实践。

说明文档

- AWS 标记策略
- AWS 计费与成本管理
- Amazon CloudWatch 警报

相关工具

- AWS 成本管理器

随时间推移进行优化

在 AWS 当中，您可以利用一系列不同方案实现随时间推移的成本优化目标。以下章节将阐述如何切实运用这些方案：

- 衡量、监控与改进
- 始终更新（移动至最新服务、功能与实例类型）

衡量、监控与改进

当您对用户及应用程序进行衡量与监控时，以及将您收集到的数据同 AWS 平台监控数据相结合时，您可以执行差距分析以了解系统利用率与您当前要求间的一致性差异。通过持续调整，您能够尽可能缩小这种差异并确保您的系统符合成本效益要求。以下四种关键性方法有助于实现这一目标：

- 建立成本优化功能
- 建立目标与指标
- 收集见解并进行分析
- 报告与验证

关于指标收集以及监控机制构建的相关最佳实践，请参阅《AWS **良好架构框架**：

可靠性支柱》白皮书内的“监控”章节。

建立成本优化功能

考虑在企业之内建立起一项匹配成本优化责任与归属方针的功能。此功能可由云卓越中心等现有团队负责执行，亦可为其单独建立起一支由关键性相关者（主要来自企业内各业务部门）组成的新团队。此项功能将协调与成本优化相关的各项管理工作，且涵盖技术系统、人力资源乃至业务流程。投资以获取符合需求的 AWS 支持服务级别能够帮助您确切执行此项功能。AWS 企业支持服务为您提供一位专项技术客户经理（简称 TAM），其将负责各项具体优化工作。TAM 拥有深厚的技术积累与先进工具方案，能够为您的企业提供极具指导意义的报告与分析结论。

建立目标与指标

建立起目标与指标，以帮助您的的企业用于衡量自身工作进展。目标与指标可立足于宏观与微观两大层面，且必须与各平台及系统的最终用户保持一致。立足宏观层面，其目标将跨越各帐户与企业部门，而微观层面的着眼点则针对个别系统。

以下为部分企业目标示例：

- 按需 EC2 弹性：每天开启及关闭之按需 EC2 实例百分比。这一指标应尽可能接近 100%，且实际处于 80%到 100%区间。在衡量过程中，我们可以将工作时段启用的实例数量除以任意时段内运行的最大实例数量。
- 总体弹性：每天开启及关闭之按需与预留实例百分比。这一指标应尽可能接近 100%，且实际处于 80%到 100%区间。不过如果大家的运营系统以 24/7 全天候方式运行，则实际范围可进一步延伸。如果您的系统峰值在工作时段内由 20%增长至 100%，那么弹性水平应该按照 80%的水平设计。
- 基础负载预留实例覆盖率：即运行在保留容量之上的“始终开启”实例数量。企业应当在这方面实现 80%到 90%的覆盖率。

这份列表可作为其它资源发展目标的指导性资料，具体包括 Amazon EBS 分卷利用率、Amazon RDS 存储资源利用率以及 Amazon ElastiCache 内存利用率。

收集见解并进行分析

相关工具允许您衡量并监控您的平台利用率，同时为业务运行状况提供见解与分析结论。要选择工具时，请尽可能使用那些能够告知客户正在做什么以及系统组件如何响应这种使用方式的方案选项。

AWS 提供以下工具以实现资源使用情况的衡量与监控：计费与成本管理仪表板（其中包括成本管理器与预算服务）、Amazon CloudWatch 以及 AWS Trusted Advisor。

报告与验证

利用以上工具与分析机制生成的报告应进行定期审查，从而确保资源分配方式与您预先定义的目标保持一致并同步发展。其亦可用于验证各项成本衡量结论——例如由保留容量带来的成本削减数额，并根据需要进行调整。

始终更新

随着 AWS 不断推出新型服务与功能，另一大最佳实践在于审查您的现有架构决策以确保其仍然符合最佳成本效益。如果您的要求发生变化，则应主动清退那些您不再需要的资源、整体服务或者系统。

AWS 提供的托管服务通常能够显著实现解决方案优化，因此在上线之后，大家应积极对其进行了解。举例来说，运行 Amazon RDS 数据库在成本方面要远优于立足 Amazon EC2 运行您自己的数据库。

关键性 AWS 服务

随时间推移进行优化工作中的关键 AWS 功能为 AWS 博客以及 AWS 官方网站上的**最新消息**频道。您可定期加以访问以了解 AWS 新近发布的各类功能与服务。

以下服务亦在随时间推移进行成本优化方面发挥着重要作用：

- **AWS Trusted Advisor:** 检查您的 AWS 环境并发现各类可通过清除未使用或闲置资源，或采用预留实例容量以实现的成本节约机会。

资源

欲了解更多与随时间推移成本优化工作相关的 AWS 最佳实践信息，请参阅以下资源。

说明文档

- AWS 博客
- AWS 最新消息频道
- AWS Trusted Advisor

相关工具

- 计费与成本管理仪表板

结论

成本优化是一项需要持续不懈的工作。从初步试水与审查到工作负载测试，再到立足 AWS 部署成熟的生产基础设施，您应定期审查自身架构方案及组件选择。而在本份白皮书所提及的各类程序化功能与 AWS 服务的有力支持之下，这项工作将会变得更为简单明了。

AWS 一直致力于帮助您最大限度降低成本水平，同时建立起具备高弹性、响应能力以及自适应性的部署机制。本份白皮书的核心主旨，在于帮助您真正在部署工作中凭借各类工具、技术以及最佳实践实现成本优化目标。

贡献者

以下个人及组织为本文的编撰工作作出贡献：

- Philip Fitzsimons，Amazon Web Services 良好架构高级经理
- Nathan Besh，Amazon Web Services 企业支持服务经理
- PT Ng， Amazon Web Services 商业架构师
- Arthur Basbaum，Amazon Web Services 企业开发者经理
- Jarman Hauser，Amazon Web Services 商业架构师

扩展阅读

若您需要更多帮助，请参考以下资料：

- AWS 良好架构框架