AWS Well-Architected Framework

2015 年 10 月

版权声明

目录

摘要

本文介绍了 **AWS Well-Architected Framework**，该框架可供客户评估和改进其基于云的架构并更好地了解其设计决策对业务的影响。我们在四个概念领域提出了一般设计原则以及具体最佳实践和指南，并将这些概念区域定义为 Well-Architected Framework 的*支柱*。

简介

在 Amazon Web Services (AWS)，我们了解对客户进行架构最佳实践方面的教育对于在云中设计可靠、安全、高效且具有成本效益的系统的价值。作为此工作的一部分，我们开发了 AWS Well-Architected Framework，它将帮助您了解您在 AWS 上构建系统时所作的决策的优缺点。我们相信，部署精心构建的系统能够极大地提高业务成功的可能性。

AWS 解决方案架构师拥有多年的架构设计解决方案方面的经验，涉及各种业务垂直市场和使用案例，我们已帮助设计和审核了数千位客户在 AWS 上的架构。借此，我们确定了在云中构建系统的最佳实践和核心策略。AWS Well-Architected Framework 记录了一系列基本问题，这些问题可让您了解某个特定架构是否很好地与云最佳实践保持一致。此框架提供了一个一致的方法，用来评估系统是否达到了您希望从现代化的基于云的系统得到的质量，并提供了实现这些质量所需的补救措施。随着 AWS 平台继续发展，以及我们继续从与客户的合作中获得经验教训，我们将继续完善“精心构建”这一定义。

本文面向担任技术角色的人员，如首席技术官 (CTO)、架构师、开发人员和运营团队成员。阅读本文后，您将了解在设计云架构时可使用的 AWS 最佳实践和策略。本文未提供实施详细信息或架构模式，但包含对此信息的相应资源的引用。

AWS Well-Architected Framework 的定义

AWS 的专家每天都帮助客户构建系统以利用云中的最佳实践。我们根据您的设计的进度与您一起进行架构权衡。随着您将这些系统部署到真实环境中，我们将了解这些系统的表现以及这些权衡的后果。

我们根据当前经验创建了 AWS Well-Architected Framework，此框架是一组问题，这些问题可用来评估架构与 AWS 最佳实践的一致程度。

AWS Well-Architected Framework 基于安全性、可靠性、性能效率和成本优化这四个支柱，我们对这四个支柱的定义如下：

支柱名称	描述
安全性	通过风险评估和缓解策略在提供业务价值的同时保护信息、系统和资产的能力。
可靠性	系统从基础设施故障或服务故障恢复、动态获取计算资源以满足需求以及减少中断（如错误配置或暂时性网络问题）的能力。
性能效率	有效地使用计算资源以满足系统要求的能力以及在需求变化和技术改进时保持此效率的能力。
成本优化	避免或消除不必要的成本或不够理想的资源的能力。

一般设计原则

Well-Architected Framework 确定了一组一般设计原则以帮助在云中打造优秀的设计：

- **停止猜测您的容量需求：**消除对基础设施容量需求的猜测。当您在部署系统前制定容量决策时，您最终可能要面对成本高昂的闲置资源，或者承受由于容量有限而对性能产生的影响。而利用云计算，这些问题都不会出现。您可以根据需要使用尽可能多或尽可能少的容量，并自动向上或向下扩展。
- **在生产规模下测试系统：**在传统的非云环境中，单独创建用于测试的重复环境通常成本高昂。因此，大多数测试环境都未在真实的生产需求水平下进行测试。在云中，您可按需创建重复环境，完成您的测试，然后停止使用资源。由于该测试环境仅在运行时产生费用，因此您只需支付本地测试的费用的零头即可模拟您的真实环境。
- **降低架构更改的风险：**由于您可以自动创建模拟生产配置的测试环境，因此您可以轻松执行测试。您还可消除本地环境中出现的测试序列化，从而让团队在使用测试资源时避免排队。
- **通过自动化来简化架构实验：**自动化使您能够以低成本创建和复制您的系统（无需人工操作）。您可跟踪对自动化的更改、审核影响以及在必要时还原到之前的参数。
- **支持进化式架构：**在传统环境中，架构决策通常是作为静态、一次性的事件实现的，同时系统在其生命周期内存在几个主要版本。由于业务及其环境持续变化，这些初始决策可能会影响系统满足不断变化的业务需求的能力。在云中，您可以进行自动化的按需测试，从而降低由于设计变化而产生影响的风险。这样，系统便能随着时间的推移而演变，从而使企业能够利用新的创新作为标准实践。

Well-Architected Framework 的四个支柱

创建软件系统与建造大楼非常相似。如果基础不牢固，则可能出现破坏建筑的完整性和功能的结构问题。在制定技术解决方案时，如果忽略安全性、可靠性、性能效率和成本优化这四个支柱，则可能难以构建能满足您的期望和要求的系统。将这些支柱包含到设计中有助于打造稳定且高效的系统。这使您能够专注于设计的其他方面，如功能要求。

本节将逐一介绍这四个支柱，包括定义、最佳实践、问题、考虑事项和相关的关键 AWS 服务。

安全支柱

安全性支柱包含通过风险评估和缓解策略在提供业务价值的同时保护信息、系统和资产的能力。

设计原则

在云中，有很多可帮助您提高系统安全性的原则。

- **在所有层面应用安全性：**不是仅在基础设施的边缘运行安全设备（如防火墙），而是对您的所有资源（例如所有虚拟服务器、负载均衡器和网络子网）使用防火墙和其他安全控制。
- **启用可追溯性：**记录并审核针对您的环境的所有操作和更改。
- **自动响应安全事件：**监控并自动触发对事件推动或条件推动的警报的响应。
- **专注于保护您的系统：**利用 AWS 责任共担模型，您可以专注于保护您的应用程序、数据和操作系统，同时让 AWS 提供安全的基础设施和服务。
- **自动实施安全最佳实践：**基于软件的安全机制提高了您更加快速、实惠和安全地进行扩展的能力。创建并保存虚拟服务器的自定义基线映像，然后在您启动的每台新服务器上自动使用该映像。创建在模板中定义和管理的完整基础设施。

定义

云中的安全性包括以下四个方面：

1. 数据保护
2. 特权管理
3. 基础设施保护
4. 探测性控制

AWS 责任共担模型使采用云的组织能够实现其安全性和合规性目标。由于 AWS 在物理上保护用于支持我们的云服务的基础设施，因此 AWS 客户可以专注于使用服务来实现其目标。AWS 云还方便了对安全数据的访问并提供了响应安全事件的自动化方法。

最佳实践

数据保护

在构建任何系统之前，应该先建立起影响安全性的基本实践。例如，*数据分类* 提供了根据敏感度级别为组织数据分类的方法；*最小特权* 在保证正常功能的同时将访问权限限制为尽可能低的级别；而*加密* 通过以无法识别的形式向未经授权的访问呈现数据来保护数据。这些工具和技术很重要，因为它们有助于实现目标（如防止经济损失或遵守法规义务）。

数据保护涉及使用一些控制和模式，这些控制和模式旨在保持您的数据机密性，同时保持数据的完整性并确保数据在您需要时可用。

在 AWS 中，以下实践有助于保护数据：

- AWS 客户保持对其数据的完全控制。
- AWS 简化数据加密和密钥管理，包括定期密钥轮换（可由 AWS 轻松地在本机自动完成也可由客户维护）。
- 提供详细的日志记录，包含文件访问和更改等重要内容。
- AWS 对存储系统进行了专门设计，旨在提供极强的复原能力。例如，Amazon Simple Storage Service (S3) 凭借优越的设计达到了 99.999999999% 的耐用率。（例如，如果您使用 Amazon S3 存储了 10,000 个对象，预计平均每 10,000,000 年才会丢失一个对象。）
- 版本控制（可能是较大的数据生命周期管理过程的一部分）可防止意外覆盖、删除和类似危害。
- AWS 绝不发起区域之间的数据移动。放置在某个区域中的内容仍将保留在该区域，除非客户显式启用了某个功能或利用了提供该功能的服务。

以下问题重点关注数据安全性的考虑事项（有关安全性问题、答案和最佳实践的列表，请参阅附录）：

章节 1. 您如何加密和保护静态数据？

章节 2. 您如何加密和保护传输中的数据？

AWS 提供了多种加密静态数据和传输中的数据的方法。我们将功能内置于我们的产品和服务中，这些产品和服务简化了数据的加密。例如，我们已实施适用于 Amazon S3 的服务器端加密 (SSE) 以方便您以加密的形式存储数据。您还可安排将由 Elastic Load Balancing 处理的完整 HTTPS 加密和解密过程（一般称为“SSL 终端”）。

特权管理

特权管理是信息安全计划的关键部分；它确保了只有经过授权和通过身份验证的用户才能访问您的资源，并且只能以应有的方式访问。例如，访问控制列表 (ACL) 是附加到对象的访问权限的列表，基于角色的访问控制 (RBAC) 是与最终用户的角色或职能一致的权限集，而密码管理包含复杂性要求和更改间隔。这些特权管理元素在信息安全架构中至关重要，因为它们代表了用户身份验证和授权的核心概念。

在 AWS 中，特权管理主要由 AWS Identity and Access Management (IAM) 服务提供支持，该服务允许客户控制用户对 AWS 服务和资源的访问。您可以应用粒度策略，以便向用户、组、角色或资源分配权限。您还可以要求强密码实践（如复杂性、重复使用和多重验证 (MFA)），并且您可以使用与现有目录服务的联合。

以下问题重点关注针对安全性的特权管理考虑事项：

章节 3. **您如何保护对 AWS 根账户凭证的访问和使用？**

章节 4. **您如何定义系统用户的角色和责任以控制人员对 AWS 管理控制台和 API 的访问？**

章节 5. **您如何限制对 AWS 资源的自动访问（如通过应用程序、脚本或者第三方工具或服务）？**

章节 6. **您如何管理密钥和凭证？**

保障根账户凭证的安全至关重要，为此，AWS 建议向根账户附加 MFA 并将带 MFA 的凭证锁定在物理上受保护的位置。利用 IAM 服务，您可以创建和管理其他（非根）用户权限并建立对资源的访问级别。

基础设施保护

基础设施保护包含实施最佳实践和履行行业/法规义务所需的控制方法（如深度防御和多重验证）。使用这些方法对于在云中或本地实现成功的持续运营至关重要。

在 AWS 中，您可以使用 AWS 本机技术或使用通过 AWS Marketplace 提供的合作伙伴产品和服务来实施有状态和无状态的数据包检查。您还可以使用 Amazon Virtual Private Cloud (VPC) 创建私有、安全且可扩展的环境，在该环境中，您可定义您的拓扑 — 包括网关、路由表和公有和/或私有子网。

以下问题重点关注针对安全性的基础设施保护考虑事项：

章节 7. **您如何强制实施网络和主机级别的边界保护？**

章节 8. **您如何强制实施 AWS 服务级别保护？**

章节 9. **您如何在 Amazon EC2 实例上保护操作系统的完整性？**

多层防御在任何类型的环境中都适用，并且就基础设施保护而言，很多概念和方法在云和本地模型中都有效。强制实施边界保护、监控入口点和出口点、全面的日志记录、监控和警报对于制定有效的信息安全计划必不可少。

如上面的*设计委托人*部分所述，AWS 客户能够定制或强化 EC2 实例的配置，并能够持久使用此配置以使之成为不可变的 Amazon 系统映像 (AMI)。然后，使用此 AMI 启动的所有新虚拟服务器（实例）（无论由 Auto Scaling 触发还是手动启动）都将获得经过强化的配置。

探测性控制

您可以使用探测性控制来探测或识别安全违规。它们是治理框架的正常部分，可用于支持质量流程、法律合规义务和/或威胁识别和响应工作。探测性控制有多种类型。例如，库存资产及其详细属性将推动更有效的决策制定（和生命周期控制）以帮助建立运营基线。您也可以使用内部审核（对与信息系统相关的控制的检查）来确保实践符合策略和要求，并确保您已基于定义的条件设置正确的自动警报通知。这些控制是重要的反应因素，可帮助组织识别和了解异常活动的范围。

在 AWS 中，以下服务支持探测性控制：

- **AWS CloudTrail —** 一项记录 API 调用的 Web 服务，记录内容包括调用者的身份、调用的时间、源 IP 地址、参数和响应元素。
- **Amazon CloudWatch —** 一项针对 AWS 资源的监控服务，用于记录以下方面：Amazon Elastic Compute Cloud (EC2) CPU、磁盘和网络活动；Amazon Relational Database Service (RDS) 数据库实例；Amazon Elastic Block Store (EBS) 卷等。CloudWatch 提供了对这些指标和其他指标发出警报的功能。
- **AWS Config —** 一项库存和配置历史记录服务，用于提供基础设施中的配置随着时间的推移而变化的相关信息。
- **Amazon Simple Storage Service (S3) —** 通过使用 Amazon S3 数据访问审核，客户可配置 Amazon S3 存储桶来记录访问请求的详细信息，包括类型、资源、日期和时间。
- **Amazon Glacier—** 客户可借助保险库锁功能，利用旨在支持可审核长期保留的合规性控制来保留任务关键数据。

以下问题重点关注针对安全性的探测性控制考虑事项：

章节 10. 您如何捕获和分析 AWS 日志？

出于安全性/取证、法规要求或法律要求等原因，日志管理对于精心构建的设计很重要。AWS 提供了可简化日志管理的实施的功能，方法为使客户能够定义数据保留生命周期或定义将保存、存档和/或最终删除数据的位置。这使可预测和可靠的数据处理更简单且更经济高效。

关键 AWS 服务

对安全性必不可少的 AWS 服务是 AWS Identity and Access Management (IAM)，它使您能够为用户安全地控制对 AWS 服务和资源的访问。以下服务和功能支持安全性的四个方面：

数据保护：Elastic Load Balancing、Amazon Elastic Block Store (EBS)、Amazon Simple Storage Service (S3) 和 Amazon Relational Database Service (RDS) 等服务包含用于保护您的静态数据和传输中的数据的加密功能。AWS Key Management Service (KMS) 使客户能够更轻松地创建和控制用于加密的密钥。

特权管理：IAM 使您能够安全地控制对 AWS 服务和资源的访问。Multi-factor authentication (MFA) 在用户名和密码之外再加了一道安全屏障。

基础设施保护：Amazon Virtual Private Cloud (VPC) 可让您预置 AWS 云（用于在虚拟网络中启动 AWS 资源）的私有部分和隔离部分。

探测性控制：AWS CloudTrail 记录 AWS API 调用，AWS Config 提供您的 AWS 资源和配置的详细库存，Amazon CloudWatch 是针对 AWS 资源的监控服务。

资源

请参阅以下资源以详细了解我们针对安全性的最佳实践。

文档和博客

- AWS 安全中心
- AWS 合规性
- AWS 安全性博客

白皮书

- AWS 安全性概述
- AWS 安全性最佳实践
- AWS 风险和合规性

视频

- AWS 云的安全性
- 责任共担概述

可靠性支柱

可靠性支柱包含系统从基础设施中断或服务中断恢复、动态获取计算资源以满足需求以及减少中断（如错误配置或暂时性网络问题）的能力。

设计原则

在云中，有很多可帮助您提高可靠性的原则：

- **测试恢复过程：**在本地环境中，执行测试通常是为了证明系统在特定情景下可以正常运行；测试通常不用于验证恢复策略。在云中，您可以测试您的系统是如何失败的，并可以验证您的恢复过程。您可以使用自动化来模拟不同的故障或重建之前导致故障的情景。这将揭示您可在故障真正发生之*前*测试和纠正的故障路径，并会降低存在之前未测试到的组件故障情景的风险。
- **自动从故障恢复：**通过监控系统的关键性能指标 (KPI)，您可在超过阈值时触发自动化。这将实现自动通知和故障跟踪，并将执行可绕开或修复故障的自动恢复过程。利用更先进的自动化，用户可在故障发生之前预测并修复它们。

- **水平扩展以提高聚合系统可用性：**将一个大型资源替换为多个小型资源以降低单个故障对整个系统的影响。在多个小型资源中分发请求以确保它们不会共用公共故障点。
- **停止猜测容量：**本地系统中的故障的常见原因是资源饱和，即对系统提出的需求超出了系统的容量（这通常是拒绝服务攻击的目标）。在云中，您可以监控需求和系统利用率，并可以自动添加或删除资源以将其维持在可满足需求但不会预置失衡的最佳水平。

定义

云中的可靠性包括以下三个方面：

1. 基础
2. 变更管理
3. 失败管理

要实现可靠性，系统必须实施妥善规划的基础和监控，并拥有处理需求或要求变化的机制。系统应被设计为可检测失败和自动自行修复。

最佳实践

基础

在构建任何系统之前，应确定影响可靠性的基本要求；例如，您必须为数据中心提供足够的网络带宽。这些要求有时会被忽略（因为它们超出了单个项目的范围）。这种忽略可能严重影响提供可靠的系统的能力。在本地环境中，这些要求可能由于依赖关系而导致前置时间过长，因此必须在最初规划时纳入考虑范围。

在 AWS 中，大多数这些基本要求已纳入考量或可能已按需满足。云在设计上基本是无限的，因此 AWS 负责满足对充足的联网和计算容量的要求，而您可以根据需求随意更改资源大小和分配（如存储设备的大小）。

以下问题重点关注可靠性的基本考虑事项（有关可靠性问题、答案和最佳实践的完整列表，请参阅附录）：

可靠性 1. 如何管理您的账户的 AWS 服务限制？

可靠性 2. 如何在 AWS 上规划您的网络拓扑？

可靠性 3. 您是否有用于处理技术问题的上报途径？

AWS 设置了服务限制（您的团队可请求的每种资源的数量上限）以防止您意外过度预置资源。您需要安排监控和制定流程以监控和更改这些限制，从而满足您的业务需求。由于您采用了云，您可能需要规划与现有本地资源的集成（混合方法）。混合模式支持逐步向一体式的云方法过渡，因此您必须设计您的 AWS 和本地资源作为网络拓扑交互的方式。最后，您需要确保您的 IT 团队接受培训和更新过的流程以支持您的公有云使用，并确保您在适当时拥有合作伙伴或支持协议。

变更管理

了解变更对系统的影响可让您主动进行规划，而监控可让您快速识别可能导致容量问题或 SLA 违规的趋势。在传统环境中，变更控制流程通常是手动实施的，并且必须与审核小心配合以有效控制执行更改的人员和时间。

通过使用 AWS，您可以监控系统行为并自动响应 KPI，例如，在系统获得更多用户时添加更多服务器。您可以控制有权执行系统更改和审核这些更改的历史记录的人员。

以下问题重点关注针对可靠性的变更相关考虑事项：

可靠性 4. 您的系统如何适应需求的变化？

可靠性 5. 您如何监控 AWS 资源？

可靠性 6. 您如何执行变更管理？

如果您将系统构建为可自动添加和删除资源以响应需求变化，则不仅能提高可靠性，还能确保业务成功不会成为负担。实施监控后，如果 KPI 偏离预期标准，则系统会自动通知您的团队。自动记录您的环境的更改可让您审核和快速识别可能已影响可靠性的操作。控制变更管理将确保您可以强制实施提供所需的可靠性的规则。

失败管理

在复杂度合理的任何系统中，可以预料失败将会发生，了解获知这些失败、响应它们并防止它们再次发生的方法通常很有趣。

在 AWS 中，我们可利用自动化来响应监控数据。例如，当特定指标超出阈值时，您可以触发自动操作来纠正问题。此外，您无需尝试诊断并修复属于生产环境一部分的失败资源，而是可以将其替换为新资源并对带外的失败资源执行分析。由于云使您能够以低成本支持整个系统的临时版本，因此您可以使用自动测试来验证完整的恢复过程。

以下问题重点关注针对可靠性的失败管理注意事项：

可靠性 7. 您如何备份数据？

可靠性 8. 您的系统如何承受组件故障？

可靠性 9. 您如何计划恢复？

定期备份您的数据并测试您的备份文件，以确保您可以从逻辑和物理错误恢复。管理失败的一个关键是在失败到恢复的过程中对系统进行频繁的自动测试（最好根据定期计划执行，但也可在发生重大系统变更后触发）。主动跟踪 KPI（如恢复时间目标 (RTO) 和恢复点目标 (RPO)）以评估系统的健康状况（尤其是在失败测试情景下）和帮助您确定和缓解单点故障。这样做的目的是彻底测试您的系统恢复过程，以便让您坚信您可以恢复所有数据并继续为您的客户服务，即使面对持续的问题时也是如此。您的恢复过程也应与正常生产过程一样执行。

关键 AWS 服务

若要确保可靠性，关键的 AWS 服务是 Amazon CloudWatch，该服务可监控运行时指标。支持可靠性的三个方面的其他服务和功能如下所示：

基础：AWS Identity and Access Management (IAM) 使您能够安全地控制对 AWS 服务和资源的访问。通过 Amazon VPC，您可以为 AWS 云预置私有、隔离的部分，并借助该部分在虚拟网络中启动 AWS 资源。

变更管理：AWS CloudTrail 记录了您的账户的 AWS API 调用，并向您传送日志文件以供审核。AWS Config 提供了您的 AWS 资源和配置的详细库存，并持续记录配置更改。

失败管理：AWS CloudFormation 实现了 AWS 资源的模板创建并以有序且可预测的方式预置这些资源。

资源

请参阅以下资源以了解有关与可靠性相关的最佳实践的更多信息。

视频和分析员报告

- 利用故障：错误注入和服务可靠性
- 云中的基准测试可用性和可靠性

文档和博客

- 服务限制文档
- 服务限制报告博客文章

白皮书

- 《使用 AWS 时的备份存档和还原方法》白皮书
- 管理您的大型 AWS 基础设施白皮书
- 《AWS 灾难恢复》白皮书
- 《AWS Amazon VPC 连接性选项》白皮书

AWS Support

- AWS Premium Support
- Trusted Advisor

性能效率支柱

性能效率支柱专注于高效使用计算资源以满足要求，以及在需求变化和技术进步的同时保持这一效率。

设计原则

在云中，有很多可帮助您实现性能效率的原则：

- **将高级技术大众化：**通过将知识和复杂性转移到云供应商的域中，难以实施的技术将变得更容易使用。您的 IT 团队不必了解如何掌握和运用新技术，他们只需将其作为服务使用。例如，NoSQL 数据库、媒体转码和机器学习都是需要专业知识的技术，但专业知识在技术社区内的传播并不均匀。在云中，这些技术将变成一些服务，您的团队可以使用它们，同时将精力放在产品开发而不是资源预置和管理上。

- **数分钟内实现全球化部署：**只需几次单击，即可轻松地在全球多个区域内部署您的系统。这使您能够用最少的成本为您的客户提供更低的延迟和更好的体验。
- **使用*无服务器* 架构：**在云中，无服务器架构使您无需运行和维护服务器来执行传统的计算活动。例如，存储服务可充当静态网站，从而免除了对 Web 服务器的需求；而事件服务可为您托管代码。这不仅消除了管理这些服务器的运营负担，还降低了事务成本，因为这些托管服务以云规模运行。
- **提高实验频率：**借助虚拟和可自动化的资源，您可以使用不同类型的实例、存储或配置快速执行比较测试。

定义

云中的**性能效率**包括以下四个方面：

1. 计算
2. 存储
3. 数据库
4. 空间—时间权衡

这些方面的考虑事项都包括 a) 如何选择最佳方法和资源；b) 考虑到不断发展的云功能，如何让该方法总是保持最新；c) 如何基于预期监控运行时性能；以及 d) 资源如何根据需求来扩展。

最佳实践

计算

特定架构的最佳服务器配置可能因应用程序设计、使用模式和配置设置而异。许多系统对不同的组件使用不同的服务器配置并启用不同的功能来提升性能。为使用案例选择错误的服务器配置可能导致性能效率降低。

在 AWS 中，服务器已经过虚拟化，因此，您可通过按钮单击或 API 调用来更改其功能。由于资源决策不再固定，您可试验不同的服务器类型。在 AWS 中，这些虚拟服务器*实例* 的系列和大小不一，并提供了各种功能（如 SSD 和 GPU）。在 AWS 中，还能执行无服务器计算。例如，AWS Lambda 允许您在不运行实例的情况下执行代码。

以下示例问题侧重于计算注意事项（有关性能效率问题、答案和最佳实践的完整列表，请参阅附录）：

性能 1. **您如何选择适用于系统的实例类型？**

性能 2. **您如何确保在引入新的实例类型和功能时仍拥有最适合的实例类型？**

性能 3. **如何在启动后监控实例以确保其按预期方式执行？**

性能 4. **如何确保实例数满足需求？**

在选择要使用的实例类型时，必须拥有指明匹配*该*工作负载的实例类型（或无服务器方法）的测试数据。这些测试应是可重复的（理想情况下为持续交付 (CD) 管道的一部分），以便您能在新的实例类型或功能可用时轻松地对其进行测试。从操作的角度看，您应实施监控以便在出现任何性能下降时收到通知。

存储

特定系统的最佳存储解决方案因访问方式的类型（数据块、文件或对象）、访问模式（随机或按顺序）、所需的吞吐量、访问频率（在线、离线、存档）、更新频率（缓慢、动态）以及可用性和持久性约束而异。精心设计的系统使用多个存储解决方案并启用不同的功能来提高性能。

在 AWS 中，存储经过虚拟化且可用于许多不同的类型。这样一来，可以更轻松地使存储方法更能满足您的需求，并且还提供难以通过本地基础设施实现的存储选项。例如，Amazon S3 的设计可实现 99.999999999% 的耐用率。您也可以从使用磁性硬盘 (HDD) 更改为使用固态硬盘 (SSD)，并在几秒钟内轻松将虚拟驱动器从一个实例移至另一个实例。

以下示例问题侧重于性能效率的存储注意事项：

性能 5. **如何选择适用于系统的存储解决方案？**

性能 6. **您如何确保在启动新的存储解决方案和功能时仍拥有最适合的存储解决方案？**

性能 7. **如何监控存储解决方案以确保其按预期方式执行？**

性能 8. **如何确保存储解决方案的容量和吞吐量满足需求？**

在选择存储解决方案时，必须拥有指明将交付*该*工作负载所需的成本/价值利润的存储解决方案的测试数据。这些测试应是可重复的（理想情况下为 CD 管道的一部分），以便您能在新的存储解决方案或功能可用时轻松地对其进行测试。用于不同实例的存储类型（EBS 与实例存储，或 HDD 与 SSD）可显著改变您系统的性能效率。从操作的角度看，您应实施监控以便在出现任何性能下降时收到通知。

数据库

最适合特定系统的数据库解决方案会因一致性、可用性、分区容忍性和延迟的要求而异。许多系统对各种子系统使用不同的数据库解决方案，并启用各种功能来提高性能。如果为系统选择的数据库解决方案和功能是错误的，则会导致较低的性能效率。

在 AWS 中，Amazon Relational Database Service (RDS) 提供了一种完全托管的关系数据库。借助 Amazon RDS，您可以扩展您数据库的计算和存储资源，通常不会造成停机。我们还提供了其他数据库和存储解决方案。Amazon DynamoDB 是一种完全托管的 NoSQL 数据库，该数据库提供任何规模的不超过十毫秒的延迟。Amazon Redshift 是一种托管的 PB 级数据仓库，可让您根据性能或容量需求的变化来更改节点的数量或类型。

以下示例问题侧重于性能效率的数据库注意事项：

性能 9. **如何选择适用于系统的数据库解决方案？**

性能 10. **您如何确保在启动新的数据库解决方案和功能时仍拥有最适合的数据库解决方案和功能？**

性能 11. **您如何监控数据库以确保获得预期性能？**

性能 12. **您如何确保数据库的容量和吞吐量满足需求？**

虽然组织的数据库方法（RDBMS、NoSQL 等）对系统的性能效率有很大的影响，但它的选择通常是根据组织的默认设置而不是评估来确定的。在构建和部署数据库解决方案期间，将数据库视为代码，以允许数据库随时间不断演变而不是成为一次性固定决策。使用测试数据标识最匹配每个工作负载的数据库解决方案。这些测试应是可重复的（理想情况下为 CD 管道的一部分），以便您能在新的数据库解决方案或功能可用时轻松地对其进行测试。例如，评估只读副本是否能在不违反其他非功能要求的情况下提高性能效率。从操作的角度看，应实施监控以便在出现任何性能下降时收到通知。

空间和时间权衡

在构建解决方案时，需要在用于缩短处理时间（计算）的空间（内存或存储）或用于减少空间的时间之间进行一系列的权衡。您也可以将资源或缓存数据放置到更接近最终用户的位置来缩短时间。

借助 AWS，您可在几分钟内实现全球化并在全球多个位置部署资源以更接近最终用户。您还可以向信息存储动态添加只读副本（例如数据库系统）来减少主数据库上的负载。

使用 AWS 的全球基础设施实现更低的延迟性和更高的吞吐量，并确保数据仅驻留在指定的区域。联网解决方案（例如 AWS Direct Connect）旨在提供您的本地网络和 AWS 基础设施之间的可预见的延迟。AWS 还提供了缓存解决方案，例如 Amazon ElastiCache 和 Amazon CloudFront，前者可帮助提高效率，后者可缓存更接近最终用户的静态内容的副本。

以下示例问题侧重于性能效率的空间和时间权衡：

性能 13. 如何选择适用于系统的邻近度和缓存解决方案？

性能 14. 您如何确保在启动新的解决方案时仍拥有最适合的邻近度和缓存解决方案？

性能 15. 您如何监控邻近度和缓存解决方案以确保获得预期性能？

性能 16. 您如何确保您拥有的邻近度和缓存解决方案满足需求？

在选择要使用的实例类型时，必须拥有指明最匹配*该*工作负载的实例类型（或无服务器方法）的测试数据。这些测试应是可重复的（理想情况下为 CD 管道的一部分），以便您能在新的方法或功能可用时轻松地对其进行测试。例如，测试以了解使用 Amazon ElastiCache 作为旁置写入缓存是否能在不违反其他非功能要求的情况下提高性能效率。从操作的角度看，您应实施监控以便在出现任何性能下降时收到通知。架构应按需求扩展并留有余地。

关键 AWS 服务

针对性能效率的关键 AWS 服务为 Amazon CloudWatch，此服务将监控您的资源和系统，从而提供总体性能和运行状况的可见性。对于性能效率的四个方面，以下服务很重要：

计算：Auto Scaling 是确保您拥有足量实例来满足需求并保持响应能力的关键。

存储：Amazon EBS 提供了允许您针对使用案例进行优化的广泛存储选项（例如 SSD 和 PIOPS）。Amazon S3 提供了低冗余存储、针对 Amazon Glacier（存档存储）的生命周期策略和无服务器内容分发。

数据库：Amazon RDS 提供了允许您针对使用案例进行优化的广泛数据库功能（例如预置的 IOPS 和只读副本）。Amazon DynamoDB 提供任意规模的不超过十毫秒的延迟。

空间和时间权衡：AWS 的区域遍布全球，使您能够为资源、数据和处理选择最佳位置。使用 Amazon CloudFront 缓存更接近用户的内容。

资源

请参阅以下资源以了解有关与性能效率相关的最佳实践的更多信息。

视频

- 性能渠道
- AWS 上的性能基准测试

文档

- Amazon S3 性能优化文档
- Amazon EBS 卷性能文档

成本优化支柱

使用**成本优化**支柱评估您的避免或消除不必要的成本或非最优资源的能力，以及将节省的成本和资源应用于业务的差异化优势上的能力。成本优化的系统可使您支付的价格尽可能达到最低，并且仍将实现您的商业目标并达到或超出其他精心设计的支柱的关键要求。您可使用技术选择适当的架构、减少未使用的资源和选择最经济的方法，从而实现成本优化。

设计原则

在云中，您可遵循大量可帮助您实现成本优化的原则：

- **以透明方式确定支出的归属：**借助云，可轻松标识系统成本并将 IT 成本归属于各个业务所有者。这有助于标识投资回报，从而激励这些所有者优化其资源并降低成本。
- **使用托管服务降低拥有成本：**在云中，托管服务移去维护用来执行任务（例如，发送电子邮件或管理数据库）的服务器的运行负担。此外，由于托管服务以云规模运行，因此它们可提供每事务或服务的更低成本。
- **以资本费用换取运营费用：**您只需为自己所使用的计算资源付费，而不是在知道如何使用数据中心和服务器之前就对其大量投资。例如，在工作周中，开发和测试环境通常每天仅运行 8 小时，您可以在这些资源未被使用时将其停止，从而实现 75% 的潜在成本节省（40 小时和 168 小时）。
- **从规模经济中受益：**由于 AWS 可实现更佳的规模经济，因此您可使用云计算来实现自行无法实现的更低的可变成本。几十万个客户在 AWS 云中聚集，这将转化为更低的即用即付价格。
- **停止在数据中心运营上投入资金：**有了 AWS，您无需忙于搬动沉重的机架、堆栈和电源服务器，而能够专注于客户和业务项目而不是 IT 基础设施。

定义

云中的**成本优化**包含四个方面：

1. 已匹配的供需
2. 经济高效的资源
3. 支出感知
4. 随时间进行优化

与其他支柱一样，需要考虑权衡，例如，是否针对上市速度或成本进行优化。在某些情况下，最好是针对速度进行优化 — 快速上市、推出新功能或按时完成任务 — 而不是在前期成本优化方面投资。由于始终存在为了“预防万一”而过度补偿（而不是花时间进行基准测试以实现成本最优部署）的诱惑，因此设计决策有时以速度而不是经验数据为导向。这通常会带来显著过度预置的和有待优化的部署。以下部分提供了针对部署的初始和持续成本优化的技术和战略指导。

最佳实践

已匹配的供需

以最合理的方式匹配供需可交付最低系统成本，但仍需要足够的额外供应来允许预置时间和单个资源故障。需求可以是固定或可变的，这需要指标和自动化来确保管理不会产生大量成本。

在 AWS 中，您可以自动预置资源来满足需求。Auto Scaling 和基于时间的、事件驱动型的、基于队列的方法使您能够根据需要添加或删除资源。如果您可以预计需求的变化，则可节省更多成本并确保资源满足系统需求。

以下示例问题侧重于针对成本优化的已匹配的供需（有关成本优化问题、答案和最佳实践的完整列表，请参阅附录）：

成本 1. 您如何确保您的容量匹配而不会大幅度超出所需容量？

成本 2. 您如何优化 AWS 服务的使用率？

监控工具和定期基准测试可帮助您实现更大的资源利用率。按需计算、Auto Scaling 和其他自动部署机制的灵活性促进了更大程度的优化，确保您仅预置所需的资源并且能够水平扩展。

经济高效的资源

使用适用于系统的实例和资源是实现成本节省的关键。例如，在小型服务器上，报告流程可能需要耗费 5 个小时才能完成；在大型服务器上，报告流程可在 1 个小时内完成，但费用是前者的 2 倍。通过这两种方式完成该流程所获得的结果相同，但小型服务器会随时间的推移产生更多成本。

精心设计的系统将使用最经济高效的资源，从而产生显著、积极的经济影响。您还有机会使用托管服务来降低成本。例如，您可以使用按消息收费的服务，而不是维护服务器来交付电子邮件。

AWS 提供了各种灵活的、经济实用的定价选项，使您能够以最能满足您的需求的方式获取 Amazon EC2 实例。*按需实例*可让您按小时支付计算容量费用，无需承诺最低消费。*预留实例* (RI) 可让您预留容量并节省最多 75% 的按需定价成本。利用*竞价型实例*，您可以在对未使用的 Amazon EC2 容量出价时享受大幅度的折扣。在系统可容忍使用一组可动态进出的服务器的情况（例如，在使用 HPC 和大数据时）下，竞价型实例将适用。

以下示例问题侧重于选择针对成本优化的经济高效的资源：

成本 3. **您是否已选择适当的资源类型来满足您的成本目标？**

成本 4. **您是否已选择适当的定价模式来满足您的成本目标？**

成本 5. **是否有可用来提高 ROI 的托管服务（级别高于 Amazon EC2、Amazon EBS 和 Amazon S3 的服务）？**

通过使用 AWS Trusted Advisor 等工具定期检查 AWS 使用率，您可以主动监控利用率并相应地调整部署。您也可以利用托管的 AWS 服务（例如，Amazon RDS、Amazon Elastic MapReduce (EMR) 和 Amazon DynamoDB），这可降低每项目成本和管理成本。考虑 Amazon CloudFront 等 CDN 解决方案来潜在地降低与网络流量关联的成本。

支出感知

云支持的更大的灵活性和敏捷性鼓励创新和快速开发和部署。它消除了与预置本地基础设施关联的手动流程和时间，其中包括标识硬件规格、协商报价、管理采购订单、安排发货和部署资源。但是，易用性和几乎不受限制的按需容量可能需要通过一种新的方式来考虑支出。

许多企业都包含由不同的团队运营的多个系统。将资源成本归属于单个业务或产品所有者的能力可推动高效的使用行为并帮助减少浪费。此外，准确的成本归属使您能够理解哪些产品是真正有利可图的，并使您能够作出有关预算分配的更明智的决策。

以下示例问题侧重于针对成本优化的支出感知：

成本 6. 您现在使用哪种访问控制和程序来管理 AWS 成本？

成本 7. 您如何监控使用率和开支？

成本 8. 您如何停用不再需要的资源或停止暂时不需要的资源？

成本 9. 您如何在设计架构时考虑数据传输费用？

您可以使用成本分配标签对 AWS 成本进行分类和跟踪。当您对 AWS 资源（例如，Amazon EC2 实例或 Amazon S3 存储桶）应用标签时，AWS 会生成成本分配报告，其中按标签汇总了您的使用率和成本。您可以应用代表业务类别（例如成本中心、系统名称或所有者）的标签，以便整理多种服务的成本。

利用针对已标记资源的成本可见性，可以更轻松地标识不再创造业务价值且应停用的孤立的资源或项目。您可以设置账单提醒来通知您预测超支，并且可使用 AWS 简单月度成本结算器来计算数据传输成本。

随时间进行优化

在 AWS 发布新的服务和功能时，最佳实践是重新评估您的现有架构决策以确保其仍是最经济高效的决策。随着需求的变化，要积极地停用不再需要的资源和整个服务或系统。

来自 AWS 的托管服务通常可大大优化解决方案，因此在发布新的托管服务后了解这些服务是不错的做法。例如，在 Amazon EC2 上运行 Amazon RDS 数据库比运行您自己的数据库的成本更低。

以下示例问题侧重于针对成本优化的成本重新评估：

成本 10. 您如何管理和/或考虑新服务的采用？

通过定期重新评估部署，通常可利用新的 AWS 服务来降低成本。此外，评估更新的服务的适用性可帮助您节省成本；例如，AWS RDS for Aurora 可帮助您降低关系数据库的成本。

关键 AWS 服务

支持成本优化的关键 AWS 服务是成本分配标签，可帮助您了解系统的成本。对于成本优化的四个方面，以下服务和功能很重要：

已匹配的供需：Auto Scaling 使您能够在不超支的情况下添加或删除资源以满足需求。

经济高效的资源：您可以使用预留实例和预付容量来降低您的成本。AWS Trusted Advisor 可用于检查您的 AWS 环境，并找到节省成本的机会。

支出感知：Amazon CloudWatch 警报和 Amazon Simple Notification Service (SNS) 通知将在您已超过或预计将超出预算金额时提醒您。

随时间进行优化：AWS 网站上的 AWS 博客和*新增功能* 部分是了解有关新发布的功能和服务的资源。AWS Trusted Advisor 通过消除未使用的/闲置的资源或通过提供预留实例容量来检查您的 AWS 环境并找出可节省成本的机会。

资源

请参阅以下资源以了解有关成本优化的 AWS 最佳实践的更多信息。

视频

- 基于 AWS 的成本优化

文档

- AWS 成本中心

工具

- AWS 总拥有成本 (TCO) 计算器
- AWS 详细账单报告
- AWS 简单月度成本结算器
- AWS 成本管理器

结论

AWS 精心设计的框架提供了跨在云中设计可靠的、安全的、高效的、经济实用的系统的四个支柱的架构最佳实践。该框架提供了一组可让您评估现有的或计划的架构的问题和一组针对每个支柱的 AWS 最佳实践。在您的架构中使用该框架将帮助您构建稳定而高效的系统，这使您能够专注于您的功能需求。

贡献者

以下为对此文档有贡献的个人和组织：

- Philip Fitzsimons，Amazon Web Services 的解决方案架构经理
- Erin Rifkin，Amazon Web Services 的高级项目经理
- Callum Hughes，Amazon Web Services 的解决方案架构师
- Max Ramsay，Amazon Web Services 的首席安全解决方案架构师
- Scott Paddock，Amazon Web Services 的安全解决方案架构师

文档历史记录

2015 年 11 月 20 日。使用最新的 Amazon CloudWatch 日志信息更新了附录。

附录：精心设计的问题、答案和最佳实践

此附录包含精心设计的问题和答案的完整列表，其中包括按支柱组织的最佳实践：

安全支柱

章节 1. 您如何加密和保护静态数据？

传统的安全控制是加密静态数据。AWS 通过使用客户端（例如，SDK 支持的、OS 支持的、Windows Bitlocker、dm-crypt、Trend Micro SafeNet 等）和服务器端（例如 Amazon S3）来支持这一点。您也可以使用服务器端加密 (SSE) 和 Amazon Elastic Block Store 加密卷等。

最佳实践：

- 使用 AWS 服务特定控制（例如，Amazon S3 SSE、Amazon EBS 加密卷、Amazon Relational Database Service (RDS)、透明数据加密 (TDE) 等）对静态数据进行加密。
- 使用客户端技术对静态数据进行加密。
- 来自 AWS Marketplace 或 APN 合作伙伴的解决方案。

章节 2. 您如何加密和保护传输中的数据？

最佳实践是使用加密来保护传输中的数据。AWS 支持使用服务 API 的加密终端节点。此外，客户可在其 Amazon EC2 实例内使用各种技术。

最佳实践：

- 适当地使用支持 SSL 的 AWS API。
- 使用 SSL 或等效项进行通信。
- 基于 VPN 的解决方案。
- 私有连接（例如 AWS Direct Connect）。
- 正在使用 AWS Marketplace 解决方案。

章节 3. 您如何保护对 AWS 根账户凭证的访问和使用？

AWS 根账户凭证与其他操作系统中的根账户或本地管理员账户凭证类似，使用时应非常谨慎。当前的最佳实践是创建 AWS Identity and Access Management (IAM) 用户、将这些用户关联到管理员组并使用 IAM 用户来管理账户。AWS 根账户不应拥有 API 密钥，而应拥有强密码并且应与硬件多重验证 (MFA) 设备关联；这将强制仅能通过 AWS 管理控制台使用根标识，并且不允许将其用于应用程序编程接口 (API) 调用。请注意，一些经销商或区域不分发或支持 AWS 根账户凭证。

最佳实践：

- AWS 根账户凭证仅用于最少的所需活动。
- 存在与 AWS 根账户关联的 MFA 硬件设备。
- 正在使用 AWS Marketplace 解决方案。

章节 4. 您如何定义系统用户的角色和责任以控制用户对 AWS 管理控制台和 API 的访问？

当前的最佳实践是客户通过创建用户组来分离系统用户的已定义的角色和责任。可使用多种不同的方法来定义用户组：通过安全断言标记语言 (SAML) 集成（例如，定义 Active Directory 中的角色）或通过使用通常通过 SAML 或 AWS Security Token Service (STS) 集成的第三方解决方案（例如，Okta、Ping Identity 或另一种自定义技术）定义 Identity and Access Management (IAM) 组、用于跨账户访问的 IAM 角色和 Web 身份。强烈反对使用共享账户。

最佳实践：

- IAM 用户和组
- SAML 集成
- Web 联合身份
- AWS Security Token Service (STS)
- 用于跨账户访问的 IAM 角色
- 来自 AWS Marketplace 的解决方案（例如，Okta、Ping Identity 等）或来自 APN 合作伙伴的解决方案
- 已定义和实施员工生命周期策略
- 已清楚定义用户、组和角色并仅向其授予达到业务要求所需的最小权限

章节 5. 您如何限制对 AWS 资源的自动访问？（例如，应用程序、脚本和/或第三方工具或服务）

应以与为用户创建用户组类似的方式定义系统化访问。对于 Amazon EC2 实例，这些组称为 EC2 的 IAM 角色。当前的最佳实践是使用 EC2 的 IAM 角色和 AWS 软件开发工具包或 CLI（拥有对检索 EC2 的 IAM 角色凭证的内置支持）。传统上，用户凭证将注入 EC2 实例中，但强烈反对将凭证硬编码为脚本和源代码。

最佳实践：

- Amazon EC2 的 IAM 角色
- 使用 IAM 用户凭证，但不将其硬编码到脚本和应用程序中
- SAML 集成
- AWS Security Token Service (STS)
- 特定于操作系统的控制用于 EC2 实例
- 正在使用 AWS Marketplace 解决方案

章节 6. 您如何管理密钥和凭证？

密钥和凭证是应保护的机密信息，并且应定义和使用适当的轮换策略。最佳实践是不将这些机密信息硬编码到管理脚本和应用程序中，但这种情况确实经常出现。

最佳实践：

- 正在使用适当的密钥和凭证轮换策略。
- 使用 AWS CloudHSM。
- AWS 服务器端技术与 AWS 托管密钥（例如，Amazon S3 SSE、Amazon EBS 加密卷等）结合使用。
- AWS Marketplace 解决方案（例如，SafeNet、TrendMicro 等）。

章节 7. 您如何强制实施网络和主机级别的边界保护？

在本地数据中心中，DMZ 使用防火墙将单独的系统引入可信的和不可信的区域。在 AWS 上，使用有状态和无状态的防火墙。有状态的防火墙称为安全组，无状态的防火墙称为网络访问控制列表 (ACL)（可保护 Amazon Virtual Private Cloud (VPC) 中的子网）。当前的最佳实践是在 VPC 中运行系统，并在安全组中定义基于角色的安全性（例如，Web 层和应用层等）以及网络 ACL 中的基于位置的安全性（例如，每个可用区一个子网内的 Elastic Load Balancing 层、每个可用区中另一个子网内的 Web 层等）。

最佳实践：

- 使用具有最小授权的安全组强制执行基于角色的访问。
- 系统在一个或多个 VPC 中运行。
- 通过私有机制（例如，虚拟专用网络 (VPN)、IPsec 隧道、AWS Direct Connect、AWS Marketplace 解决方案等）实现可信 VPC 访问。
- 适当地使用子网和网络 ACL。
- 使用具有最小授权的基于主机的防火墙。
- 使用特定于服务的访问控制（例如，存储桶策略）。
- 使用到 VPC 的私有连接（例如，VPN、AWS Direct Connect、VPC 对等，等）
- 使用堡垒主机技术管理实例。
- 定期执行安全测试。
- 定期审核 AWS Trusted Advisor 检查。

章节 8. 您如何强制实施 AWS 服务级别保护？

另一个最佳实践是控制对资源的访问。AWS Identity and Access Management (IAM) 允许定义各种资源级控制（例如，加密的使用、时间、源 IP 等），并且各种服务允许使用其他技术（例如，Amazon S3 存储桶策略等）。此外，客户可在其 Amazon EC2 实例内使用各种技术。

最佳实践：

- 使用最小特权配置的凭证。
- 责任分离。
- 定期审核权限。
- 为敏感 API 调用定义资源要求，例如，要求 MFA 身份验证和加密。

- 定义和使用特定于服务的要求。
- 正在使用 AWS Marketplace 解决方案。

章节 9. 您如何在 Amazon EC2 实例上保护操作系统的完整性？

另一个传统控制是保护操作系统的完整性。可使用传统的基于主机的技术（例如，OSSEC、Tripwire、Trend Micro 深度安全防护等）在 EC2 中轻松实现此操作。

最佳实践：

- 文件完整性控制用于 EC2 实例。
- 基于主机的入侵检测控制用于 EC2 实例。
- 使用来自 AWS Marketplace 或 APN 合作伙伴的解决方案。
- 使用默认情况下受保护的自定义 AMI 或配置管理控制（例如，Puppet 或 Chef）。

章节 10. 您如何捕获和分析 AWS 日志？

捕获日志对于调查从性能到安全事故的一切来说都是至关重要的。当前的最佳实践是定期将日志从源直接移至日志处理系统（例如，CloudWatch Logs、Splunk、Papertrail 等）或存储在 Amazon S3 存储桶中以便将来根据业务需求进行处理。常见的日志源为 AWS API 和与用户相关的日志（例如，AWS CloudTrail）、特定于 AWS 服务的日志（例如，Amazon S3、Amazon CloudFront 等）、操作系统生成的日志以及特定于第三方应用程序的日志。您可以使用 Amazon CloudWatch Logs 监控、存储和访问来自 Amazon EC2 实例、AWS CloudTrail 或其他来源的日志文件。

最佳实践：

- AWS CloudTrail。
- Amazon CloudWatch 日志。
- Elastic Load Balancing (ELB) 日志。
- Amazon Virtual Private Cloud (VPC) 筛选日志。
- Amazon S3 存储桶日志。
- 其他特定于 AWS 服务的日志源。
- 操作系统或第三方应用程序日志。
- 正在使用 AWS Marketplace 解决方案。

可靠性支柱

可靠性 1. 如何管理您的账户的 AWS 服务限制？

使用默认服务限制配置 AWS 账户以防止新用户预置的资源意外超出其所需的资源。AWS 客户应评估其 AWS 服务需求并为使用的每个区域请求适当的限制更改。

最佳实践：

- **监控和管理限制** 评估 AWS 上的潜在使用率，适当地增大区域限制并允许计划的使用率增长。
- **设置自动化监控** 实施工具（例如，软件开发工具包）以便在接近阈值时向您发送提醒。
- **了解固定的服务限制** 了解不可更改的服务限制以及围绕这些限制实现的构建。

可靠性 2. 如何在 AWS 上规划您的网络拓扑？

应用程序可存在于一个或多个环境：EC2 Classic 或 VPC（默认情况下为 VPC）。网络注意事项（例如，系统连接、EIP/公有 IP 地址管理、VPC/私有地址管理和名称解析）是在云中使用资源的基础。精心设计和制定的部署对于降低重叠和争用的风险来说是必不可少的。

最佳实践：

- **AWS 的高度可用连接** 多个 DX 线路、多个 VPN 隧道、AWS Marketplace 设备。
- **系统的高度可用连接** 高度可用的负载均衡和/或代理、基于 DNS 的解决方案、AWS Marketplace 设备等。
- **非重叠私有 IP 范围** 在您的 Virtual Private Cloud 中使用的 IP 地址范围和子网不应相互重叠、与其他云环境重叠或与本地环境重叠。
- **IP 子网分配** 单个 Amazon VPC IP 地址范围应足够大以满足应用程序的要求，包括在未来扩展中进行因子分解和跨可用区将 IP 地址分配到子网。

可靠性 3. 您是否有用于处理技术问题的上报途径？

客户应利用 AWS Support 或 AWS 合作伙伴。定期交互将帮助解决和防止已知问题、知识差距和设计问题。这将降低实施故障和大规模中断的风险。

最佳实践：

- **已计划** 与 AWS Support 或 APN 合作伙伴建立持续合作关系。
- **利用 AWS Support API** 将 AWS Support API 与您的内部监控和服务单系统集成。

可靠性 4. 您的系统如何适应需求的变化？

可扩展的系统可为自动添加和删除资源带来灵活性，以便这些资源能够在任意给定时间点高度适应当前需求。

最佳实践：

- **自动扩展** 使用自动可扩展服务，例如：Amazon S3、Amazon CloudFront、Auto Scaling、Amazon DynamoDB、AWS Elastic Beanstalk 等。
- **负载测试** 采用负载测试方法来度量扩展活动是否将满足应用程序要求。

可靠性 5. 您如何监控 AWS 资源？

日志与指标是一种功能强大的工具，用于深入了解您的应用程序的运行状况。您可以将系统配置为监控日志与指标并在超过阈值或发生重大事件时发送通知。理想情况下，在超过低性能阈值或发生故障时，系统应已设计为自动自我修复或扩展来进行响应。

最佳实践：

- **监控** 利用 Amazon CloudWatch 或第三方工具监控您的应用程序。
- **通知** 计划在发生重大事件时接收通知。
- **自动响应** 使用自动化以在检测到故障时采取措施，例如更换发生故障的组件。
- **审核** 基于重大事件经常地审核系统以评估架构。

可靠性 6. 您如何执行变更管理？

必须实施预置的 AWS 资源和应用程序的变更管理，以确保应用程序和操作环境正在运行已知的软件，并且可通过受控方式进行修补或替换。

最佳实践：

- **CM 自动化** 自动部署/修补。

可靠性 7. 您如何备份数据？

备份数据、应用程序和操作环境（定义为利用应用程序配置的操作系统）以满足平均恢复时间 (MTTR) 和恢复点目标 (RPO) 的要求。

最佳实践：

- **数据已备份** 使用 Amazon S3、Amazon EBS 快照或第三方软件对重要数据进行备份以满足 RPO。
- **自动备份** 使用 AWS 功能、AWS Marketplace 解决方案或第三方软件自动备份。
- **保护和/或加密备份** 参阅“AWS 安全最佳实践”白皮书。
- **定期恢复测试** 通过恢复测试验证备份过程实施是否满足 RTO 和 RPO。

可靠性 8. 您的系统如何承受组件故障？

您的应用程序是否有高可用性和较短平均恢复时间 (MTTR) 方面的要求（隐式或显式）？如果有的话，请构建您的应用程序以实现弹性，并分配这些应用程序以承受中断。要实现较高级别的可用性，此分配应跨不同的物理位置。构建单个层（例如 Web 服务器、数据库）以实现弹性，其中包括监控、自我修复和重大事件中断和故障的通知。

最佳实践：

- **负载均衡** 使用资源池前面的负载均衡器。
- **多可用区/区域** 跨多个可用区/区域分发应用程序。
- **自动修复** 使用自动功能检测故障并执行修复操作。
- **监控** 持续监控系统的运行状况。

- **通知** 计划接收任何重大事件的通知。

可靠性 9. 您如何计划恢复？

如果备份方法需要还原数据，则数据恢复很重要。您对此数据的目标、资源、位置和功能的定义和执行必须符合 RTO 和 RPO 目标。

最佳实践：

- **定义的目标** 定义 RTO 和 RPO。
- **灾难恢复** 制定 DR 策略。
- **配置偏差** 确保 Amazon 系统映像 (AMI) 和系统配置状态在 DR 站点/区域保持最新。
- **服务限制** 请求提高 DR 站点的服务限制以适应故障转移。
- **已测试和验证 DR** 定期测试针对 DR 的故障转移以确保满足 RTO 和 RPO。
- **已实施自动恢复** 使用 AWS 和/或第三方工具自动实施系统恢复。

性能支柱

性能 1. 您如何选择适用于系统的实例类型？

Amazon EC2 提供多种经过优化、适用于不同使用案例的实例类型以供选择。实例类型由 CPU、内存、存储和网络容量组成不同的组合，可让您灵活地为您的应用程序选择适当的资源组合。每种实例类型都包括一种或多种实例大小，从而使您能够将您的资源扩展到符合目标工作负载的要求。AWS 支持无服务器架构（如 AWS Lambda），这可彻底改变工作负载的运行效率。

最佳实践：

- **策略/参考架构** 按照内部管理标准基于预测的资源需求来选择实例类型和大小。
- **成本/预算** 在内部成本控制下基于预测的资源需求选择实例类型和大小。
- **基准测试** 对 AWS 上的已知工作负载进行负载测试，并根据测试结果来估计最佳选择 — 测试已知性能基准与已知工作负载。

- **来自 AWS 或 AWS 合作伙伴网络 (APN) 成员的指导** 基于最佳实践建议进行选择。
- **负载测试** 使用不同的实例类型和大小在 AWS 上部署最新版本的系统，使用监控来捕获性能指标，然后基于性能/成本计算进行选择。

性能 2. 您如何确保在引入新的实例类型和功能时仍拥有最适合的实例类型？

AWS 倾听客户反馈并且仍将利用新的实例类型和大小进行创新，从而提供新的 CPU、内存、存储和网络容量组合。这意味着，可能会发布新的实例类型，从而提供比最初选择的实例类型更高的性能效率。

最佳实践：

- **审核** 基于预测的资源需求周期性地重新选择新的实例类型。
- **基准测试** 在发布每种新实例类型后，对 AWS 上的已知工作负载进行负载测试，并根据测试结果来估计最佳选择。
- **负载测试** 在发布每种新的实例类型后，在 AWS 上部署最新版本的系统，使用监控来捕获性能指标，然后基于性能/成本计算进行选择。

性能 3. 如何在启动后监控实例以确保其按预期方式执行？

系统性能可能会因内部和/或外部因素随时间而下降。通过监控系统的性能，您可以标识此下降情况并修复内部或外部因素（如操作系统或应用程序负载）。

最佳实践：

- **Amazon CloudWatch 监控** 使用 CloudWatch 监控实例。
- **第三方监控** 使用第三方工具监控系统。
- **定期审核** 定期审核您的监控仪表板。
- **基于警报的通知** 在指标超出安全范围时接收来自监控系统的自动警报。
- **基于触发器的操作** 警报会触发修复或升级问题的自动操作。

性能 4. 如何确保实例数满足需求？

对系统施加的需求量通常会因周期而异：产品生命周期（例如启动或增长）；暂时的周期（例如一天中的某个时间、一周中的某天或某个月）；不可预测的周期（例如社交媒体可见性）；及可预测的周期（例如电视剧集）。因实例不足而无法满足您的工作负载会降低用户体验，最糟糕的情况是导致系统故障。

最佳实践：

- **已计划** 基于指标和/或已计划的事件进行计划。
- **自动编写脚本** 使用工具来实现自动管理。
- **自动 — Auto Scaling** 使用 Auto Scaling 实现自动管理。

性能 5. 如何选择适用于系统的存储解决方案？

AWS 旨在提供具有高持久性和可用性的低成本数据存储。AWS 提供用于备份、存档和灾难恢复的存储选项以及数据块、文件和对象存储。

最佳实践：

- **策略/参考架构** 按照内部管理标准基于预测的资源需求来选择存储解决方案和功能。
- **成本/预算** 在内部成本控制下基于预测的资源需求来选择存储解决方案和功能。
- **基准测试** 对 AWS 上的已知工作负载进行负载测试，并根据测试结果来估计最佳选择 — 测试已知性能基准与已知工作负载。
- **来自 AWS 或 APN 合作伙伴的指导** 基于最佳实践建议选择解决方案。
- **负载测试** 使用不同的存储解决方案在 AWS 上部署最新版本的系统；使用监控来捕获性能指标；然后基于性能/成本计算进行选择。

性能 6. 您如何确保在启动新的存储解决方案和功能时仍拥有最适合的存储解决方案？

AWS 倾听客户反馈并且仍将利用新的存储解决方案和功能进行创新，从而提供新的容量、吞吐量和持久性组合。这意味着，可能会发布新的存储解决方案，从而提供比最初选择的存储解决方案更高的性能效率。

最佳实践：

- **审核** 基于预测的资源需求周期性地重新选择新的存储解决方案和功能。
- **基准测试** 在发布每种新的存储解决方案和功能后，在 AWS 上对已知工作负载进行负载测试，并根据测试结果来估计最佳选择。
- **负载测试** 在发布每种新的存储解决方案后，在 AWS 上部署最新版本的系统，使用监控来捕获性能指标，然后基于性能/成本计算进行选择。

性能 7. 如何监控存储解决方案以确保其按预期方式执行？

系统性能可能会因内部和/或外部因素随时间而下降或在一段时间内下降。通过监控系统的性能，您可以标识此下降情况并修复内部或外部因素。

最佳实践：

- **Amazon CloudWatch 监控** 使用 CloudWatch 监控存储系统。
- **第三方监控** 使用第三方工具监控存储系统。
- **定期审核** 定期审核您的监控仪表板。
- **基于警报的审核** 规划您的监控系统，使其在指标超出安全范围时自动向您发送提醒。
- **基于触发器的操作** 规划警报以触发修复或升级问题的自动操作。

性能 8. 如何确保存储解决方案的容量和吞吐量满足需求？

对系统施加的需求量通常会因周期而异：产品生命周期（例如启动或增长）；暂时的周期（例如一天中的某个时间、一周中的某天或某个月）；不可预测的周期（例如社交媒体可见性）；及可预测的周期（例如电视剧集）。因存储容量或吞吐量不足而无法满足您的工作负载会降低用户体验，最糟糕的情况是导致系统故障。

最佳实践：

- **反应式** 基于指标进行手动管理。
- **已计划** 基于指标和/或已计划的事件来计划将来的容量和吞吐量。
- **自动** 针对指标实现自动化。

性能 9. 如何选择适用于系统的数据库解决方案？

最适合特定系统的数据库解决方案会因一致性、可用性、分区容忍性和延迟的要求而异。许多系统对不同的子系统使用不同的数据库解决方案，并启用各种功能来提高性能。如果为系统工作负载选择的数据库解决方案和功能是错误的，则会导致较低的性能效率。

最佳实践：

- **策略/参考架构** 按照内部管理标准基于预测的资源需求来选择数据库解决方案和功能。
- **成本/预算** 在内部成本控制下基于预测的资源需求选择数据库解决方案和功能。
- **基准测试** 对 AWS 上的已知工作负载进行负载测试，并根据测试结果来估计最佳选择 - 测试已知性能基准与已知工作负载。
- **来自 AWS 或 APN 合作伙伴的指导** 基于最佳实践建议选择解决方案。
- **负载测试** 使用不同的数据库解决方案和功能在 AWS 上部署最新版本的系统，使用监控来捕获性能指标，然后基于性能/成本计算进行选择。

性能 10. 您如何确保在启动新的数据库解决方案和功能时仍拥有最适合的数据库解决方案和功能？

AWS 倾听客户反馈并且仍将利用新的数据库解决方案和功能进行创新，从而提供新的一致性、可用性、分区容忍性和延迟组合。这意味着，可能会发布新的数据库解决方案或功能，从而提供比最初选择的存储解决方案或功能更高的性能效率。

最佳实践：

- **审核** 基于预测的资源需求周期性地重新选择数据库解决方案和功能。
- **基准测试** 在发布每种新的数据库解决方案或功能后，在 AWS 上对已知工作负载进行负载测试，并根据测试结果来估计最佳选择。
- **负载测试** 在发布每种新的数据库解决方案或功能后，在 AWS 上部署最新版本的系统，使用监控来捕获性能指标，然后基于性能/成本计算进行选择。

性能 11. 您如何监控数据库以确保获得预期性能？

系统性能可能会因内部或外部因素随时间而下降。通过监控系统的性能，您可以标识此下降情况并修复内部或外部因素。

最佳实践：

- **Amazon CloudWatch 监控** 使用 CloudWatch 监控数据库。
- **第三方监控** 使用第三方工具监控数据库。
- **定期审核** 定期审核您的监控仪表板。
- **基于警报的通知** 实施规划，使您的监控系统在指标超出安全范围时自动向您发送提醒。
- **基于触发器的操作** 实施规划，使警报触发修复或升级问题的自动操作。

性能 12. 您如何确保数据库的容量和吞吐量满足需求？

对系统施加的需求量通常会因周期而异：产品生命周期（例如启动、增长等）；暂时的周期（例如一天中的某个时间、工作日或某个月等）；不可预测的周期（例如社交媒体可见性）；以及可预测的周期（例如电视剧集）。因数据库容量和吞吐量不足而无法满足工作负载会降低用户体验，最糟糕的情况是导致系统故障。

最佳实践：

- **已计划** 基于指标和/或已计划的事件来规划将来的容量和吞吐量。
- **自动** 针对指标实现自动化。

性能 13. 如何选择适用于系统的邻近度和缓存解决方案？

物理距离、网络距离或长时间运行的请求会导致系统延迟。延迟未得到解决会致使占用系统资源的时间比所需时间更长，并且会导致内部和外部性能下降。要减少延迟，请从最终用户的角度考虑整个系统的端到端性能，并寻求机会来调整资源或缓存解决方案的物理邻近度。

最佳实践：

- **策略/参考架构** 按照内部管理标准基于预测的资源需求来选择邻近度和缓存解决方案。
- **成本/预算** 在内部成本控制下基于预测的资源需求来选择邻近度和缓存解决方案。
- **基准测试** 对 AWS 上的已知工作负载进行负载测试，并根据测试结果来估计最佳选择；测试已知性能基准与已知工作负载。
- **来自 AWS 或 APN 合作伙伴的指导** 基于最佳实践建议选择邻近度和缓存解决方案。
- **负载测试** 使用不同的邻近度和缓存解决方案在 AWS 上部署最新版本的系统，使用监控来捕获性能指标，然后基于性能/成本计算进行选择。

性能 14. 您如何确保在启动新的解决方案时仍拥有最适合的邻近度和缓存解决方案？

AWS 倾听客户反馈并且仍将利用新的邻近度和缓存解决方案进行创新，从而提供新的邻近度、缓存和延迟组合。这意味着，可能会发布新的邻近度和缓存解决方案，从而提供比最初选择的邻近度和缓存解决方案更高的性能效率。寻求机会减少延迟并提高整个系统的性能。例如，您是已完成一次性优化还是仍将根据需求的变化来优化系统？

最佳实践：

- **审核** 基于预测的资源需求周期性地重新选择邻近度和缓存解决方案。
- **基准测试** 在发布每种新的邻近度和缓存解决方案后，在 AWS 上对已知工作负载进行负载测试，并根据测试结果来估计最佳选择。
- **负载测试** 在发布每种新的邻近度和缓存解决方案后，在 AWS 上部署最新版本的系统，使用监控来捕获性能指标，然后基于性能/成本计算进行选择。
- **主动监控 — Amazon Cloud Watch 监控** 使用 Amazon CloudWatch 监控邻近度和缓存解决方案。
- **主动监控 — 第三方监控** 使用第三方工具监控邻近度和缓存解决方案。
- **基于警报的通知** 计划您的监控系统，使其在指标超出安全范围时自动向您发送提醒。
- **基于触发器的操作** 规划警报以触发修复或升级问题的自动操作。

性能 15. 您如何监控邻近度和缓存解决方案以确保获得预期性能？

系统性能可能会因内部或外部因素随时间而下降。通过监控系统的性能，您可以标识此下降情况并修复内部或外部因素。

最佳实践：

- **Amazon CloudWatch 监控** 使用 CloudWatch 监控实例。
- **第三方监控** 使用第三方工具监控系统。
- **定期审核** 定期审核您的监控仪表板。
- **基于警报的通知** 计划您的监控系统，使其在指标超出安全范围时自动向您发送提醒。
- **基于触发器的操作** 规划警报以触发修复或升级问题的自动操作。

性能 16. 您如何确保您拥有的邻近度和缓存解决方案满足需求？

对系统施加的需求量通常会因周期而异：产品生命周期（例如启动、增长等）；暂时的周期（例如一天中的某个时间、工作日或某个月等）；不可预测的周期（例如社交媒体可见性）；以及可预测的周期（例如电视剧集）。因拥有错误的邻近度和缓存解决方案而无法满足工作负载会降低用户体验，最糟糕的情况是导致系统故障。如果您拥有或计划拥有全球用户群，则尤为如此。

最佳实践：

- **已计划** 基于指标和/或已计划的事件来计划将来的邻近度或缓存解决方案。
- **监控** 监控随时间变化的缓存使用率和需求。
- **定期审核** 审核随时间变化的缓存使用率和需求。

成本优化支柱

成本 1. 您如何确保您的容量匹配而不会大幅度超出所需容量？

对于在支出和性能方面已均衡的架构，请确保您已使用已付费的所有资源并有效避免未充分利用的实例。任一方向的偏向利用率指标将对您业务的运营成本（因过度利用导致性能降低）或浪费的 AWS 支出（因超额预置）产生负面影响。

最佳实践：

- **基于需求的方法** 使用 Auto Scaling 响应不断变化的需求。
- **基于队列的方法** 运行您自己的 Amazon Simple Queue Service (SQS) 队列并基于需求运行/关闭实例。
- **基于时间的方法** 示例：利用不同的时区，在周末关闭开发与测试实例，根据季度或年度计划（例如黑色星期五）。
- **适当预置** 适当地预置服务（例如，Amazon DynamoDB、Amazon EBS（预置的 IOPS）、Amazon RDS、Amazon EMR 等）的吞吐量、大小和存储。

成本 2. 您如何优化 AWS 服务的使用率？

如果您使用应用程序级别服务，请确保很好地使用它们。例如，引入生命周期策略来控制 Amazon S3 使用率或利用服务（如 Amazon RDS 和 Amazon DynamoDB）实现极大的灵活性。检查相应的使用率包括验证 Amazon RDS 的多可用区部署或验证 Amazon DynamoDB 表中的预置 IOPS 是否适用。

最佳实践：

- **特定于服务的优化** 示例包括：最大程度地减少 Amazon EBS 的 I/O；避免将过多的小文件上传到 Amazon S3 中；对 Amazon EMR 广泛使用竞价型实例；等等。

成本 3. 您是否已选择适当的资源来满足您的成本目标？

确保您选择的 Amazon EC2 实例适合现有任务。AWS 鼓励使用基准测试评估，以确保所选实例类型已针对其工作负载进行优化。

最佳实践：

- **基于需求匹配实例配置文件** 例如，基于工作负载和实例描述（计算、内存或存储密集型）进行匹配。
- **第三方产品** 例如，使用第三方产品（如 CopperEgg 或 New Relic）确定适当的实例类型。
- **Amazon CloudWatch** 使用 CloudWatch 确定处理器负载。
- **自定义指标** 加载自定义内存脚本并使用 CloudWatch 检查内存使用率。
- **配置的应用程序** 配置您的应用程序，以便您了解何时使用哪种类型的 Amazon EBS（磁性、通用型 (SSD)、预置的 IOPS）。仅在需要时使用针对 EBS 优化的实例。

成本 4. 您是否已选择适当的定价模式来满足您的成本目标？

使用最适合您的工作负载的定价模式来最大程度地减少开支。最佳部署可以是完全的按需实例、按需实例和预留实例的组合，或者您可以包含竞价型实例（如果适用）。

最佳实践：

- **竞价型** 对所选工作负载使用竞价型实例。
- **分析使用情况** 定期分析使用情况并相应地购买预留实例。
- **出售预留实例** 随着您的需求不断变化，在预留实例市场上出售您不再需要的预留实例，然后购买其他实例。
- **自动操作** 拥有架构可让您关闭未使用的实例（例如，使用 Auto Scaling 在非工作时间内向下扩展）。
- **考虑成本** 将成本纳入区域选择中。

成本 5. 是否有可用来提高 ROI 的托管服务（级别高于 Amazon EC2、Amazon EBS、Amazon S3 的服务）？

Amazon EC2、Amazon EBS 和 Amazon S3 是“构建块”AWS 服务。托管服务（如 Amazon RDS 和 Amazon DynamoDB）是“更高级别”的 AWS 服务。通过使用这些托管服务，您可以减少或消除大量管理和运营开销，从而使您能够将精力放在应用程序和与业务相关的活动上。

最佳实践：

- **分析服务** 分析应用程序级别服务以了解您可使用的服务。
- **考虑适当的数据库** 在适当的情况下，使用 Amazon Relational Database Service (RDS)（Postgres、MySQL、SQL Server、Oracle Server）或 Amazon DynamoDB（或其他键值存储、NoSQL 备选项）。
- **考虑其他应用程序级别服务** 在适当的情况下，使用 Amazon Simple Queue Service (SQS)、Amazon Simple Notification Service (SNS)、Amazon Simple Email Service (SES)。
- **考虑 AWS CloudFormation、AWS Elastic Beanstalk 或 AWS Opsworks** 使用 AWS CloudFormation 模板/AWS Elastic Beanstalk/AWS OpsWorks 获得标准化和成本控制的好处。

成本 6. 您现在使用哪种访问控制和程序来管理 AWS 使用情况？

制定策略和机制以确保实现目标时耗费的成本是适当的。通过标记和 IAM 控制采用相互制衡方法，您可以进行创新而不会超支。

最佳实践：

- **建立组和角色**（示例：Dev/Test/Prod）；使用 AWS 管理机制（如 IAM）来控制每个组中可运行实例和资源的人员。（这适用于 AWS 服务或第三方解决方案。）
- **跟踪项目生命周期** 跟踪、度量和审核项目、团队和环境的生命周期以避免使用不必要的资源并为其付费。

成本 7. 您如何监控使用率和开支？

制定策略和程序以监控、控制和适当地分配您的成本。利用 AWS 提供的工具了解哪些人正在以多少成本使用何种资源。这将使您能够更深入地了解您的业务需求和团队运营。

最佳实践：

- **标记所有资源** 能够将账单中的更改与基础设施和使用率的更改关联在一起。
- **审核详细账单报告** 制定标准流程来加载和解释详细账单报告。
- **经济高效型架构** 制定针对使用率和开支（每单位 — 如用户、数据（以 GB 为单位））的计划。

- **监控** 使用 Amazon CloudWatch 或第三方提供程序（示例：Cloudability、CloudCheckr）定期监控使用情况和开支。
- **通知** 让我们团队的关键成员了解我们的开支是否超出了明确的限制。
- **使用 AWS 成本管理器**
- **财务驱动型退款方法** 使用此方法可将实例和资源分配给成本中心（如标记）。

成本 8. 您是否停用了不再需要的资源或停止了暂时不需要的资源？

确保您只为所使用的服务付费。从项目开始到结束期间实施变更控制和资源管理，以便您能在适当的情况下标识所需的流程更改或改进。使用 AWS Support 以获得有关如何为您的工作负载优化项目的建议：例如，何时使用 Auto Scaling、AWS OpsWorks、AWS Data Pipeline 或不同的 Amazon EC2 预置方法。

最佳实践：

- 将您的系统设计为在标识和停用低利用率的非关键型或非必需的实例/资源时正常处理实例终止。
- 使用现有流程来标识和停用孤立的资源。
- 基于系统或流程协调已停用的资源。

成本 9. 您在设计架构时是否考虑了数据传输费用？

确保您监控数据传输费用，以便您能制定可缓解这类成本的架构决策。例如，如果您是一个内容提供商并且一直在将内容直接从 Amazon S3 存储桶传输到您的最终用户，则可在将您的内容推送到 Amazon CloudFront CDN 时大大减少成本。请记住，小而有效的架构更改可大大减少您的操作成本。

最佳实践：

- 使用 CDN
- 构建以优化数据传输（应用程序设计、WAN 加速等）。
- 分析情况并使用 AWS Direct Connect 来节省资金和提高性能。
- 使架构的数据传输成本与高可用性 (HA) 和可靠性需求达到平衡。

成本 10. 您如何管理和/或考虑新服务的采用？

在 AWS，我们的目标是帮助您尽可能以最佳且经济高效的方式进行构建。新的服务和功能可直接降低您的成本。Amazon Glacier 就是其中的一个好例子，它提供了适用于不经常访问但出于业务或法律原因必须保留的数据的低成本的“冷”存储解决方案。另一个示例是 Amazon S3 的低冗余存储，它允许您选择更少的 Amazon S3 对象副本（较低冗余级别）来降低价格。制定这些决策时需要考虑一些影响，例如：“减少我的数据副本意味着什么？”或“我是否比我意识到的更需要访问此数据？”

最佳实践：

- 定期与您的 AWS 解决方案架构师、顾问或客户团队会面，并考虑您可采用哪些新的服务或功能来节省资金。