AWS Well-Architected 框架

性能效率支柱

性能效率支柱: AWS Well-Architected 框架

Table of Contents

性能效率支柱 – AWS Well-Architected Framework

发布日期：2024 年 11 月 6 日（文档修订）

本白皮书重点介绍 AWS Well-Architected Framework 的性能效率支柱。文中所述指导可帮助客户在 AWS 环境的设计、交付和维护过程中应用最佳实践。

简介

AWS Well-Architected Framework 能够帮助您理解在 AWS 上构建工作负载时所做决策的利弊。使用此框架有助于您了解在云中设计和运行可靠、安全、高效且经济实惠的可持续工作负载的架构最佳实践。该框架提供了一种方法，让您能够根据最佳实践持续衡量架构，从而确定需要改进的方面。我们相信，拥有架构完善的工作负载能够大大提高实现业务成功的可能性。

该框架基于六大支柱：

- 卓越运营
- 安全性
- 可靠性
- 性能效率
- 成本优化
- 可持续性

本白皮书重点介绍如何将性能效率支柱的原则应用于您的工作负载。在传统的本地环境中，实现持久的高性能比较困难。遵循本白皮书中的原则将帮助您在 AWS 上构建能够长期高效提供持续性能的架构。本文档中的指导和最佳实践涉及五个关键重点领域，可用作在 AWS 上构建高性能云解决方案的指导原则。这些重点领域是：

- 架构选择
- 计算和硬件
- 数据管理
- 网络和内容分发
- 流程和文化

本白皮书的目标读者是技术岗位的人员，例如首席技术官（CTO）、架构师、开发人员和运营团队成员。阅读本白皮书后，您将了解在设计高性能云架构时可以使用的 AWS 最佳实践和策略。

性能效率

性能效率支柱涉及高效地使用云资源以满足性能要求的能力，以及在需求变化和技术发展时保持该效率的能力。

主题

设计原则

以下设计原则可帮助您在云中实现并维护高效工作负载。

- 普及先进技术：通过将复杂的任务委派给云供应商，降低您的团队实施高级技术的难度。与要求您的 IT 团队学习有关托管和运行新技术的知识相比，考虑将新技术作为服务使用是一种更好的选择。例如，NoSQL 数据库、媒体转码和机器学习都是需要专业知识才能使用的技术。在云中，这些技术会转变为团队可以使用的服务，让团队能够专注于产品开发，而不是资源预置和管理。
- 数分钟内实现全球化部署：您可以在全球多个 AWS 区域中部署工作负载，从而以更低的成本为客户提供更低的延迟和更好的体验。
- 使用无服务器架构：借助无服务器架构，您无需运行和维护物理服务器即可执行传统计算活动。例如，无服务器存储服务可以充当静态网站（从而无需再使用 Web 服务器），事件服务则可以实现代码托管。这不仅能够消除管理物理服务器产生的运营负担，还可以借由以云规模运行的托管服务来降低业务成本。
- 提升实验频率：利用虚拟资源和可自动化的资源，您可以使用不同类型的实例、存储或配置来快速进行比较测试。
- 考虑软硬件协同编程：使用最符合自己目标的技术方法。例如，在为工作负载选择数据库或存储时考虑数据访问模式。

定义

通过重点关注以下领域，在云中实现高性能效率：

- 数据管理
- 网络和内容分发
- 流程和文化

采用数据驱动方法来构建高性能架构。收集架构各方面的数据，从总体设计到资源类型的选择与配置都包括在内。

定期审核自己的选择，确保充分利用不断发展的 AWS 云的优势。监控可以确保自己随时发现与预期性能的偏差。权衡架构以便提高性能，例如使用压缩或缓存，或放宽一致性要求。

架构选择

针对特定工作负载的最佳解决方案各不相同，而且解决方案通常会结合多种方法。Well-Architected 工作负载会使用多种解决方案，并且允许使用各种不同的功能来提高性能。

我们提供多种类型和配置的 AWS 资源，可让您更轻松地找到最能满足您需求的方法。此外，我们还提供了无法使用本地基础设施轻松实现的选项。例如，Amazon DynamoDB 之类的托管服务可以提供完全托管的 NoSQL 数据库，确保在任何规模下都只会有几毫秒的延迟。

该重点领域分享了有关如何选择高效、高性能的云资源和架构模式的指导和最佳实践。

最佳实践

- PERF01-BP01 了解并掌握可用的云服务和功能
- PERF01-BP02 使用云提供商或合适的合作伙伴提供的指导来了解架构模式和最佳实践
- PERF01-BP03 制定架构决策时考虑成本因素
- PERF01-BP04 评估权衡机制对客户和架构效率的影响
- PERF01-BP05 使用策略和参考架构
- PERF01-BP06 使用基准测试来推动制定架构决策
- PERF01-BP07 使用数据驱动的方法进行架构选择

PERF01-BP01 了解并掌握可用的云服务和功能

不断了解和发现可用的服务和配置，这些服务和配置有助于作出更好的架构决策，并提高工作负载架构的性能效率。

常见反模式：

- 将云用作联合数据中心。
- 迁移到云后，没有对应用程序进行现代化改造。
- 仅使用一种存储类型来存储所有需要继续保留的内容。
- 使用的实例类型最接近当前标准，但有时候需要使用更大的实例。
- 部署和管理作为托管服务提供的技术。

建立此最佳实践的好处：通过考虑采用新的服务和配置，可以大大提高性能、降低成本并减少维护工作负载所需的工作量。还有助于缩短支持云的产品的价值实现时间。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

AWS 不断发布新的服务和功能，可提高性能并降低云工作负载的成本。及时了解这些新服务和功能对于保持云的性能效率至关重要。对工作负载架构进行现代化改造还有助于提高工作效率、推动创新并解锁更多增长机会。

实施步骤

- 盘点相关服务的工作负载软件和架构。决定要进一步了解哪一类产品。
- 探索 AWS 产品，确定并了解有助于提高性能、降低成本和运营复杂性的相关服务和配置选项。
  - Amazon Web Services Cloud
  - AWS Academy
  - AWS 的新功能
  - AWS 博客
  - AWS Skill Builder
  - AWS 活动和网络研讨会
  - AWS 培训 和认证
  - AWS YouTube 频道
  - AWS 讲习会
  - AWS 社区
- 使用 Amazon Q 获取有关服务的相关信息和建议。
- 使用沙盒（非生产）环境来了解和试验新服务，且不会产生额外费用。
- 不断了解新的云服务和功能。

资源

相关文档：

- Overview of Amazon Web Services
- Amazon EC2 功能
- 通过 AWS 合作伙伴学习计划逐步学习
- AWS 培训和认证

- My learning path to become an AWS solutions architect
- AWS 架构中心
- AWS Partner Network
- AWS 解决方案库
- AWS Knowledge Center
- 在 AWS 上构建现代应用程序

相关视频：

- AWS re:Invent 2023 – What's new with Amazon EC2
- AWS re:Invent 2022 - Reduce your operational and infrastructure costs with Amazon ECS
- AWS re:Invent 2023 - Build with the efficiency, agility & innovation of the cloud with AWS
- AWS re:Invent 2022 - Deploy ML models for inference at high performance and low cost
- 这是我的架构

相关示例：

- AWS 示例
- AWS SDK 示例

PERF01-BP02 使用云提供商或合适的合作伙伴提供的指导来了解架构模式和最佳实践

利用云服务公司提供的资源（如文档、解决方案架构师、专业服务或合适的合作伙伴）来指导您制定架构决策。这些资源有助于您审查并改进架构，从而实现最佳性能。

常见反模式：

- 您将 AWS 视为普通的云提供商。
- 您没有按 AWS 服务的既定用途使用这些服务。
- 您在遵循所有指导时没有考虑到业务环境。

建立此最佳实践的好处：使用云提供商或合适的合作伙伴提供的指导，有助于您为工作负载选择合适的架构，让您对自己的决策充满信心。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

AWS 提供广泛的指导、文档和资源，有利于您构建和管理高效的云工作负载。AWS 文档提供了代码示例、教程和详细的服务说明。除文档外，AWS 还提供培训和认证计划、解决方案架构师和专业服务，可协助客户探索云服务的不同方面，并在 AWS 上实施高效的云架构。

利用这些资源深入了解宝贵的知识和最佳实践，节省时间，并在 AWS Cloud 中取得更好的成果。

实施步骤

- 查看 AWS 文档和指导并遵循最佳实践。这些资源有助于您高效地选择和配置服务来实现更好的性能。
  - AWS 文档（例如用户指南和白皮书）
  - AWS 博客
  - AWS 培训 和认证
  - AWS YouTube 频道
- 参加 AWS 合作伙伴活动（如 AWS 全球峰会、AWS re:Invent、用户群组和讲习会），向 AWS 专家学习关于使用 AWS 服务的最佳实践。
  - 通过 AWS 合作伙伴学习计划逐步学习
  - AWS 活动和网络研讨会
  - AWS 讲习会
  - AWS 社区
- 如需其他指导或产品信息，请联系 AWS 获取帮助。AWS解决方案架构师和AWS 专业服务提供关于实施解决方案的指导。AWS合作伙伴提供 AWS 专业知识，可帮助您实现业务敏捷性和创新能力。
- 如需技术支持来高效利用服务，请使用 支持。我们的支持计划旨在为您提供理想的工具组合以及获取专业知识的渠道，让您可以在优化性能、管理风险和控制成本的同时，使用 AWS 取得成功。

资源

相关文档：

- AWS 架构中心
- AWS Partner Network

- AWS 解决方案库
- AWS Knowledge Center
- AWS Enterprise Support

相关视频：

- 这是我的架构
- AWS re:Invent 2023 - Advanced event-driven patterns with Amazon EventBridge
- AWS re:Invent 2023 – 在 AWS 上实施分布式设计模式
- AWS re:Invent 2023 – 应用程序架构即代码

相关示例：

- AWS 示例
- AWS SDK 示例
- AWS 分析参考架构

PERF01-BP03 制定架构决策时考虑成本因素

制定架构决策时考虑成本因素，以便提高云工作负载的资源利用率和性能效率。意识到云工作负载的成本影响时，就更有可能充分利用有效资源，减少浪费。

常见反模式：

- 只使用一个系列的实例。
- 没有对照开源解决方案对许可的解决方案进行评估。
- 没有定义存储生命周期策略。
- 没有查看 AWS Cloud 的新服务和功能。
- 只使用数据块存储。

建立此最佳实践的好处：通过在制定决策时考虑成本因素，可以让您使用更有效的资源，并探索其他投资方式。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

针对成本优化工作负载能够提高资源利用率，避免在云工作负载中出现浪费。要在制定架构决策时考虑成本因素，通常包括合理调整工作负载组件的大小和实现弹性，从而提高云工作负载的性能效率。

实施步骤

- 制定成本目标，如云工作负载的预算限额。
- 确定会增加工作负载成本的关键组件（如实例和存储）。可以使用 AWS 定价计算器 和 AWS Cost Explorer 来确定工作负载中的关键成本驱动因素。
- 了解云中的定价模式，例如按需型实例、预留实例、节省计划和竞价型实例。
- 使用 Well-Architected 成本优化最佳实践来优化这些关键组件的成本。
- 持续监控和分析成本，发现工作负载中的成本优化机会。
  - 使用 AWS Budgets，针对无法接受的成本获取相关提醒。
  - 使用 AWS Compute Optimizer 或 AWS Trusted Advisor 获取成本优化建议。
  - 使用 AWS 成本异常检测自动进行成本异常检测和根本原因分析。

资源

相关文档：

- What is AWS Billing and Cost Management?
- 借助 AWS 实现成本优化
- Choosing an AWS cost management strategy
- A Beginner's Guide to AWS Cost Management
- A Detailed Overview of the Cost Intelligence Dashboard
- AWS 架构中心
- AWS 解决方案库
- AWS Knowledge Center

相关视频：

- 这是我的架构
- AWS re:Invent 2023 - What's new with AWS cost optimization

- AWS re:Invent 2023 - Optimize cost and performance and track progress toward mitigation
- AWS re:Invent 2023 - AWS storage cost-optimization best practices
- AWS re:Invent 2023 - Optimize costs in your multi-account environments

相关示例：

- AWS Compute Optimizer 演示代码
- Cost Optimization 讲习会
- Cloud Financial Management Technical Implementation Playbooks
- Startup optimization: Tuning application performance for maximum efficiency
- Serverless Optimization 讲习会（Performance and Cost）
- Scaling cost effective architectures

PERF01-BP04 评估权衡机制对客户和架构效率的影响

在评估与性能相关的改进时，确定哪些选择会对客户和工作负载效率产生影响。例如，如果使用键值数据存储可以提高系统性能，则评估这种更改的最终一致性对客户的影响就非常重要。

常见反模式：

- 您认为即便需要实施一些权衡机制，也要实现所有性能收益。
- 在性能问题已经非常严重时，只评估对工作负载的更改。

建立此最佳实践的好处：评估与性能相关的潜在改进时，必须决定更改时所采用的权衡机制是否符合工作负载要求。在某些情况下，可能必须实施额外的控制来补偿权衡机制。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

根据性能和客户影响确定架构中的关键领域。确定可以如何进行改进、这些改进带来的利弊，并确定改进对系统和用户体验的影响。例如，缓存数据有助于大幅提高性能，但需要就如何以及何时更新缓存的数据或使其变得无效而制定明确的策略，以便防止产生不正确的系统行为。

实施步骤

- 了解工作负载要求和 SLA。

- 明确定义评估因素。这些因素可能与工作负载的成本、可靠性、安全性和性能有关。
- 选择可以满足要求的架构和服务。
- 开展试验工作并执行概念验证（POC），评估权衡因素以及对客户和架构效率的影响。高度可用、高性能和安全的工作负载往往会消耗更多的云资源，但同时也会提供更好的客户体验。了解工作负载的复杂性、性能和成本之间的权衡因素。通常，重视其中两个因素会以牺牲第三个因素为代价。

资源

相关文档：

- Amazon Builders' Library
- Quick KPI
- Amazon CloudWatch RUM
- X-Ray 文档
- Understand resiliency patterns and trade-offs to architect efficiently in the cloud

相关视频：

- Optimize applications through Amazon CloudWatch RUM
- AWS re:Invent 2023 - Capacity, availability, cost efficiency: Pick three
- AWS re:Invent 2023 - Advanced integration patterns & trade-offs for loosely coupled systems

相关示例：

- Measure page load time with Amazon CloudWatch Synthetics
- Amazon CloudWatch RUM Web Client

PERF01-BP05 使用策略和参考架构

在选择服务和配置时使用内部策略和现有参考架构，可提高设计和实施工作负载时的效率。

常见反模式：

- 允许使用各种各样的技术，而这些技术可能会影响公司的管理开销。

建立此最佳实践的好处：制定架构、技术和供应商选择策略，有助于快速作出决策。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

在选择资源和架构时需要制定内部策略，这样在进行架构方面的选择时，可以提供应遵循的标准和指导方针。在选择合适的云服务时，这些指导方针可以简化决策过程，并提高性能效率。使用策略或参考架构部署工作负载。将服务集成到云部署中，然后使用性能测试来验证是否能继续满足性能要求。

实施步骤

- 清楚了解云工作负载的要求。
- 审查内部和外部策略，找出最有效的策略。
- 使用 AWS 提供的合适参考架构或行业最佳实践。
- 创建一个连续体，其中包含策略、标准、参考架构和针对常见情况的规范性指南。这样做可以让团队更快地开展工作。请酌情为垂直行业量身定制资产。
- 在沙盒环境中，为工作负载验证这些策略和参考架构。
- 随时了解行业标准和 AWS 更新，确保策略和参考架构有助于优化云工作负载。

资源

相关文档：

- AWS 架构中心
- AWS Partner Network
- AWS 解决方案库
- AWS Knowledge Center
- AWS Architecture Blog

相关视频：

- 这是我的架构
- AWS re:Invent 2022 - Accelerate value for your business with SAP & AWS reference architecture

相关示例：

- AWS 示例
- AWS SDK 示例

PERF01-BP06 使用基准测试来推动制定架构决策

对现有工作负载的性能进行基准测试，了解工作负载在云中的表现情况，并根据这些数据推动制定架构决策。

常见反模式：

- 启用普通的基准测试，而这些基准测试并不能反映出工作负载的特征。
- 将客户反馈和看法作为唯一的基准。

建立此最佳实践的好处：对当前实现进行基准测试可以衡量性能改进。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

结合使用基准测试与综合测试，评测工作负载组件的性能。相比负载测试，基准测试通常可以更快速地设置，并且适用于评估特定组件的技术。基准测试通常在新项目开始时进行，因为此时您还没有用于进行负载测试的完整解决方案。

您可以构建自己的自定义基准测试，也可以使用行业标准测试（如 TPC-DS），对工作负载进行基准测试。行业基准测试适用于比较不同的环境。对于在架构中想要执行的特定类型操作，自定义基准测试十分有用。

进行基准测试时，为了确保获得有效的结果，预热测试环境尤为重要。多次运行同一基准测试，确保捕获一段时间内的所有差异。

由于基准测试运行速度通常比负载测试快，它们可以在部署管道的早期使用，并能更快地提供有关性能偏差的反馈。评估组件或服务的重要更改时，可以使用基准测试快速了解是否有合理的理由来执行更改。结合使用基准测试与负载测试这一点很重要，因为负载测试能告知工作负载在生产环境中的表现如何。

实施步骤

- 规划和定义：

  - 为基准测试定义目标、基准、测试场景、指标（如 CPU 利用率、延迟或吞吐量）和 KPI。
  - 关注用户在用户体验方面的要求，以及响应时间和可访问性等因素。
  - 确定适用于工作负载的基准测试工具。可以使用与工作负载兼容的 Amazon CloudWatch 等 AWS 服务或第三方工具。
- 配置和检测：
  - 设置环境并配置资源。
  - 实施监控和日志记录来捕获测试结果。
- 基准测试和监控：
  - 执行基准测试并在测试期间监控指标。
- 分析和记录：
  - 记录基准测试过程和测试结果。
  - 对结果进行分析，确定瓶颈、趋势和需要改进的方面。
  - 利用测试结果制定架构决策并调整工作负载。这可能包括更改服务或采用新功能。
- 优化并重复：
  - 根据基准测试调整资源配置和分配。
  - 调整后重新测试工作负载，验证改进情况。
  - 记录经验教训，并重复该过程，确定其他需要改进的方面。

资源

相关文档：

- AWS 架构中心
- AWS Partner Network
- AWS 解决方案库
- AWS Knowledge Center
- Amazon CloudWatch RUM
- Amazon CloudWatch Synthetics
- Genomics workflows, Part 5: automated benchmarking
- Benchmark and optimize endpoint deployment in Amazon SageMaker AI JumpStart

相关视频：

- AWS re:Invent 2023 - Benchmarking AWS Lambda cold starts
- Benchmarking stateful services in the cloud
- 这是我的架构
- Optimize applications through Amazon CloudWatch RUM
- Demo of Amazon CloudWatch Synthetics

相关示例：

- AWS 示例
- AWS SDK 示例
- 分布式负载测试
- Measure page load time with Amazon CloudWatch Synthetics
- Amazon CloudWatch RUM Web Client

PERF01-BP07 使用数据驱动的方法进行架构选择

为架构选择确定清晰的数据驱动方法，确保使用合适的云服务和配置来满足特定业务需求。

常见反模式：

- 您认为当前的架构是静态的，不应随着时间的推移而更新。
- 选择架构时基于猜测和假设。
- 不断对架构进行更改，而不提供正当理由。

建立此最佳实践的好处：通过使用明确定义的方法来选择架构，可以利用数据来优化工作负载设计，在未来作出明智的决策。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

利用内部经验和云知识或外部资源（如已发布的应用场景或白皮书）来选择架构中的资源和服务。应制定一个明确定义的流程，该流程支持对可能会用于工作负载的不同服务进行试验和基准测试。

关键工作负载的积压工作不仅应包括用户案例（提供与业务和用户相关的功能），还应包括技术案例（创建工作负载的架构跑道）。该跑道依托于技术和新服务领域新的改进，并根据数据和适当的理由采用这些改进。这可以确保架构经得起未来考验，不会停滞不前。

实施步骤

- 与关键利益相关方一起确定工作负载要求，包括性能、可用性和成本方面的考量。考虑诸如用户数量和工作负载使用模式之类的因素。
- 创建架构跑道或技术积压工作，统筹确定它们与功能型待办事项的优先级。
- 评估和评测不同的云服务（有关详细信息，请参阅 PERF01-BP01 了解并掌握可用的云服务和功能）。
- 探索满足性能要求的不同架构模式，如微服务或无服务器（有关详细信息，请参阅 PERF01-BP02 使用云提供商或合适的合作伙伴提供的指导来了解架构模式和最佳实践）。

- 咨询其他团队、架构图和资源，例如 AWS 解决方案架构师、AWS 架构中心和 AWS Partner Network，协助为工作负载选择合适的架构。

- 定义吞吐量和响应时间等性能指标，以便于评估工作负载的性能。
- 进行试验并使用定义的指标来验证所选架构的性能。
- 持续监控并根据需要进行调整，从而使架构保持最佳性能。
- 记录所选架构和决策，作为将来更新和学习的参考。
- 根据经验教训、新技术以及可表明当前方法需要进行更改或存在问题的指标，不断审查和更新架构选择方法。

资源

相关文档：

- AWS 解决方案库
- AWS Knowledge Center
- Architectural Patterns to Build End-to-End Data Driven Applications on AWS

相关视频：

- 这是我的架构

- AWS re:Invent 2021 - Data-driven enterprise: Going from vision to value
- AWS re:Invent 2022 - Delivering sustainable, high-performing architectures
- AWS re:Invent 2023 - Optimize cost and performance and track progress toward mitigation
- AWS re:Invent 2022 - AWS optimization: Actionable steps for immediate results

相关示例：

- AWS 示例
- AWS SDK 示例

计算和硬件

适合特定工作负载的最佳计算方案会因应用程序设计、使用模式和配置设置而有所不同。架构可能会使用不同的计算方案来支持各种组件，并允许使用不同的功能来提高性能。为架构选择错误的计算方案可能会降低性能效率。

该重点领域分享了有关如何识别和优化计算选项以提高云端性能效率的指导和最佳实践。

最佳实践

- PERF02-BP01 为工作负载选择最佳计算方案
- PERF02-BP02 了解可用的计算配置和功能
- PERF02-BP03 收集与计算相关的指标
- PERF02-BP04 配置计算资源并合理调整资源规模
- PERF02-BP05 动态扩展计算资源
- PERF02-BP06 使用基于硬件的优化型计算加速器

PERF02-BP01 为工作负载选择最佳计算方案

通过为工作负载选择最合适的计算方案，可以提高性能，减少不必要的基础设施成本以及维护工作负载所需的运营工作。

常见反模式：

- 使用本地所用的计算方案。
- 对云计算方案、功能和解决方案以及这些解决方案可以如何提高计算性能缺乏认识。
- 为了满足扩展或性能要求，过度预置现有计算方案，而使用替代计算方案可以更准确地满足工作负载特征需求。

建立此最佳实践的好处：通过确定计算要求并对可用方案进行评估，可以让工作负载更高效地利用资源。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

为了提高性能效率而优化云工作负载时，请务必根据应用场景和性能要求选择最合适的计算方案。AWS 提供了多种计算方案，可满足云中不同工作负载的需求。例如，您可以使用 Amazon EC2 启动和管理虚拟服务器，使用 AWS Lambda 运行代码而不必预置或管理服务器，使用 Amazon ECS 或 Amazon EKS 运行和管理容器，或者使用 AWS Batch 并行处理大量数据。应根据自己的规模和计算需求，选择和配置最适合自己情况的计算解决方案。也可以考虑在单个工作负载中使用多种类型的计算解决方案，因为每种解决方案都有自己的优缺点。

以下步骤将指导您根据自身工作负载的特征和性能要求，选择合适的计算方案。

实施步骤

- 了解工作负载计算要求。需要考虑的关键要求包括：处理需求、流量模式、数据访问模式、扩展需求和延迟要求。
- 了解适用于工作负载的不同 AWS 计算服务。有关更多信息，请参阅 PERF01-BP01 了解并掌握可用的云服务和功能。以下介绍了一些关键的 AWS 计算方案、这些方案的特征和常见应用场景：

AWS 服务	主要特性	常见使用案例
Amazon Elastic Compute Cloud（Amazon EC2）	具有硬件专用选项、许可证要求，多种不同实例系列、处理器类型和计算加速器可供选择	直接迁移、整体式应用程序、混合环境、企业应用程序
Amazon Elastic Container Service（Amazon ECS）、Amazon Elastic Kubernetes Service（Amazon EKS）	轻松部署、环境一致、可扩展	微服务、混合环境
AWS Lambda	无服务器计算服务运行代码来响应事件并自动管理底层计算资源。	微服务、事件驱动的应用程序
AWS Batch	高效、动态地预置和扩展 Amazon Elastic Container Service（Amazon ECS）、Amazon Elastic	HPC，训练机器学习模型

AWS 服务	主要特性	常见使用案例
	Kubernetes Service（Amazon EKS）和 AWS Fargate 计算资源，并可根据任务要求选择使用按需型实例或竞价型实例	
Amazon Lightsail	预先配置的 Linux 和 Windows 应用程序，用于运行小型工作负载	简单的 Web 应用程序、自定义网站

- 评估与每种计算方案相关的成本（如每小时费用或数据传输）和管理开销（如修补和扩展）。
- 在非生产环境中进行试验和基准测试，确定哪种计算方案最能满足工作负载要求。
- 试验并确定新的计算解决方案后，规划迁移并验证性能指标。
- 使用 Amazon CloudWatch 等 AWS 监控工具和 AWS Compute Optimizer 等优化服务，根据实际使用模式持续优化计算资源。

资源

相关文档：

- 使用 AWS 进行云计算
- Amazon EC2 实例类型
- Amazon EKS 容器：Amazon EKS Worker 节点
- Amazon ECS 容器：Amazon ECS 容器实例
- 函数：Lambda 函数配置
- Prescriptive Guidance for Containers
- Prescriptive Guidance for Serverless

相关视频：

- AWS re:Invent 2023 - AWS Graviton: The best price performance for your AWS workloads
- AWS re:Invent 2023 - New Amazon Elastic Compute Cloud generative AI capabilities in AMS

- AWS re:Invent 2023 - What's new with Amazon Elastic Compute Cloud
- AWS re:Invent 2023 - Smart savings: Amazon Elastic Compute Cloud cost-optimization strategies
- AWS re:Invent 2021 - Powering next-gen Amazon Elastic Compute Cloud: Deep dive on the Nitro System
- AWS re:Invent 2019 - Optimize performance and cost for your AWS compute
- AWS re:Invent 2019 - Amazon Elastic Compute Cloud foundations
- AWS re:Invent 2022 - Deploy ML models for inference at high performance and low cost
- AWS re:Invent 2019 - Optimize performance and cost for your AWS compute
- Amazon EC2 foundations
- 部署 ML 模型，以便进行高性能和低成本的推理

相关示例：

- Migrating the Web application to containers
- 运行无服务器程序"Hello World"
- Amazon EKS 研讨会
- Amazon EC2 讲习会
- Efficient and Resilient Workloads with Amazon Elastic Compute Cloud Auto Scaling
- Migrating to AWS Graviton with Container Services

PERF02-BP02 了解可用的计算配置和功能

了解计算服务的可用配置选项和功能，帮助预置适量的资源并提高性能效率。

常见反模式：

- 没有依据工作负载特征评估计算方案或可用的实例系列。
- 过度预置计算资源来满足高峰需求。

建立此最佳实践的好处：熟悉 AWS 计算功能和配置，以便使用经过优化的计算解决方案，满足工作负载特征和需求。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

每种计算解决方案都有独特的配置和功能，可支持不同的工作负载特征和需求。了解这些方案如何完善工作负载，并确定哪些配置选项最适合您的应用程序。这些选项的示例包括实例系列、规模、功能（GPU、I/O）、突增、超时、函数大小、容器实例和并发。如果工作负载已经使用同一计算方案超过四周，并且预计这些特征在未来将保持不变，则可以使用 AWS Compute Optimizer 从 CPU 和内存角度查明当前计算方案是否适合工作负载。

实施步骤

- 了解工作负载要求（如 CPU 需求、内存和延迟）。
- 查看 AWS 文档和最佳实践，了解有助于提高计算性能的推荐配置选项。以下是一些需要考虑的关键配置选项：

配置选项	示例
实例类型	• 计算优化型实例非常适合需要较高 vCPU 与内存比的工作负载。 • 内存优化型实例提供大量内存来支持内存密集型工作负载。 • 存储优化型实例专为需要对本地存储进行大量顺序读写访问（IOPS）的工作负载而设计。
定价模式	• 按需型实例允许按小时或按秒使用计算容量，而无需做出长期承诺。这些实例非常适合超出性能基准需求的突增情况。 • 相比按需型实例，节省计划能够节省大量成本，从而换取在一年或三年内使用特定数量计算能力的承诺。 • 竞价型实例允许以折扣价将未使用的实例容量用于无状态的容错工作负载。
Auto Scaling	使用自动扩缩配置来让计算资源与流量模式相匹配。

配置选项	示例
调整大小	• 借助 Compute Optimizer，可以就哪种计算配置最符合计算特征，获得基于机器学习的建议。 • 使用 AWS Lambda Power Tuning 为 Lambda 函数选择最佳配置。
基于硬件的计算加速器	• 与基于 CPU 的替代方案相比，加速型计算实例执行图形处理或数据模式匹配等功能的效率更高。 • 对于机器学习工作负载，请利用特定于工作负载的专用硬件，例如 AWS Trainium、AWS Inferentia 和 Amazon EC2 DL1

资源

相关文档：

- 使用 AWS 进行云计算
- Amazon EC2 实例类型
- Amazon EC2 实例的处理器状态控制
- Amazon EKS 容器：Amazon EKS Worker 节点
- Amazon ECS 容器：Amazon ECS 容器实例
- 函数：Lambda 函数配置

相关视频：

- AWS re:Invent 2023 – AWS Graviton: The best price performance for your AWS workloads
- AWS re:Invent 2023 – New Amazon EC2 generative AI capabilities in AWS 管理控制台
- AWS re:Invent 2023 – What's new with Amazon EC2
- AWS re:Invent 2023 – Smart savings: Amazon EC2 cost-optimization strategies
- AWS re:Invent 2021 – Powering next-gen Amazon EC2: Deep dive on the Nitro System

- AWS re:Invent 2019 – Amazon EC2 foundations
- AWS re:Invent 2022 – Optimizing Amazon EKS for performance and cost on AWS

相关示例：

- Compute Optimizer 演示代码
- Amazon EC2 竞价型实例讲习会
- Efficient and Resilient Workloads with Amazon EC2 AWS Auto Scaling
- Graviton 开发人员讲习会
- AWS for Microsoft workloads immersion day
- AWS for Linux workloads immersion day
- AWS Compute Optimizer 演示代码
- Amazon EKS 讲习会

PERF02-BP03 收集与计算相关的指标

记录和跟踪与计算相关的指标，以便更好地了解计算资源的表现情况，并提高计算资源的性能和利用率。

常见反模式：

- 只手动搜索日志文件来查找指标。
- 只使用由监控软件记录的默认指标。
- 只在出现问题时审查指标。

建立此最佳实践的好处：收集与性能相关的指标有助于您根据业务要求调整应用程序性能，从而确保满足工作负载需求。收集指标还有利于您持续提高工作负载中的资源性能和利用率。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

云工作负载会生成大量数据，例如指标、日志和事件。在 AWS Cloud 中，收集指标是提高安全性、成本效率、性能和可持续性的关键步骤。AWS 使用监控服务（如 Amazon CloudWatch）提供各种与性能相关的指标，从而为您提供宝贵的洞察。CPU 利用率、内存利用率、磁盘 I/O 以及网络入站和出站

等指标有助于您深入了解利用率水平或性能瓶颈。将这些指标用作数据驱动方法的一部分，以便主动调整和优化工作负载的资源。 理想情况下，您应该在单一平台上收集与计算资源相关的所有指标，并实施留存策略以支持成本目标和运营目标。

实施步骤

- 确定哪些与性能相关的指标与您的工作负载相关。您应该收集有关资源利用率和云工作负载运行方式的指标（例如响应时间和吞吐量）。
  - Amazon EC2 默认指标
  - Amazon ECS 默认指标
  - Amazon EKS 默认指标
  - Lambda 默认指标
  - Amazon EC2 内存和磁盘指标
- 为工作负载选择并设置合适的日志记录和监控解决方案。
  - AWS native Observability
  - 适用于 OpenTelemetry 的 AWS Distro
  - Amazon Managed Service for Prometheus
- 根据工作负载要求为指标确定所需的筛选和聚合。
  - Quantify custom application metrics with Amazon CloudWatch Logs and metric filters
  - Collect custom metrics with Amazon CloudWatch strategic tagging
- 为指标配置数据留存策略，从而符合安全目标和运营目标。
  - CloudWatch 指标的默认数据留存
  - CloudWatch Logs 的默认数据留存
- 如有需要，可为指标创建警报和通知，协助您主动应对与性能相关的问题。
  - Create alarms for custom metrics using Amazon CloudWatch anomaly detection
  - Create metrics and alarms for specific web pages with Amazon CloudWatch RUM
- 使用自动化技术来部署指标和日志聚合代理。
  - AWS Systems Manager 自动化
  - OpenTelemetry Collector

资源

相关文档：

- 监控和可观测性
- Best practices: implementing observability with AWS
- Amazon CloudWatch 文档
- 使用 CloudWatch 代理从 Amazon EC2 实例和本地部署服务器中收集指标和日志
- 访问 AWS Lambda 的 Amazon CloudWatch Logs
- 将 CloudWatch Logs 与容器实例结合使用
- 发布自定义指标
- AWS Answers：集中式日志记录
- 发布 CloudWatch 指标的 AWS 服务
- Monitoring Amazon EKS on AWS Fargate

相关视频：

- AWS re:Invent 2023 – [LAUNCH] Application monitoring for modern workloads
- AWS re:Invent 2023 – Implementing application observability
- AWS re:Invent 2023 – Building an effective observability strategy
- AWS re:Invent 2023 – Seamless observability with AWS Distro for OpenTelemetry
- Application Performance Management on AWS

相关示例：

- AWS for Linux Workloads Immersion Day- Amazon CloudWatch
- Monitoring Amazon ECS clusters and containers
- Monitoring with Amazon CloudWatch dashboards
- Amazon EKS 讲习会

PERF02-BP04 配置计算资源并合理调整资源规模

配置计算资源并合理调整资源规模，使其满足您工作负载的性能要求，避免资源利用不足或过度利用。

常见反模式：

- 忽略工作负载性能要求，导致计算资源预置过度或预置不足。

- 只选择适用于所有工作负载的最大或最小实例。
- 为了便于管理，只使用一个实例系列。
- 忽略来自 AWS Cost Explorer 或 Compute Optimizer 的关于合理调整规模的建议。
- 没有重新评估新实例类型是否适合工作负载。
- 只为组织认证少量实例配置。

建立此最佳实践的好处：合理调整计算资源的规模后，可避免资源预置过度和预置不足，从而确保资源在云端以最佳方式运行。适当调整计算资源的规模通常可以提高性能和改进客户体验，同时还可以降低成本。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

合理调整规模使组织能够以经济高效的方式运营云基础设施，同时满足业务需求。云资源预置过度可能会导致额外成本，而预置不足可能导致性能和客户体验不佳。AWS 提供 AWS Compute Optimizer 和 AWS Trusted Advisor 之类的工具，这些工具使用历史数据为计算资源提供合理调整规模的建议。

实施步骤

- 选择最能满足您需求的实例类型：
  - 如何为我的工作负载选择适当的 Amazon EC2 实例类型？
  - Amazon EC2 Fleet 的基于属性的实例类型选择
  - 使用基于属性的实例类型选择创建自动扩缩组
  - Optimizing your Kubernetes compute costs with Karpenter consolidation
- 分析您的工作负载的各种性能特性，以及这些特性与内存、网络和 CPU 使用率之间的关系。根据这些数据选择最符合您的工作负载情况和性能目标的资源。
- 使用 AWS 监控工具（如 Amazon CloudWatch）监控资源使用情况。
- 为计算资源选择合适的配置。
  - 对于临时工作负载，请评估实例 Amazon CloudWatch 指标（例如 `CPUUtilization`），确定实例是利用不足还是利用过度。
  - 对于稳定工作负载，请定期检查 AWS 合理调整规模工具（如 AWS Compute Optimizer 和 AWS Trusted Advisor），从而挖掘优化计算资源和合理调整计算资源规模的机会。
- 在实际环境中实施之前，先在非生产环境中测试配置更改。
- 持续重新评估新的计算产品/服务，并与工作负载的需求进行比较。

资源

相关文档：

- 使用 AWS 进行云计算
- Amazon EC2 实例类型
- Amazon ECS 容器：Amazon ECS 容器实例
- Amazon EKS 容器：Amazon EKS Worker 节点
- 函数：Lambda 函数配置
- Amazon EC2 实例的处理器状态控制

相关视频：

- Amazon EC2 foundations
- AWS re:Invent 2023 – AWS Graviton: The best price performance for your AWS workloads
- AWS re:Invent 2023 – New Amazon EC2 generative AI capabilities in AWS 管理控制台
- AWS re:Invent 2023 – What's new with Amazon EC2
- AWS re:Invent 2023 – Smart savings: Amazon EC2 cost-optimization strategies
- AWS re:Invent 2021 – Powering next-gen Amazon EC2: Deep dive on the Nitro System
- AWS re:Invent 2019 – Amazon EC2 foundations

相关示例：

- AWS Compute Optimizer 演示代码
- Amazon EKS 讲习会
- 合理调整规模建议

PERF02-BP05 动态扩展计算资源

利用云的弹性根据需求动态增减计算资源，避免为工作负载预置的容量过多或者不足。

常见反模式：

- 通过手动增加容量来对警报做出反应。

- 使用本地所用的规模调整指南（通常是静态基础设施）。
- 在扩展事件之后保留增加的容量，而不是缩减容量。

建立此最佳实践的好处：配置和测试计算资源的弹性将有助于您节省资金、维护性能基准，以及在流量变化时提高可靠性。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

AWS 让您能够通过各种扩展机制灵活地动态扩展或缩减资源，以便满足不断变化的需求。动态扩展结合计算相关的指标，可使工作负载自动响应变化，并利用一系列最优的计算资源来实现目标。

您可以使用大量不同方法来实现资源的供需匹配。

- 目标跟踪方法：监控您的扩缩指标，并根据需要自动增加或减少容量。
- 预测性扩缩：根据每日和每周的趋势进行扩缩。
- 基于计划的方法：根据可预测的负载变化设置自己的扩缩计划。
- 服务扩缩：选择可根据设计自动扩缩的服务（如无服务器）。

您必须确保工作负载部署可以处理扩展事件和缩减事件。

实施步骤

- 计算实例、容器和函数都能够与自动扩缩服务相结合或作为此服务的一项功能来提供可实现弹性的机制。以下是自动扩缩机制的一些示例：

自动扩缩机制	使用情形
Amazon EC2 Auto Scaling	确保您具有正确数量的 Amazon EC2 实例，可用于处理应用程序负载。
Application Auto Scaling	自动扩缩 Amazon EC2 以外的各项 AWS 服务的资源，例如 AWS Lambda 函数或 Amazon Elastic Container Service（Amazon ECS）服务。
Kubernetes Cluster Autoscaler/Karpenter	自动扩缩 Kubernetes 集群。

- 扩缩通常与计算服务（如 Amazon EC2 实例或 AWS Lambda 函数）相关。此外，务必考虑配置非计算服务（如 AWS Glue）来满足需求。
- 验证扩缩指标是否与正在部署的工作负载的特性相匹配。如果您正在部署一个视频转码应用程序，预计 CPU 利用率为 100%，但不应将此作为主要指标，而应使用转码作业队列的深度。如果需要，可以对扩缩策略使用自定义指标。要选择正确的指标，请考虑以下关于 Amazon EC2 的指导：
  - 指标应该是有效的利用率指标，并描述实例的繁忙程度。
  - 指标值必须随着自动扩缩组中的实例数按比例增加或减少。
- 确保对自动扩缩组使用动态扩缩而不是手动扩缩。我们还建议在动态扩缩中使用目标跟踪扩缩策略。
- 确认工作负载部署可以同时处理扩展事件和缩减事件。例如，可以使用活动历史记录来验证自动扩缩组的扩缩活动。
- 评估工作负载，得出可预测的模式，从而在预期需求会发生预测性的计划内变化时主动扩缩。预测性扩缩可以避免过度预置容量。有关更多详细信息，请参阅 Predictive Scaling with Amazon EC2 Auto Scaling。

资源

相关文档：

- 使用 AWS 进行云计算
- Amazon EC2 实例类型
- Amazon ECS 容器：Amazon ECS 容器实例
- Amazon EKS 容器：Amazon EKS Worker 节点
- 函数：Lambda 函数配置
- Amazon EC2 实例的处理器状态控制
- Deep Dive on Amazon ECS Cluster Auto Scaling
- Introducing Karpenter – An Open-Source High-Performance Kubernetes Cluster Autoscaler

相关视频：

- AWS re:Invent 2023 – AWS Graviton: The best price performance for your AWS workloads
- AWS re:Invent 2023 – New Amazon EC2 generative AI capabilities in AWS Management Console
- AWS re:Invent 2023 – What's new with Amazon EC2
- AWS re:Invent 2023 – Smart savings: Amazon EC2 cost-optimization strategies

- AWS re:Invent 2021 – Powering next-gen Amazon EC2: Deep dive on the Nitro System
- AWS re:Invent 2019 – Amazon EC2 foundations

相关示例：

- Amazon EC2 Auto Scaling Group Examples
- Amazon EKS 研讨会
- Scale your Amazon EKS workloads by running on IPv6

PERF02-BP06 使用基于硬件的优化型计算加速器

与基于 CPU 的替代方案相比，使用硬件加速器可以更高效地执行某些功能。

常见反模式：

- 在工作负载中，没有对照性能更高和成本更低的专用实例，对通用实例进行基准测试。
- 使用基于硬件的计算加速器执行任务，而使用基于 CPU 的替代方案能更高效地完成这些任务。
- 不监控 GPU 使用情况。

建立此最佳实践的好处：通过使用基于硬件的加速器 [如图形处理单元（GPU）和现场可编程门阵列（FPGA）]，可以更高效地执行某些处理功能。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

加速型计算实例提供对基于硬件的计算加速器（如 GPU 和 FPGA）的访问。这些硬件加速器能够比基于 CPU 的替代方案更有效地执行某些功能，例如图形处理或数据模式匹配。许多加速工作负载（如渲染、转码和机器学习）在资源使用方面变化很大。仅在需要时运行此硬件，并在不需要时自动将其停用，从而提高整体性能效率。

实施步骤

- 确定可以满足要求的加速型计算实例。
- 对于机器学习工作负载，请利用针对工作负载的专用硬件，例如 AWS Trainium、AWS Inferentia 和 Amazon EC2 DL1。AWSInf2 实例等 Inferentia 实例相比同类 Amazon EC2 实例，性能功耗比提升了 50%。

- 收集加速型计算实例的使用情况指标。例如，按照使用 Amazon CloudWatch 收集 NVIDIA GPU 指标所述，使用 CloudWatch 代理收集 GPU 的 `utilization_gpu` 和 `utilization_memory` 等指标。
- 优化硬件加速器的代码、网络运营和设置，确保底层硬件得到充分利用。
  - 优化 GPU 设置
  - GPU Monitoring and Optimization in the Deep Learning AMI
  - Optimizing I/O for GPU performance tuning of deep learning training in Amazon SageMaker AI
- 使用最新的高性能库和 GPU 驱动程序。
- 使用自动化功能在不使用 GPU 实例时将其释放。

资源

相关文档：

- 在 Amazon ECS 上使用 GPU
- GPU 实例
- 使用 AWS Trainium 的实例
- 使用 AWS Inferentia 的实例
- Let's Architect! Architecting with custom chips and accelerators

- 加速计算
- Amazon EC2 VT1 Instances
- 如何为我的工作负载选择适当的 Amazon EC2 实例类型？
- Choose the best AI accelerator and model compilation for computer vision inference with Amazon SageMaker AI

相关视频：

- AWS re:Invent 2021 - How to select Amazon Elastic Compute Cloud GPU instances for deep learning
- AWS re:Invent 2022 - [NEW LAUNCH!] Introducing AWS Inferentia2-based Amazon EC2 Inf2 instances
- AWS re:Invent 2022 - Accelerate deep learning and innovate faster with AWS Trainium

- AWS re:Invent 2022 - Deep learning on AWS with NVIDIA: From training to deployment

相关示例：

- Amazon SageMaker AI and NVIDIA GPU Cloud (NGC)
- Use SageMaker AI with Trainium and Inferentia for optimized deep learning training and inferencing workloads
- Optimizing NLP models with Amazon Elastic Compute Cloud Inf1 instances in Amazon SageMaker AI

数据管理

针对特定系统的最佳数据管理解决方案往往取决于数据类型（数据块、文件或对象）、访问模式（随机或连续）、所需吞吐量、访问频率（在线、离线、归档）、更新频率（WORM、动态）以及可用性与持久性限制等因素。Well-Architected 工作负载使用专门构建的数据存储，这些存储允许使用不同的功能来提高性能。

该重点领域分享了优化数据存储、移动和访问模式以及数据存储性能效率的指导和最佳实践。

最佳实践

- PERF03-BP01 使用最能满足数据访问和存储要求的专用数据存储
- PERF03-BP02 评估数据存储的可用配置选项
- PERF03-BP03 收集和记录数据存储性能指标
- PERF03-BP04 实施可提高数据存储查询性能的策略
- PERF03-BP05 实施利用缓存的数据访问模式

PERF03-BP01 使用最能满足数据访问和存储要求的专用数据存储

了解数据特性（如数据的可共享性、大小、缓存大小、访问模式、延迟、吞吐量和持久性），为工作负载选择合适的专用数据存储（存储或数据库）。

常见反模式：

- 由于内部对某种特定类型的数据库解决方案具备相关经验且比较了解，因此坚持使用一种数据存储。
- 认为所有工作负载都有类似的数据存储和访问要求。
- 没有实施数据目录来清点数据资产。

建立此最佳实践的好处：了解数据特性和要求，有助于确定效率最高、性能最高的存储技术来满足工作负载需求。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

选择和实施数据存储时，要确保查询、扩展和存储特性支持工作负载数据要求。AWS 提供多种数据存储和数据库技术，包括数据块存储、对象存储、流式存储、文件系统、关系数据库、键值数据库、文档

数据库、内存数据库、图形数据库、时间序列数据库和分类账数据库等。每种数据管理解决方案都有可供您使用的选项和配置，可支持应用场景和数据模型。通过了解数据特性和要求，您可以摆脱单一存储技术以及有很多局限性的一刀切方法，专注于合理管理数据。

实施步骤

- 清点工作负载中存在的各种数据类型。
- 了解并记录数据特性和要求，包括：
  - 数据类型（非结构化、半结构化、关系型）
  - 数据量和增长
  - 数据持久性：持久、短暂、瞬时
  - ACID（原子性、一致性、隔离性、持久性）要求
  - 数据访问模式（读取密集型或写入密集型）
  - 延迟
  - 吞吐量
  - IOPS（每秒输入/输出操作数）
  - 数据留存期
- 了解可用于 AWS 工作负载的不同数据存储（存储和数据库服务），这些存储可以满足您的数据特性要求（如 PERF01-BP01 了解并掌握可用的云服务和功能中所述）。AWS 存储技术及其关键特性的一些示例包括：

Type	AWS 服务	主要特性
对象存储	Amazon S3	无限的可扩展性、高可用性以及多种可访问性选项。要在 Amazon S3 内外传输和访问对象，可以使用传输加速或接入点等服务来支持位置、安全需求和访问模式。
存档存储	Amazon Glacier	专为数据存档而打造。
流媒体存储	Amazon Kinesis	高效摄取和存储流媒体数据。

Type	AWS 服务	主要特性
	Amazon Managed Streaming for Apache Kafka（Amazon MSK）	
共享文件系统	Amazon Elastic File System (Amazon EFS)	可装载文件系统，可由多种类型的计算解决方案访问。
共享文件系统	：Amazon FSx	基于最新 AWS 计算解决方案而构建，支持四种常用文件系统：NetApp ONTAP、OpenZFS、Windows File Server 和 Lustre。Amazon FSx 延迟、吞吐量和 IOPS 因文件系统而不同，因此，在为您的工作负载需求选择合适的文件系统时应考虑这些因素。
数据块存储	Amazon Elastic Block Store (Amazon EBS)	可扩展、高性能的数据块存储服务，专为 Amazon Elastic Compute Cloud（Amazon EC2）设计。Amazon EBS 包括用于事务型、IOPS 密集型工作负载的 SSD 支持型存储，以及用于吞吐量密集型工作负载的 HDD 支持型存储。
关系数据库	Amazon Aurora、Amazon RDS、Amazon Redshift。	旨在支持 ACID（原子性、一致性、隔离性、持久性）事务，并保持参照完整性和数据强一致性。许多传统应用程序、企业资源规划（ERP）、客户关系管理（CRM）和电子商务都使用关系数据库来存储其数据。

Type	AWS 服务	主要特性
键值数据库	Amazon DynamoDB	已针对常见的访问模式进行优化，通常用于存储和检索大量数据。键值数据库的典型使用案例包括高流量 Web 应用程序、电子商务系统和游戏应用程序。
文档数据库	Amazon DocumentDB	旨在将半结构化数据存储为类似 JSON 的文档。这些数据库可帮助开发人员快速构建和更新应用程序，例如内容管理、目录和用户配置文件。
内存数据库	Amazon ElastiCache、适用于 Redis 的 Amazon MemoryDB	用于需要实时访问数据、最低延迟和最高吞吐量的应用程序。您可以将内存数据库用于应用程序缓存、会话管理、游戏排行榜、低延迟机器学习特征存放区、微服务消息传送系统和高吞吐量流式传输机制。
图形数据库	Amazon Neptune	用于需要大规模以毫秒延迟在高度连接的图形数据集之间浏览和查询数百万关系的应用程序。许多公司将图形数据库用于欺诈检测、社交网络和推荐引擎。
时间序列数据库	Amazon Timestream	用于高效收集、合成数据，并从不断变化的数据中获得见解。IoT 应用程序、DevOps 和工业遥测可以利用时间序列数据库。

Type	AWS 服务	主要特性
宽列	Amazon Keyspaces（Apache Cassandra 兼容）	使用表、行和列，但是与关系数据库不同的是，同一个表中各行的列名称和格式可能会有所不同。宽列存储常见于用于设备维护、队列管理和路线优化的大规模工业应用程序。
分类账	Amazon Quantum Ledger Database (Amazon QLDB)	提供可信中央机构，以维护每个应用程序的可扩展、不可变和允许以加密方式进行验证的交易记录。分类账数据库用于记录系统、供应链、注册甚至银行交易。

- 若要构建数据平台，可利用 AWS 上的现代数据架构来集成数据湖、数据仓库和专用数据存储。
- 为工作负载选择数据存储时需要考虑的关键问题如下：

问题	需要考虑的事项
数据结构如何？	• 若是非结构化数据，可以考虑使用对象存储（例如 Amazon S3）或 NoSQL 数据库（例如 Amazon DocumentDB） • 若是键值数据，可以考虑 DynamoDB、与 Redis OSS 兼容的 Amazon ElastiCache 或 Amazon MemoryDB
需要什么级别的参照完整性？	• 对于外键约束，Amazon RDS 和 Aurora 等关系数据库可以提供这种级别的完整性。 • 通常，在 NoSQL 数据模型中，您可以将数据去规范化到单个文档或文档集合，以便在单个请求中进行检索，而不是跨各文档或各表联接。

问题	需要考虑的事项
是否要求符合 ACID（原子性、一致性、隔离性、持久性）？	• 如果需要与关系数据库关联的 ACID 属性，请考虑使用关系数据库，例如 Amazon RDS 和 Aurora。 • 如果 NoSQL 数据库需要强一致性，则可以在 DynamoDB 中使用强一致性读取。
存储要求将如何随时间变化？ 这对可扩展性有何影响？	• Dynamo DB 和 Amazon Quantum Ledger Database（Amazon QLDB）等无服务器数据库会动态扩展。 • 关系数据库的预置存储空间设有上限，一旦达到这些限制，往往必须使用分片等机制进行水平分区。
读查询与写查询的比例是多少？ 缓存有可能提高性能吗？	• 读取密集型工作负载可以从缓存层中受益，例如 ElastiCache 或 DAX（若数据库为 DynamoDB）。 • 读取操作也可以通过关系数据库（如 Amazon RDS）分流到只读副本。
存储和修改（OLTP – Online Transaction Processing，联机事务处理）还是检索和报告（OLAP – Online Analytical Processing，联机分析处理）具有更高的优先级？	• 对于高吞吐量的按原样读取事务处理，可以考虑使用 NoSQL 数据库，例如 DynamoDB。 • 对于具有一致性的高吞吐量和复杂的读取模式（如联接），请使用 Amazon RDS。 • 对于分析查询，可以考虑使用 Amazon Redshift 等列存数据库，或者将数据导出到 Amazon S3 后使用 Athena 或 Amazon Quick 进行分析。

问题	需要考虑的事项
数据需要什么级别的持久性？	• Aurora 自动在一个区域内的三个可用区复制数据，这意味着数据具有高度的持久性，丢失的可能性较小。 • DynamoDB 自动跨多个可用区复制，提供高可用性和数据持久性。 • Amazon S3 提供 11 个 9 的持久性。许多数据库服务（如 Amazon RDS 和 DynamoDB）支持将数据导出到 Amazon S3，以便进行长期留存和存档。
是否希望摆脱商用数据库引擎或许可成本？	• 考虑使用 Amazon RDS 或 Aurora 上的开源引擎，如 PostgreSQL 和 MySQL。 • 利用 AWS Database Migration Service 和 AWS Schema Conversion Tool 执行从商用数据库引擎到开源引擎的迁移
对数据库的运维有什么期望？ 迁移到托管服务是主要的关注点吗？	• 利用 Amazon RDS 而不是 Amazon EC2，以及利用 DynamoDB 或 Amazon DocumentDB 而不是自行托管 NoSQL 数据库，可以减少运维开销。
当前如何访问数据库？ 是只有应用程序访问，还是有商业智能（BI）用户和其他互联的现成应用程序？	• 如果依赖外部工具，可能需要保持与这些工具支持的数据库的兼容性。Amazon RDS 与其支持的差异引擎版本完全兼容，包括 Microsoft SQL Server、Oracle、MySQL 和 PostgreSQL。

- 在非生产环境中进行试验和基准测试，确定哪种数据存储可以满足工作负载要求。

资源

相关文档：

- Amazon EBS 卷类型
- Amazon EC2 存储

- Amazon EFS：Amazon EFS 性能
- 适用于 Lustre 的 Amazon FSx 性能
- 适用于 Windows File Server 的 Amazon FSx 性能
- Amazon Glacier：Amazon Glacier 文档
- Amazon S3：请求速率和性能注意事项
- 使用 AWS 进行云存储
- Amazon EBS I/O 特性
- AWS 云数据库
- AWS 数据库缓存
- DynamoDB Accelerator
- Amazon Aurora 最佳实践
- Amazon Redshift 性能
- Amazon Athena 十大性能技巧
- Amazon Redshift Spectrum 最佳实践
- Amazon DynamoDB 最佳实践
- 在 Amazon EC2 和 Amazon RDS 之间进行选择
- 实施 Amazon ElastiCache 的最佳实践

相关视频：

- AWS re:Invent 2023: Improve Amazon Elastic Block Store efficiency and be more cost-efficient
- AWS re:Invent 2023: Optimizing storage price and performance with Amazon Simple Storage Service
- AWS re:Invent 2023: Building and optimizing a data lake on Amazon Simple Storage Service
- AWS re:Invent 2022: Building modern data architectures on AWS
- AWS re:Invent 2022: Building data mesh architectures on AWS
- AWS re:Invent 2023: Deep dive into Amazon Aurora and its innovations
- AWS re:Invent 2023: Advanced data modeling with Amazon DynamoDB
- AWS re:Invent 2022: Modernize apps with purpose-built databases
- Amazon DynamoDB deep dive: Advanced design patterns

相关示例：

- AWS Purpose Built Databases 讲习会
- Databases for Developers
- AWS Modern Data Architecture Immersion Day
- Build a Data Mesh on AWS
- Amazon S3 示例
- Optimize Data Pattern using Amazon Redshift Data Sharing
- Database Migrations
- MS SQL Server - AWS Database Migration Service (AWS DMS) Replication Demo
- Database Modernization Hands On 讲习会
- Amazon Neptune 示例

PERF03-BP02 评估数据存储的可用配置选项

了解并评估数据存储的各种可用功能和配置选项，从而优化工作负载的存储空间和性能。

常见反模式：

- 对所有工作负载都只使用一种存储类型，例如 Amazon EBS。
- 对所有工作负载都使用预调配 IOPS，而没有对所有存储层进行真实测试。
- 不了解所选数据管理解决方案的配置选项。
- 只依赖于增加实例大小，而没有考虑其他可用的配置选项。
- 没有测试数据存储的扩展特性。

建立此最佳实践的好处：通过探索和试用数据存储选项，也许能够降低基础设施成本、提高性能并减少维护工作负载所需的工作量。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

根据数据存储和访问要求，一个工作负载能够使用一个或多个数据存储。要优化性能效率和成本，必须评估数据访问模式来确定适当的数据存储配置。在研究数据存储选项时，要考虑存储选项、内存、计

算、只读副本、一致性要求、连接池和缓存选项等各个方面。尝试使用这些不同的配置选项来改进性能效率指标。

实施步骤

- 了解数据存储的当前配置（如实例类型、存储大小或数据库引擎版本）。
- 查看 AWS 文档和最佳实践，了解有助于提高数据存储性能的推荐配置选项。需要考虑的关键数据存储选项如下：

配置选项	示例
分流读取操作（例如只读副本和缓存）	• 对于 DynamoDB 表，您可以使用 DAX 缓存功能来分流读取操作。 • 您可以创建一个 Amazon ElastiCache（Redis OSS）集群，并将应用程序配置为首先从缓存中读取，并在请求的项目不存在时使用数据库。 • 关系数据库（如 Amazon RDS 和 Aurora）以及预置的 NoSQL 数据库（如 Neptune 和 Amazon DocumentDB）全部支持添加只读副本，以便分流工作负载的读取部分。 • DynamoDB 等无服务器数据库将自动扩展。确保您预置了足够的读取容量单位（RCU）来处理工作负载。
扩展写入（例如分区键分片或引入队列）	• 对于关系数据库，您可以增加实例的大小来适应增加的工作负载，或增加预调配 IOPS 来增加底层存储的吞吐量。 • 您还可以在数据库前面引入队列，而不是直接写入数据库。借助此模式，您可以将摄取操作与数据库解耦，并控制流量，这样数据库就不会过载。 • 对写入请求进行批处理，而不是创建许多短期事务，这样有助于提高有大量写入的关系数据库的吞吐量。

配置选项	示例
	• 像 DynamoDB 这样的无服务器数据库可以自动扩展写入吞吐量，也可以根据容量模式调整预置的写入容量单位（WCU）。 • 当达到给定分区键的吞吐量限制时，仍然会遇到热分区问题。这可以通过选择更均匀分布的分区键或对分区键进行写分片来缓解。
用于管理数据集生命周期的策略	• 您可以使用 Amazon S3 生命周期在对象的整个生命周期中对其进行管理。如果访问模式未知、变化或不可预测，则可以使用 Amazon S3 Intelligent-Tiering，因为此功能可监控访问模式并自动将尚未访问的对象移动到成本较低的访问层。您可以利用 Amazon S3 Storage Lens 存储统计管理工具指标来识别生命周期管理中的优化机会和差距。 • Amazon EFS 生命周期管理会自动为文件系统管理文件存储。
连接管理和连接池	• Amazon RDS 代理可与 Amazon RDS 和 Aurora 结合使用来管理与数据库的连接。 • DynamoDB 等无服务器数据库没有与之关联的连接，但会考虑根据预置容量和自动扩展策略来处理负载峰值。

- 在非生产环境中进行试验和基准测试，确定哪种配置选项可以满足工作负载要求。
- 试验完成后，规划迁移并验证性能指标。
- 使用 AWS 监控工具（如 Amazon CloudWatch）和优化工具（如 Amazon S3 Storage Lens 存储统计管理工具），在实际使用模式下持续优化数据存储。

资源

相关文档：

- 使用 AWS 进行云存储

- Amazon EBS 卷类型
- Amazon EC2 存储
- Amazon EFS：Amazon EFS 性能
- 适用于 Lustre 的 Amazon FSx 性能
- 适用于 Windows File Server 的 Amazon FSx 性能
- Amazon Glacier：Amazon Glacier 文档
- Amazon S3：请求速率和性能注意事项
- Amazon EBS I/O 特性
- 云数据库AWS
- AWS 数据库缓存
- DynamoDB Accelerator
- Amazon Aurora 最佳实践
- Amazon Redshift 性能
- Amazon Athena 十大性能技巧
- Amazon Redshift Spectrum 最佳实践
- Amazon DynamoDB 最佳实践

相关视频：

- AWS re:Invent 2023: Improve Amazon Elastic Block Store efficiency and be more cost-efficient
- AWS re:Invent 2023: Optimize storage price and performance with Amazon Simple Storage Service
- AWS re:Invent 2023: Building and optimizing a data lake on Amazon Simple Storage Service
- AWS re:Invent 2023: What's new with AWS file storage
- AWS re:Invent 2023: Dive deep into Amazon DynamoDB

相关示例：

- AWS Purpose Built Databases 讲习会
- Databases for Developers
- AWS Modern Data Architecture Immersion Day

- Amazon EBS Autoscale
- Amazon S3 示例
- Amazon DynamoDB 示例
- AWS 数据库迁移示例
- Database Modernization 讲习会
- Working with parameters on your Amazon RDS for Postgress DB

PERF03-BP03 收集和记录数据存储性能指标

跟踪并记录数据存储的相关性能指标，了解数据管理解决方案的执行情况。这些指标有助于您优化数据存储，验证是否满足工作负载要求，并清晰地概述工作负载的表现情况。

常见反模式：

- 只手动搜索日志文件来查找指标。
- 只将指标发布到团队使用的内部工具，而没有全面了解工作负载。
- 只使用由自己选定的监控软件记录的默认指标。
- 只在出现问题时审查指标。
- 只监控系统级指标，而不捕获数据访问或使用情况指标。

建立此最佳实践的好处：建立性能基准有助于了解工作负载的正常行为和要求。可以更快地识别和调试异常模式，从而提高数据存储的性能和可靠性。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

要监控数据存储的性能，必须记录一段时间的多项性能指标。这样您便可以检测异常并根据业务指标衡量性能，确保满足您的工作负载需求。

指标既应包括支持数据存储的底层系统指标，也应包括数据库指标。底层系统指标可能包括 CPU 利用率、内存、可用磁盘存储、磁盘 I/O、缓存命中率以及网络入站和出站指标，而数据存储指标可能包括每秒事务数、最多的查询、平均查询速率、响应时间、索引使用情况、表锁定、查询超时和打开的连接数。这些数据对于了解工作负载的表现情况以及数据管理解决方案的使用方式至关重要。在数据驱动方法中使用这些指标，以便调整和优化工作负载的资源。

使用各种工具、库和系统来记录与数据库性能相关的性能测量值。

实施步骤

- 确定要跟踪的数据存储关键性能指标。
  - Amazon S3 指标与维度
  - 监控 Amazon RDS 实例中的指标
  - 在 Amazon RDS 上使用性能详情监控数据库负载
  - 增强监测概述
  - DynamoDB 指标与维度
  - 监控 DynamoDB Accelerator
  - 使用 Amazon CloudWatch 监控 Amazon MemoryDB
  - 应监控哪些指标？
  - 监控 Amazon Redshift 集群性能
  - Timestream 指标与维度
  - Amazon Aurora 的 Amazon CloudWatch 指标
  - 在 Amazon Keyspaces（Apache Cassandra 兼容）中记录和监控
  - 监控 Amazon Neptune 资源
- 使用经批准的日志记录和监控解决方案来收集这些指标。Amazon CloudWatch 可以收集架构中各种资源的指标。您也可以收集和发布自定义指标，用于显示业务指标或派生指标。使用 CloudWatch 或第三方解决方案来设置超出阈值时显示的警报。
- 检查数据存储监控，确定其能否受益于可检测性能异常的机器学习解决方案。
  - Amazon DevOps Guru for Amazon RDS 会显示性能问题，并提出纠正措施的建议。
- 在监控和日志记录解决方案中配置数据留存，从而满足您的安全和运营目标。
  - CloudWatch 指标的默认数据留存
  - CloudWatch Logs 的默认数据留存

资源

相关文档：

- AWS 数据库缓存
- Amazon Athena 十大性能技巧

- Amazon Aurora 最佳实践
- DynamoDB Accelerator
- Amazon DynamoDB 最佳实践
- Amazon Redshift Spectrum 最佳实践
- Amazon Redshift 性能
- AWS 云数据库
- Amazon RDS 性能详情

相关视频：

- AWS re:Invent 2022 - Performance monitoring with Amazon RDS and Aurora, featuring Autodesk
- Database Performance Monitoring and Tuning with Amazon DevOps Guru for Amazon RDS
- AWS re:Invent 2023 - What's new with AWS file storage
- AWS re:Invent 2023 - Dive deep into Amazon DynamoDB
- AWS re:Invent 2023 - Building and optimizing a data lake on Amazon S3
- AWS re:Invent 2023 - What's new with AWS file storage
- AWS re:Invent 2023 - Dive deep into Amazon DynamoDB
- Best Practices for Monitoring Redis Workloads on Amazon ElastiCache

相关示例：

- AWS Dataset Ingestion Metrics Collection Framework
- Amazon RDS Monitorng 讲习会
- AWS Purpose Built Databases 讲习会

PERF03-BP04 实施可提高数据存储查询性能的策略

实施可优化数据和改进数据查询的策略，从而提高工作负载的可扩展性和性能效率。

常见反模式：

- 没有对数据存储中的数据进行分区。
- 在数据存储中只以一种文件格式存储数据。

- 没有在数据存储中使用索引。

建立此最佳实践的好处：优化数据和查询性能可以提高效率、降低成本并改善用户体验。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

数据优化和查询调整是提高数据存储性能效率的关键环节，因为这会影响整个云工作负载的性能和响应能力。如果查询未经优化，则会耗用更多的资源并产生更多的瓶颈，从而降低数据存储的整体效率。

数据优化会涵盖多种技术，旨在确保高效的数据存储和访问，同时还有助于改进在数据存储中的查询性能。关键策略包括数据分区、数据压缩和数据去规范化，这有助于针对存储和访问优化数据。

实施步骤

- 了解并分析在数据存储中执行的关键数据查询。
- 识别数据存储中运行速度较慢的查询，并使用查询计划了解当前状态。
  - 在 Amazon Redshift 中分析查询计划
  - 在 Athena 中使用 EXPLAIN 和 EXPLAIN ANALYZE
- 实施可提高查询性能的策略。一些关键策略包括：
  - 使用列式文件格式（如 Parquet 或 ORC）。
  - 压缩数据存储中的数据，减少存储空间和 I/O 操作。
  - 进行数据分区，将数据分割成更小的部分，减少数据扫描时间。
    - 在 Athena 中对数据进行分区
    - 分区和数据分发
  - 对查询中的常用列编制数据索引。
  - 使用实体化视图频繁地进行查询。
    - 了解实体化视图
    - 在 Amazon Redshift 中创建实体化视图
  - 为查询选择合适的联接操作。联接两个表时，请在联接的左侧指定较大的表，在联接的右侧指定较小的表。
  - 实施分布式缓存解决方案，从而缩短延迟并减少数据库 I/O 操作次数。
  - 执行定期维护，例如 vaccum 操作、重新索引和进行统计。
- 在非生产环境中试验和测试策略。

资源

相关文档：

- Amazon Aurora 最佳实践
- Amazon Redshift 性能
- Amazon Athena 十大性能技巧
- AWS 数据库缓存
- 实施 Amazon ElastiCache 的最佳实践
- 在 Athena 中对数据进行分区

相关视频：

- AWS re:Invent 2023 - AWS storage cost-optimization best practices
- AWS re:Invent 2022 - Performance monitoring with Amazon RDS and Aurora, featuring Autodesk
- Optimize Amazon Athena Queries with New Query Analysis Tools

相关示例：

- AWS Purpose Built Databases 讲习会

PERF03-BP05 实施利用缓存的数据访问模式

实施可从缓存数据受益的访问模式，以便快速检索经常访问的数据。

常见反模式：

- 缓存经常变化的数据。
- 依赖缓存的数据，就好像这些数据是持久存储的，并且始终可用。
- 不考虑缓存数据的一致性。
- 不监控缓存实现方案的效率。

建立此最佳实践的好处：将数据存储在缓存中可以改善读取延迟、读取吞吐量、用户体验和整体效率，还可以降低成本。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

缓存是一种软件或硬件组件，旨在存储数据，以便将来可以更快或更高效地处理对相同数据的请求。如果存储在缓存中的数据丢失，则可以通过重复先前的计算或从其他数据存储中获取数据进行重建。

数据缓存可能是提高应用程序整体性能和减轻底层主数据源负担的最有效策略之一。数据可以在应用程序的多个级别上缓存，例如在进行远程调用的应用程序内缓存（称作客户端缓存），或者使用快速辅助服务来存储数据（称作远程缓存）。

客户端缓存

借助客户端缓存，每个客户端（查询后端数据存储的应用程序或服务）都可以在本地将特定查询的结果存储指定的时间。可以通过先检查本地客户端缓存，减少通过网络向数据存储发出的请求数量。如果结果不存在，则应用程序可以查询数据存储并将这些结果存储在本地。这种模式允许每个客户端将数据存储在尽可能近的位置（客户端本身），从而尽可能降低延迟。当后端数据存储不可用时，客户端还可以继续支持某些查询，从而提高整个系统的可用性。

这种方法的一个缺点是，当涉及多个客户端时，它们可能会在本地存储相同的缓存数据。这会导致这些客户端之间存在重复的存储使用情况和数据不一致性。一个客户端可能刚缓存查询结果，而一分钟后，另一个客户端可能运行相同的查询并得到不同的结果。

远程缓存

为了解决客户端之间的重复数据问题，可以使用快速的外部服务或远程缓存来存储查询的数据。在查询后端数据存储之前，每个客户端都将检查远程缓存，而不是检查本地数据存储。这种策略可在客户端之间实现更加一致的响应、更高的存储数据效率以及更高的缓存数据量，因为存储空间可独立于客户端进行扩展。

远程缓存的缺点是整个系统的延迟可能会更高，因为需要额外的网络跃点数来检查远程缓存。客户端缓存可以与远程缓存一起使用，形成多级缓存来缩短延迟。

实施步骤

- 确定可以从缓存中受益的数据库、API 和网络服务。读取工作负载繁重、读写比率高或扩展成本高昂的服务适合使用缓存。
  - 数据库缓存
  - 启用 API 缓存以增强响应能力
- 确定最适合您的访问模式的适当缓存策略类型。

  - 缓存策略
  - AWS 缓存解决方案
- 遵循数据存储的缓存最佳实践。
- 为所有数据配置缓存失效策略，例如生存时间（TTL），以平衡数据的时效性并减轻后端数据存储的压力。
- 启用诸如自动连接重试、指数回退、客户端超时和客户端连接池等功能（如果有），因为它们可以提高性能和可靠性。
  - Best practices: Redis clients and Amazon ElastiCache for Redis
- 监控缓存命中率，目标为 80% 或更高。低于此值可能表示缓存大小不足，或访问模式无法从缓存中受益。
  - Which metrics should I monitor?
  - Best practices for monitoring Redis workloads on Amazon ElastiCache
  - Monitoring best practices with Amazon ElastiCache (Redis OSS) using Amazon CloudWatch
- 实施数据复制，将读取操作分流到多个实例，以提高数据读取性能和可用性。

资源

相关文档：

- Using the Amazon ElastiCache Well-Architected Lens
- Monitoring best practices with Amazon ElastiCache (Redis OSS) using Amazon CloudWatch
- 应监控哪些指标？
- Performance at Scale with Amazon ElastiCache 白皮书
- 缓存挑战和策略

相关视频：

- Amazon ElastiCache Learning Path
- Design for success with Amazon ElastiCache best practices
- AWS re:Invent 2020 - Design for success with Amazon ElastiCache best practices
- AWS re:Invent 2023 - [LAUNCH] Introducing Amazon ElastiCache Serverless
- AWS re:Invent 2022 - 5 great ways to reimagine your data layer with Redis
- AWS re:Invent 2021 - Deep dive on Amazon ElastiCache (Redis OSS)

相关示例：

- 使用 Amazon ElastiCache for Redis 提升 MySQL 数据库性能

网络和内容分发

适合某个工作负载的最佳网络解决方案会因延迟、吞吐量要求、抖动和带宽而有所不同。物理约束（例如用户资源或本地资源）决定位置选项。这些约束可以通过边缘站点或资源置放来抵消。

在 AWS，网络资源以虚拟化形式存在，而且以多种类型和配置提供。这让您可以更轻松地找到贴合您需求的网络方案。AWS 提供多种产品功能（例如增强联网、Amazon EC2 联网优化实例、Amazon S3 传输加速和动态 Amazon CloudFront）来优化网络流量。AWS 还可以提供多种联网功能（例如 Amazon Route 53 延迟路由、Amazon VPC 端点、AWS Direct Connect 和 AWS Global Accelerator）来减少网络距离或抖动。

该重点领域分享了在云端设计、配置和运行高效的联网和内容分发解决方案的指导和最佳实践。

最佳实践

- PERF04-BP01 了解联网对性能的影响
- PERF04-BP02 评估可用的联网功能
- PERF04-BP03 为工作负载选择合适的专用连接或 VPN
- PERF04-BP04 使用负载平衡在多个资源之间分配流量
- PERF04-BP05 选择网络协议以提高性能
- PERF04-BP06 根据网络要求选择工作负载的位置
- PERF04-BP07 根据指标优化网络配置

PERF04-BP01 了解联网对性能的影响

分析并了解与网络相关的决策如何影响您的工作负载，从而提供更高的性能和更好的用户体验。

常见反模式：

- 所有流量都会流经现有的数据中心。
- 通过中央防火墙路由所有流量，而不是使用云原生网络安全工具。
- 在不了解实际使用要求的情况下预置 AWS Direct Connect 连接。
- 在确立联网解决方案时，未考虑工作负载特性和加密开销。
- 将本地概念和策略用于云中的联网解决方案。

建立此最佳实践的好处：通过了解联网如何影响工作负载性能，有助于您识别潜在的瓶颈、改善用户体验、提高可靠性并在工作负载发生变化时减少运营维护。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

网络负责应用程序组件、云服务、边缘网络和本地数据之间的连接，因此，它会严重影响工作负载性能。除了工作负载性能之外，用户体验也会受到网络延迟、带宽、协议、位置、网络拥塞、抖动、吞吐量和路由规则的影响。

清楚记录工作负载的联网要求列表，包括延迟、数据包大小、路由规则、协议和支持的流量模式。查看可用的联网解决方案，并确定哪种服务与您的工作负载联网特性相符。基于云的网络可以快速重建，因此有必要随着时间的推移改进网络架构，以提高性能效率。

实施步骤：

- 定义和记录网络性能要求，包括网络延迟、带宽、协议、位置、流量模式（峰值和频率）、吞吐量、加密、检查和路由规则等指标。
- 了解关键 AWS 联网服务，例如 VPC、AWS Direct Connect、弹性负载均衡（ELB）以及 Amazon Route 53。
- 捕获以下关键网络特性：

特性	工具和指标
基础网络特性	• VPC 流日志 • AWS Transit Gateway 流日志 • AWS Transit Gateway 指标 • AWS PrivateLink 指标
应用程序网络特性	• Elastic Fabric Adapter • AWS App Mesh 指标 • Amazon API Gateway 指标
边缘网络特性	• Amazon CloudFront 指标 • Amazon Route 53 指标 • AWS Global Accelerator 指标

特性	工具和指标
混合网络特性	• Direct Connect 指标 • AWS Site-to-Site VPN 指标 • AWS Client VPN 指标 • AWS Cloud WAN 指标
安全网络特性	• AWS Shield、AWS WAF 和 AWS Network Firewall 指标
跟踪特性	• AWS X-Ray • VPC Reachability Analyzer • 网络访问分析器 • Amazon Inspector – • Amazon CloudWatch RUM

- 对网络性能进行基准测试和其他测试：
  - 对网络吞吐量进行基准测试，因为当实例位于同一 VPC 中时，一些因素可能会影响 Amazon EC2 网络性能。测量同一 VPC 中的 Amazon EC2 Linux 实例之间的网络带宽。
  - 执行负载测试以试用各种联网解决方案和选项。

资源

相关文档：

- 应用程序负载均衡器
- Linux EC2 上的增强联网功能
- Windows EC2 上的增强联网功能
- EC2 置放群组
- 在 Linux 实例上启用弹性网络适配器（ENA）增强联网功能
- 网络负载均衡器
- AWS 联网产品
- Transit Gateway
- Transitioning to latency-based routing in Amazon Route 53

- VPC 端点

相关视频：

- AWS re:Invent 2023 - AWS networking foundations
- AWS re:Invent 2023 - What can networking do for your application?
- AWS re:Invent 2023 - Advanced VPC designs and new capabilities
- AWS re:Invent 2023 - A developer's guide to cloud networking
- AWS re:Invent 2019 - Connectivity to AWS and hybrid AWS network architectures
- AWS re:Invent 2019 - Optimizing Network Performance for Amazon EC2 Instances
- AWS Summit Online - Improve Global Network Performance for Applications
- AWS re:Invent 2020 - Networking best practices and tips with the Well-Architected Framework
- AWS re:Invent 2020 - AWS networking best practices in large-scale migrations

相关示例：

- AWS Transit Gateway and Scalable Security Solutions
- AWS Networking 讲习会
- Hands-on Network Firewall 讲习会
- Observing and Diagnosing your Network on AWS
- Finding and addressing Network Misconfigurations on AWS

PERF04-BP02 评估可用的联网功能

评估云中可能提高性能的联网功能。借助测试、指标和分析来衡量这些功能的影响。例如，利用可用的网络级功能来减少延迟、网络距离或抖动。

常见反模式：

- 一直待在一个区域，因为这是总部实际所在的区域。
- 使用防火墙而不是安全组来过滤流量。
- 中断 TLS 来进行流量检查，而不是依赖安全组、端点策略和其他云原生功能。
- 只使用基于子网的分段，而不是安全组。

建立此最佳实践的好处：评估所有服务功能和选项可以提高您的工作负载性能，降低基础设施的成本，减少维护工作负载所需的工作量，并提升您的整体安全态势。您可以利用 AWS 的全球主干网，为客户提供出色的联网体验。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

AWS 提供有助于提高网络性能的 AWS Global Accelerator 和 Amazon CloudFront 等服务，而大多数 AWS 服务都具有用于优化网络流量的产品功能（例如 Amazon S3 Transfer Acceleration 功能）。

查看您可以使用哪些与网络相关的配置选项，以及这些配置选项对您的工作负载有何影响。要想优化性能，需要了解这些选项如何与您的架构进行交互，以及它们将对测得的性能和用户体验产生的影响。

实施步骤

- 创建工作负载组件列表。
  - 在构建统一全球网络时，考虑使用 AWS Cloud WAN 来构建、管理和监控您组织的网络。
  - 使用 Amazon CloudWatch Logs 指标监控您的全球与核心网络。利用 Amazon CloudWatch RUM，它提供了有助于识别、理解和增强用户的数字体验的见解。
  - 查看 AWS 区域 和可用区之间以及每个可用区内的聚合网络延迟，使用 AWS Network Manager 深入了解应用程序性能与 AWS 底层网络性能的关系。
  - 使用现有的配置管理数据库（CMDB）工具或 AWS Config 等服务创建工作负载清单及其配置方式。
- 如果这是一个现有的工作负载，请确定并记录性能指标的基准，重点关注瓶颈和需要改进之处。受业务要求和工作负载特性的影响，与性能相关的联网指标会因工作负载而异。首先，对于您的工作负载，检查带宽、延迟、数据包丢失、抖动和重传等指标可能很重要。
- 如果这是新的工作负载，请执行负载测试来确定性能瓶颈。
- 对于识别的性能瓶颈，请查看解决方案的配置选项，以确定性能改进机会。查看以下主要联网选项和功能：

改进机会	解决方案
网络路径或路由	使用网络访问分析器来确定路径或路由。
网络协议	请参阅 PERF04-BP05 选择网络协议以提高性能。

改进机会	解决方案
网络拓扑	当连接多个账户时，请评估 VPC 对等连接与 AWS Transit Gateway 之间的运营和性能权衡。AWS Transit Gateway 可简化所有 VPC 之间的互连，这些 VPC 可以跨越数千个 AWS 账户 并接入您的本地网络。使用 AWS Resource Access Manager 在多个账户之间共享您的 AWS Transit Gateway。 请参阅 PERF04-BP03 为工作负载选择合适的专用连接或 VPN。
网络服务	AWS Global Accelerator 是一项网络服务，使用 AWS 全球网络基础设施，可将用户流量的性能提高多达 60%。 Amazon CloudFront 可在全球范围内提高工作负载内容分发性能并减少延迟。 使用 Lambda@Edge 运行一些函数，这些函数可自定义 CloudFront 在离用户更近的位置提供的内容、减少延迟并提高性能。 Amazon Route 53 提供基于延迟的路由、地理位置路由、地理位置邻近度路由和基于 IP 的路由选项，有助于提高面向全球受众的工作负载性能。如果工作负载分布在全球，通过查看工作负载流量和用户位置，确定哪种路由选项可以优化工作负载性能。

改进机会	解决方案
存储资源功能	Amazon S3 Transfer Acceleration 功能可让外部用户在向 Amazon S3 传输数据时通过 CloudFront 的网络优化获益。这就提高了将大量数据从没有专用连接的远程位置传输到 AWS Cloud 的能力。 Amazon S3 多区域接入点将内容复制到多个区域，并通过提供一个接入点简化了工作负载。使用多区域接入点时，您可以使用标识最低延迟存储桶的服务向 Amazon S3 请求或写入数据。
计算资源功能	Amazon EC2 实例、容器和 Lambda 函数使用的弹性网络接口（ENI）按流进行限制。查看置放群组以优化 EC2 联网吞吐量。为避免每个流上出现瓶颈，请将应用程序设计为使用多个流。要监控和查看与计算相关的联网指标，请使用 CloudWatch 指标和 ethtool。`ethtool` 命令包含在 ENA 驱动程序中，并公开了其他与网络相关的指标，这些指标可作为自定义指标发布到 CloudWatch。 Amazon 弹性网络适配器（ENA）为集群置放群组中的实例提供更大的网络吞吐量，实现进一步优化。 Elastic Fabric Adapter（EFA）是 Amazon EC2 实例的网络接口，使您能够在 AWS 上运行要求大规模高级别节点间通信的应用程序。 Amazon EBS 优化实例使用经过优化的配置堆栈，可以针对增加的 Amazon EBS I/O 提供额外的专用容量。

资源

相关文档：

- 应用程序负载均衡器
- Linux EC2 上的增强联网功能
- Windows EC2 上的增强联网功能
- EC2 置放群组
- 在 Linux 实例上启用弹性网络适配器（ENA）增强联网功能
- 网络负载均衡器
- 联网产品AWS
- Transitioning to Latency-Based Routing in Amazon Route 53
- VPC 端点
- （、、VPC 流日志）

相关视频：

- AWS re:Invent 2023 – Ready for what's next? Designing networks for growth and flexibility
- AWS re:Invent 2023 – Advanced VPC designs and new capabilities
- AWS re:Invent 2023 – A developer's guide to cloud networking
- AWS re:Invent 2022 – Dive deep on AWS networking infrastructure
- AWS re:Invent 2019 – Connectivity to AWS and hybrid AWS network architectures
- AWS re:Invent 2018 – Optimizing Network Performance for Amazon EC2 Instances
- AWS Global Accelerator

相关示例：

- AWS Transit Gateway and Scalable Security Solutions
- AWS Networking 讲习会
- Observing and diagnosing your network
- Finding and addressing network misconfigurations on AWS

PERF04-BP03 为工作负载选择合适的专用连接或 VPN

当需要混合连接来连接本地资源和云资源时，请预置足够的带宽以满足您的性能要求。估算混合工作负载的带宽和延迟要求。这些数字将推动您的规模需求。

常见反模式：

- 仅根据网络加密要求评估 VPN 解决方案。
- 不评估备用或冗余连接选项。
- 没有确定全部工作负载要求（加密、协议、带宽和流量需求）。

建立此最佳实践的好处：选择和配置适当的连接解决方案将会提高工作负载的可靠性，并最大限度地提高性能。通过确定工作负载要求、提前规划和评估混合解决方案，您可以最大限度地减少成本高昂的物理网络变更和运营开销，同时加快实现价值的速度。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

根据带宽要求开发混合网络架构。可使用 Direct Connect 将本地网络与 AWS 私密地连接。这适用于需要高带宽、低延迟，同时实现一致性能的情况。VPN 连接通过互联网建立安全连接。在以下情况下可使用 VPN 连接：只需要临时连接、需要考虑成本因素，或者在使用 Direct Connect 的情况下等待建立弹性物理网络连接时作为应急措施。

如果您的带宽要求很高，则可以考虑使用多种 Direct Connect 或 VPN 服务。可以在服务之间对流量进行负载平衡，但由于延迟和带宽差异，我们不建议在 Direct Connect 和 VPN 之间进行负载平衡。

实施步骤

- 估计现有应用程序的带宽和延迟要求。
  - 对于迁移到 AWS 的现有工作负载，利用来自内部网络监控系统的数据。
  - 对于新工作负载或您没有监控数据的现有工作负载，请咨询产品所有者，以确定足够的性能指标并提供良好的用户体验。
- 选择专用连接或 VPN 作为连接选项。根据所有工作负载要求（加密、带宽和流量需求），您可以选择 AWS Direct Connect 或 Site-to-Site VPN（或两者）。下图可协助您选择适当的连接类型。
  - AWS Direct Connect 使用专用连接或托管连接，提供指向 AWS 环境的专用连接，速度从 50 Mbps 到 100 Gbps 不等。这样一来，延迟得到管理和控制，并且拥有预置带宽，让您的工作负载

能够高效地连接到其他环境。使用 AWS Direct Connect 合作伙伴，您可以从多个环境获得端到端连接，从而提供具有一致性能的扩展网络。AWS 使用原生 100 Gbps、链接聚合组（LAG）或 BGP 同等成本多路径（ECMP）提供扩展 Direct Connect 连接带宽。

  - AWS Site-to-Site VPN 提供支持互联网协议安全（IPsec）的托管服务。创建 VPN 连接时，每个 VPN 连接包括两条隧道以实现高可用性。
- 按照 AWS 文档选择合适的连接选项：
  - 如果决定使用 Direct Connect，请为连接选择合适的带宽。
  - 如果要在多个位置使用 AWS Site-to-Site VPN 连接到 AWS 区域，则使用加速 Site-to-Site VPN 连接，以便有机会提高网络性能。
  - 如果网络设计包含通过 AWS Direct Connect 进行 IPsec VPN 连接，则考虑使用私有 IP VPN 来提高安全性并实现分段。AWS私有 IP Site-to-Site VPN 部署在中转虚拟接口（VIF）上。
  - AWS Direct Connect SiteLink 通过绕过 AWS 区域 在 AWS Direct Connect 位置之间以最快路径发送数据，从而在全球的数据中心之间创建低延迟和冗余连接。
- 在部署到生产环境之前，验证您的连接设置。执行安全和性能测试，确保其满足您的带宽、可靠性、延迟和合规性要求。
- 定期监控您的连接性能和使用情况，并在需要时进行优化。

确定性性能流程图

资源

相关文档：

- AWS 联网产品
- AWS Transit Gateway
- VPC 端点
- 构建可扩展的安全多 VPC AWS 网络基础结构
- Client VPN

相关视频：

- AWS re:Invent 2023 – Building hybrid network connectivity with AWS

- AWS re:Invent 2023 – Secure remote connectivity to AWS
- AWS re:Invent 2022 – Optimizing performance with Amazon CloudFront
- AWS re:Invent 2019 – Connectivity to AWS and hybrid AWS network architectures
- AWS re:Invent 2020 – AWS Transit Gateway Connect

相关示例：

- AWS Transit Gateway and Scalable Security Solutions
- AWS Networking 讲习会

PERF04-BP04 使用负载平衡在多个资源之间分配流量

跨多个资源或服务分配流量，以便让工作负载能够利用云提供的弹性。您也可以使用负载均衡机制来分流加密终端，以便提高性能和可靠性，并有效管理和路由流量。

常见反模式：

- 在选择负载均衡器类型时不考虑工作负载要求。
- 不利用负载均衡器功能进行性能优化。
- 工作负载直接暴露给互联网，而不使用负载均衡器。
- 通过现有负载均衡器来路由所有互联网流量。
- 使用通用 TCP 负载均衡，并让每个计算节点处理 SSL 加密。

建立此最佳实践的好处：负载均衡器可在单个可用区内或多个可用区之间处理应用程序不断变化的流量负载，并实现高可用性、自动扩展和更高的工作负载利用率。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

负载均衡器充当工作负载的接入点，从这里将流量分发到后端目标（例如计算实例或容器）来提高利用率。

优化架构的第一步是选择适合的负载均衡器类型。首先列出工作负载特性，例如协议（如 TCP、HTTP、TLS 或 WebSocket）、目标类型（如实例、容器或无服务器）、应用程序要求（如长时间运行的连接、用户身份验证或粘性）和置放（如区域、Local Zone、Outpost 或分区隔离）。

AWS 为应用程序提供多种模型来使用负载均衡。应用程序负载均衡器最适合 HTTP 和 HTTPS 流量的负载均衡，面向交付包括微服务和容器在内的现代应用程序架构，提供高级请求路由功能。

若要对需要极高性能的 TCP 流量进行负载均衡，网络负载均衡器是最佳选择。网络负载均衡器每秒能够处理数百万请求，同时能保持超低延迟，还针对处理突发和不稳定的流量模式进行了优化。

弹性负载均衡提供集成的证书管理和 SSL/TLS 解密，使您可以灵活地集中管理负载均衡器的 SSL 设置，并从工作负载中分流占用大量 CPU 的工作。

选择适合的负载均衡器之后，您可以开始利用其功能来减少后端为提供流量所付出的工作量。

例如，您可以使用应用程序负载均衡器（ALB）和网络负载均衡器（NLB）执行 SSL/TLS 加密分流，借此机会避免目标完成 CPU 密集型 TLS 握手，同时还可以改进证书管理。

当您在负载均衡器中配置 SSL/TLS 分流时，它负责加密进出客户端的流量，同时将未加密的流量传输到您的后端，释放后端资源和缩短客户端的响应时间。

应用程序负载均衡器还可以提供 HTTP/2 流量，无需在您的目标上支持它。因为 HTTP/2 可以更高效地使用 TCP 连接，所以这个简单的决定可以缩短应用程序的响应时间。

在定义架构时应考虑工作负载延迟要求。例如，如果您有延迟敏感型应用程序，则可以决定使用提供极低延迟的网络负载均衡器。您也可以决定通过在 AWS Local Zones 甚至 AWS Outposts 中利用应用程序负载均衡器，让工作负载更接近客户。

延迟敏感型工作负载的另一个考虑因素是跨可用区负载均衡。借助跨可用区负载均衡，每个负载均衡器节点在所有允许的可用区中的已注册目标之间分配流量。

使用与负载均衡器集成的自动扩缩功能。高效性能系统的其中一个关键方面与合理调整后端资源的规模有关。为此，您可以为后端目标资源使用负载均衡器集成。使用负载均衡器与自动扩缩组的集成，根据需要在负载均衡器中添加或删除目标，以应对传入流量。对于容器化工作负载，负载均衡器也可与 Amazon ECS 和 Amazon EKS 集成。

- Amazon ECS – 服务负载均衡器
- Amazon EKS 上的应用程序负载均衡
- Amazon EKS 上的网络负载均衡器

实施步骤

- 定义您的负载均衡要求，包括流量、可用性和应用程序可扩展性。
- 为您的应用程序选择正确的负载均衡器类型。

  - 为 HTTP/HTTPS 工作负载使用应用程序负载均衡器。
  - 为在 TCP 或 UDP 上运行的非 HTTP 工作负载使用网络负载均衡器。
  - 若想利用这两种产品的功能，请将两者结合使用（ALB 作为 NLB 的目标）。例如，若想将 NLB 的静态 IP 与 ALB 基于 HTTP 标头的路由结合使用，或者想将 HTTP 工作负载向 AWS PrivateLink 公开，就可以这样做。
  - 有关负载均衡器的完整比较，请参阅 ELB 产品比较。
- 如果可能，请使用 SSL/TLS 分流。
  - 配置 HTTPS/TLS 侦听器，同时使用集成了 AWS Certificate Manager 的应用程序负载均衡器和网络负载均衡器。
  - 请注意，出于合规性原因，有些工作负载可能需要端到端加密。在这种情况下，必须允许在目标上启用加密。
  - 有关安全最佳实践，请参阅 SEC09-BP02 执行传输中加密。
- 选择适合的路由算法（仅限 ALB）。
  - 路由算法会影响后端目标的使用情况，从而决定它们对性能的影响。例如，ALB 提供了两个路由算法选项：
  - 最少未完成请求：用于在应用程序的请求复杂程度不同或目标处理能力不同的情况下，实现更好的后端目标负载分布。
  - 轮询：当请求和目标类似，或需要在目标之间平均分配请求时使用。
- 考虑跨可用区或分区隔离。
  - 使用关闭的跨可用区（分区隔离）来改善延迟和分区故障域。在 NLB 中，该选项默认处于关闭状态；在 ALB 中，该选项可按目标组关闭。
  - 使用开启的跨可用区来提高可用性和灵活性。在 ALB 中，该选项默认处于打开状态；在 NLB 中，该选项可按目标组开启。
- 为 HTTP 工作负载开启 HTTP 保持活动（仅限 ALB）。借助此功能，在保持活动超时到期之前，负载均衡器可以重复使用后端连接，从而改进 HTTP 请求和缩短响应时间，还可以降低后端目标的资源利用率。有关如何为 Apache 和 Nginx 执行此操作的详细信息，请参阅使用 Apache 或 NGINX 作为 ELB 后端服务器的最佳设置是什么？
- 为您的负载均衡器开启监控。
  - 打开应用程序负载均衡器和网络负载均衡器的访问日志。
  - 主要考虑 ALB 的 `request_processing_time`、`request_processing_time` 和 `response_processing_time`。
  - 主要考虑 NLB 的 `connection_time` 和 `tls_handshake_time`。

- 准备好在需要时查询日志。可以使用 Amazon Athena 来查询 ALB 日志和 NLB 日志。
- 为性能相关指标（例如 ALB 的 `TargetResponseTime`）创建警报。

资源

相关文档：

- ELB 产品比较
- AWS 全球基础设施
- Improving Performance and Reducing Cost Using Availability Zone Affinity
- Step by step for Log Analysis with Amazon Athena
- 查询应用程序负载均衡器日志
- Monitor your Application Load Balancers
- Monitor your Network Load Balancer
- 使用弹性负载均衡跨自动扩缩组中的实例分配流量

相关视频：

- AWS re:Invent 2023: What can networking do for your application?
- AWS re:Inforce 2022: How to use Elastic Load Balancing to enhance your security posture at scale
- AWS re:Invent 2018: Elastic Load Balancing: Deep Dive and Best Practices
- AWS re:Invent 2021 - How to choose the right load balancer for your AWS workloads
- AWS re:Invent 2019: Get the most from Elastic Load Balancing for different workloads

相关示例：

- Gateway Load Balancer
- 使用 Amazon Athena 进行日志分析的 CDK 和 CloudFormation 示例

PERF04-BP05 选择网络协议以提高性能

根据对工作负载性能的影响，做出有关系统与网络之间的通信协议的决策。

延迟和带宽之间的关系可以实现高吞吐量。如果文件传输使用传输控制协议（TCP），则延迟越高，整体吞吐量很可能越低。有一些方法可以使用 TCP 调整和优化的传输协议来解决此问题，但一种解决方案是使用用户数据报协议（UDP）。

常见反模式：

- 无论有怎样的性能要求，您都可以为所有工作负载使用 TCP。

建立此最佳实践的好处：确认已为用户和工作负载组件之间的通信使用适当的协议，有助于改善应用程序的整体用户体验。例如，无连接 UDP 虽然允许较高速度，但不提供重新传输或高可靠性。TCP 虽然是一个功能全面的协议，但它在处理这些数据包时需要较高的开销。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

如果您能够为应用程序选择不同的协议，并且具有该领域的专业知识，请使用不同的协议来优化您的应用程序和最终用户体验。请注意，这种方法难度很大，只有在先用其他方法优化了应用程序后才能尝试。

提高工作负载性能的主要考虑因素是了解延迟和吞吐量要求，然后选择可优化性能的网络协议。

何时考虑使用 TCP

TCP 提供可靠的数据传输，并可用于工作负载组件之间的通信，在这种情况下，可靠性和有保证的数据传输很重要。许多基于 Web 的应用程序依赖基于 TCP 的协议（例如 HTTP 和 HTTPS）来打开 TCP 套接字，以便在应用程序组件之间进行通信。电子邮件和文件数据传输也是使用 TCP 的常见应用，因为它是应用程序组件之间简单可靠的传输机制。在 TCP 之上使用 TLS 会增加一些通信开销，进而会导致延迟增加和吞吐量降低，但它也有安全方面的优势。该开销主要来自握手过程（需要多次往返才能完成）的额外开销。握手完成后，加密和解密数据的开销相对较小。

何时考虑使用 UDP

UDP 是一种面向无连接的协议，因此适用于需要快速、高效传输的应用，例如日志、监控和 VoIP 数据。此外，如果您的工作负载组件要响应来自大量客户端的小型查询，以确保工作负载实现最佳性能，则可考虑使用 UDP。数据报传输层安全性（DTLS）是传输层安全性协议（TLS）的 UDP 等效项。在 UDP 上使用 DTLS 时，因为简化了握手过程，所以开销来自加密和解密数据。因为 DTLS 包括额外的字段，用于指明安全性参数和检测篡改，所以它也会给 UDP 数据包增加少量的开销。

何时考虑使用 SRD

可扩展的可靠数据报（SRD）是一种针对高吞吐量工作负载而优化的网络传输协议，因为它能够跨多条路径对流量进行负载均衡，并能在发生丢包或链路故障时快速恢复。因此，SRD 非常适合在计算节点之间需要高吞吐量、低延迟通信的高性能计算（HPC）工作负载。这可能包括并行处理任务（例如，涉及在节点间进行大量数据传输的模拟、建模和数据分析）。

实施步骤

- 使用 AWS Global Accelerator 和 AWS Transfer Family 服务提高在线文件传输应用程序的吞吐量。AWS Global Accelerator 服务帮助您在客户端设备与 AWS 上的工作负载之间实现更低延迟。借助 AWS Transfer Family，您可以使用基于 TCP 的协议 [安全外壳文件传输协议（SFTP）和基于 SSL 的文件传输协议（FTPS）] 安全地扩展和管理发送到 AWS 存储服务的文件传输。
- 使用网络延迟来确定 TCP 是否适合工作负载组件之间的通信。如果客户端应用程序和服务器之间的网络延迟很高，则 TCP 三次握手需要一些时间，因而会影响应用程序的响应能力。可以使用第一个字节的时间（TTFB）和往返时间（RTT）等指标来衡量网络延迟。如果工作负载向用户提供动态内容，则考虑使用 Amazon CloudFront，因为该服务会为动态内容建立到每个源的持久连接，以便减少连接设置时间，避免减慢每个客户端请求的速度。
- 由于在 TCP 或 UDP 上使用 TLS 会影响加密和解密，从而导致工作负载的延迟增加和吞吐量降低。对于此类工作负载，请考虑使用弹性负载均衡上的 SSL/TLS 分流，使负载均衡器能够处理 SSL/TLS 加密和解密过程，而不是让后端实例来处理，从而提高工作负载性能。这可以帮助降低后端实例上的 CPU 利用率，进而可以提高性能并增加容量。
- 使用网络负载均衡器（NLB）来部署依赖 UDP 协议的服务（例如，身份验证和授权、日志记录、DNS、IoT 和串流媒体），以便提高工作负载的性能和可靠性。NLB 在多个目标之间分配传入的 UDP 流量，使您可以横向扩展工作负载、提高容量和减少单个目标的开销。
- 对于高性能计算（HPC）工作负载，可以考虑使用弹性网络适配器（ENA）Express 功能。该功能使用 SRD 协议，通过为 EC2 实例之间的网络流量提供更高的单流带宽（25 Gbps）和更低的尾部延迟（99.9%）来提高网络性能。
- 使用应用程序负载均衡器（ALB）对工作负载组件之间或 gRPC 客户端与服务之间的 gRPC（远程过程调用）流量进行路由和负载均衡。gRPC 使用基于 TCP 的 HTTP/2 协议进行传输，也可带来性能优势，例如更少的网络占用、压缩、高效的二进制序列化、支持众多语言以及双向流式传输。

资源

相关文档：

- How to route UDP traffic into Kubernetes
- 应用程序负载均衡器

- Linux EC2 上的增强联网功能
- Windows EC2 上的增强联网功能
- EC2 置放群组
- 在 Linux 实例上启用弹性网络适配器（ENA）增强联网功能
- 网络负载均衡器
- AWS 联网产品
- Transitioning to Latency-Based Routing in Amazon Route 53
- VPC 端点

相关视频：

- AWS re:Invent 2022 – Scaling network performance on next-gen Amazon Elastic Compute Cloud instances
- AWS re:Invent 2022 – Application networking foundations

相关示例：

- AWS Transit Gateway and Scalable Security Solutions
- AWS Networking 讲习会

PERF04-BP06 根据网络要求选择工作负载的位置

评估资源置放选项，以便减少网络延迟和提高吞吐量，通过缩短页面加载和数据传输时间来提供最佳的用户体验。

常见反模式：

- 将所有工作负载资源整合到一个地理位置中。
- 选择的是离自己位置最近的区域，而不是离工作负载最终用户最近的区域。

建立此最佳实践的好处：用户与应用程序之间的延迟会极大地影响用户体验。通过使用适当的 AWS 区域和 AWS 专用全球网络，您可以减少延迟，为远程用户提供更好的体验。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

Amazon EC2 实例等资源被放置在 AWS 区域、AWS Local Zones、AWS Outposts 或 AWS Wavelength 区域内的可用区中。选择此位置会影响给定用户位置的网络延迟和吞吐量。Amazon CloudFront 和 AWS Global Accelerator 等边缘服务也可用于在边缘站点缓存内容或为用户提供通过 AWS 全球网络到达工作负载的最佳路径，从而提高网络性能。

Amazon EC2 为联网提供置放群组。置放群组是实例的逻辑分组，可以减少延迟。使用具有支持的实例类型和弹性网络适配器（ENA）的置放群组，可使工作负载参与低延迟、低抖动的 25Gbps 网络。建议将置放群组用于可受益于低网络延迟和/或高网络吞吐量的工作负载。

延迟敏感型服务使用 AWS 全球网络在边缘站点交付，例如 Amazon CloudFront。这些边缘站点通常提供内容分发网络（CDN）和域名系统（DNS）等服务。通过在边缘交付这些服务，工作负载可以低延迟响应内容或 DNS 解析请求。这些服务还提供地理定位服务，例如内容地理定位（基于最终用户位置提供不同内容），或基于延迟的路由（将最终用户引导至最近的区域以实现最小延迟）。

可使用边缘服务来减少延迟并启用内容缓存。为 DNS 和 HTTP/HTTPS 正确配置缓存控制，以便通过这些方式获得最大优势。

实施步骤

- 捕获有关传入和传出网络接口的 IP 流量的信息。
  - 使用 VPC 流日志记录 IP 流量
  - 如何将客户端 IP 地址保留在 AWS Global Accelerator 中
- 分析工作负载中的网络访问模式，以便确定用户如何使用应用程序。
  - 使用 Amazon CloudWatch 和 AWS CloudTrail 等监控工具收集有关网络活动的数据。
  - 分析数据以确定网络访问模式。
- 请根据以下关键元素，为您的工作负载部署选择区域：
  - 数据所在位置：对于数据密集型应用程序（如大数据和机器学习），应用程序代码的运行应尽量接近数据。
  - 用户所在位置：对于面向用户的应用程序，选择接近您工作负载用户的一个或多个区域。
  - 其他约束：考虑成本和合规性等约束因素，如 What to Consider when Selecting a Region for your Workloads 中所述。
- 使用 AWS Local Zones 运行视频渲染等工作负载。Local Zones 使计算和存储资源更接近终端用户，从而使您受益。

- 将 AWS Outposts 用于需要保留在本地的工作负载，在此您希望该工作负载与 AWS 中的其他工作负载一起无缝运行。
- 高分辨率实时视频流、高保真度音频和增强现实或虚拟现实（AR/VR）等应用要求 5G 设备具有超低延迟。对于此类应用程序，请考虑使用 AWS Wavelength。AWS Wavelength 在 5G 网络内嵌入 AWS 计算和存储服务，为开发、部署和扩展超低延迟应用提供移动边缘计算基础设施。
- 对常用资产使用本地缓存或 AWS 缓存解决方案，以提高性能，减少数据移动并减小对环境的影响。

服务	何时使用
Amazon CloudFront	用于缓存静态内容（如图像、脚本和视频）以及动态内容（如 API 响应或 Web 应用程序）。
Amazon ElastiCache	用于缓存 Web 应用程序的内容。
DynamoDB Accelerator	用于将内存中加速添加到 DynamoDB 表。

- 使用有助于您在更接近工作负载用户的位置运行代码的服务，例如：

服务	何时使用
Lambda@Edge	用于执行计算密集型操作，当对象不在缓存中时启动这些操作。
Amazon CloudFront Functions	用于处理简单应用场景，如 HTTP(S) 请求或响应操作，这些操作可由短期运行的函数启动。
AWS IoT Greengrass	用于为互联设备运行本地计算、消息收发和数据缓存。

- 有些应用程序需要固定入口点，或通过减少第一个字节延迟和抖动以及提高吞吐量来提高性能。在边缘站点提供静态任播 IP 地址和 TCP 终止的联网服务可以让这些应用程序受益。AWS Global Accelerator 可以将应用程序的性能提高多达 60%，并为多区域架构提供快速失效转移。AWS Global Accelerator 提供静态任播 IP 地址，可作为一个或多个 AWS 区域 中托管的应用程序的固定入口点。这些 IP 地址使流量在尽可能靠近用户的位置进入 AWS 全球网络。AWS Global Accelerator 在客户端和最靠近客户端的 AWS 边缘站点之间建立 TCP 连接，从而缩短初始连接设置时间。检查 AWS Global Accelerator 的使用情况，以提高 TCP/UDP 工作负载的性能，并为多区域架构提供快速失效转移。

资源

相关最佳实践：

- COST07-BP02 根据成本选择区域
- COST08-BP03 实施服务以便降低数据传输成本
- REL10-BP01 将工作负载部署到多个位置
- REL10-BP02 为多位置部署选择合适的位置
- SUS01-BP01 根据业务要求和可持续性目标选择区域
- SUS02-BP04 根据其联网要求优化工作负载的地理位置
- SUS04-BP07 最大限度地减少跨网络的数据移动

相关文档：

- AWS 全球基础设施
- AWS Local Zones and AWS Outposts, choosing the right technology for your edge workload
- 置放群组
- AWS Local Zones
- AWS Outposts
- AWS Wavelength
- Amazon CloudFront
- AWS Global Accelerator
- AWS Direct Connect
- AWS Site-to-Site VPN
- Amazon Route 53

相关视频：

- AWS Local Zones Explainer Video
- AWS Outposts: Overview and How it Works
- AWS re:Invent 2023 - A migration strategy for edge and on-premises workloads
- AWS re:Invent 2021 - AWS Outposts: Bringing the AWS experience on premises
- AWS re:Invent 2020: AWS Wavelength: Run apps with ultra-low latency at 5G edge

- AWS re:Invent 2022 - AWS Local Zones: Building applications for a distributed edge
- AWS re:Invent 2021 - Building low-latency websites with Amazon CloudFront
- AWS re:Invent 2022 - Improve performance and availability with AWS Global Accelerator
- AWS re:Invent 2022 - Build your global wide area network using AWS
- AWS re:Invent 2020: Global traffic management with Amazon Route 53

相关示例：

- AWS Global Accelerator Custom Routing 讲习会
- Handling Rewrites and Redirects using Edge Functions

PERF04-BP07 根据指标优化网络配置

使用收集和分析的数据做出有关优化网络配置的明智决策。

常见反模式：

- 认为所有性能相关的问题都与应用程序有关。
- 只从距离已部署工作负载很近的位置测试网络性能。
- 为所有网络服务使用默认配置。
- 过度预置网络资源来提供充足的容量。

建立此最佳实践的好处：收集 AWS 网络的必要指标并实施网络监控工具，使您可以了解网络性能和优化网络配置。

在未建立这种最佳实践的情况下暴露的风险等级：低

实施指导

监控进出 VPC、子网或网络接口的流量，这对于了解如何利用 AWS 网络资源以及如何优化网络配置至关重要。通过使用以下 AWS 网络工具，您可以进一步检查有关流量使用、网络访问和日志的信息。

实施步骤

- 确定要收集的关键性能指标，例如延迟或丢包。AWS 提供了多种工具，可以协助您收集这些指标。通过使用以下工具，您可以进一步检查有关流量使用、网络访问和日志的信息：

AWS 工具	使用情形
Amazon VPC IP 地址管理器	使用 IPAM 来规划、跟踪和监控 AWS 与本地工作负载的 IP 地址。这是优化 IP 地址使用和分配的最佳实践。
VPC 流日志	使用 VPC 流日志来捕获有关进出 VPC 中网络接口的流量的详细信息。借助 VPC 流日志，您可以诊断过于严格或过于宽松的安全组规则，并确定进出网络接口的流量的方向。
AWS Transit Gateway 流日志	使用 AWS Transit Gateway 流日志捕获有关进出中转网关的 IP 流量的信息。
DNS 查询日志记录	有关 Route 53 收到的公有或私有 DNS 查询的日志信息。借助 DNS 日志，您可以了解请求的域或子域，或了解响应 DNS 查询的 Route 53 边缘站点，从而优化 DNS 配置。
Reachability Analyzer	Reachability Analyzer 有助于分析并调试网络可达性。作为一种配置分析工具，Reachability Analyzer 能够在 VPC 中的源资源和目标资源之间执行连接测试。此工具可帮助确认网络配置是否符合预期连接。
网络访问分析器	您可以使用网络访问分析器来了解对资源的网络访问。您可以使用网络访问分析器来指定网络访问需求，然后确定不能满足指定要求的潜在网络路径。通过优化相应的网络配置，您可以了解和验证网络的状态，并证明 AWS 中的网络满足您的合规性要求。

AWS 工具	使用情形
Amazon CloudWatch	使用 Amazon CloudWatch 并启用适当的网络选项指标。确保为工作负载选择适合的网络指标。例如，您可以为 VPC 网络地址使用、VPC NAT Gateway、AWS Transit Gateway、VPN 隧道、AWS Network Firewall、弹性负载均衡和 AWS Direct Connect 启用指标。为了观测和了解网络状态和使用情况，以及便于您根据观测结果优化网络配置，持续监控指标是一种好方法。
AWS Network Manager	为了实现运营和规划目的，您可以使用 AWS Network Manager 监控 AWS 全球网络的实时和历史性能。Network Manager 提供 AWS 区域 和可用区之间以及每个可用区内的聚合网络延迟，让您能够更好地了解应用程序性能与 AWS 底层网络性能的关系。
Amazon CloudWatch RUM	使用 Amazon CloudWatch RUM 收集指标，以便为您提供洞察，协助您识别、理解和改善用户体验。

- 使用 VPC 和 AWS Transit Gateway 流日志识别主要贡献者和应用程序流量模式。
- 评测和优化您当前的网络架构，包括 VPC、子网和路由。例如，您可以评估不同的 VPC 对等互连或 AWS Transit Gateway 如何让您能够改善架构中的联网。
- 评测网络中的路由路径，以验证目的地之间是否始终使用最短路径。网络访问分析器可以帮助实现此目的。

资源

相关文档：

- Public DNS query logging
- 什么是 IPAM？
- What is Reachability Analyzer?
- What is Network Access Analyzer?

- VPC 的 CloudWatch 指标
- Optimize performance and reduce costs for network analytics with VPC Flow Logs in Apache Parquet format
- Monitoring your global and core networks with Amazon CloudWatch metrics
- Continuously monitor network traffic and resources

相关视频：

- AWS re:Invent 2023 – A developer's guide to cloud networking
- AWS re:Invent 2023 – Ready for what’s next? Designing networks for growth and flexibility
- AWS re:Invent 2023 – Advanced VPC designs and new capabilities
- AWS re:Invent 2022 – Dive deep on AWS networking infrastructure
- AWS re:Invent 2020 – Networking best practices and tips with the AWS Well-Architected Framework
- AWS re:Invent 2020 – Monitoring and troubleshooting network traffic

相关示例：

- AWS Networking 讲习会
- AWS Network Monitoring
- Observing and diagnosing your network on AWS
- Finding and addressing network misconfigurations on AWS

流程和文化

在最初构建工作负载时，您可以采用一些原则和实践，协助您更好地运行高效、高性能的云工作负载。该重点领域提供的最佳实践有助于您采用能提高云工作负载性能效率的文化。

要打造这种文化，请考虑以下关键原则：

- 基础设施即代码：使用 AWS CloudFormation 模板之类的方法定义您的基础设施即代码。使用模板，您可以将基础设施与应用程序代码和配置一道放入源代码控制中。这让您能够将用于开发软件的实践应用到基础设施，从而能够快速迭代。
- 部署管道：使用持续集成/连续部署（CI/CD）管道（例如，源代码存储库、构建系统、部署和测试自动化）来部署基础设施。这让您能够以可重复、一致且低成本的方式进行迭代部署。
- 明确定义的指标：设置和监控指标以捕获关键性能指标（KPI）。我们建议您使用技术和业务指标。网站或移动应用程序的关键指标是首个字节捕获时间或渲染时间。其他常规的适用指标包括线程计数、垃圾回收速率以及等待状态。业务指标，如单次请求累计总成本，可以提醒您留意降低成本的方法。仔细考虑解读指标的方式。例如，您可以选择最大值或第 99 个百分位数，而不是平均值。
- 自动性能测试：作为部署过程的一部分，在快速运行测试成功通过后自动启动性能测试。自动化应创建新环境、设置初始条件（如测试数据），然后运行一系列基准和负载测试。这些测试的结果应回绑到构建中，以便您可以随着时间推移跟踪性能变化。对于长时间运行的测试，您可以使管道的这一部分与构建的剩余部分异步进行。或者，您也可以使用 Amazon EC2 竞价型实例来通宵运行性能测试。
- 负载生成：您应该创建复制综合或预先记录的用户旅程的一系列测试脚本。这些脚本应该是幂等的，而不是耦合，您可能需要包含预热脚本以便产生有效结果。测试脚本应尽可能再现生产中的使用行为。您可以使用软件或软件即服务（SaaS）解决方案来生成负载。考虑使用 AWS Marketplace 解决方案和竞价型实例：它们是用于生成负载的经济高效的方法。
- 性能可见性：关键指标应该对您的团队可见，尤其是针对每个构建版本的指标。这让您能够随着时间推移看到所有重大的正面或负面趋势。您还应展示有关错误或异常数量的指标，确保测试的是正常工作的系统。
- 可视化：使用可视化技术，清楚了解出现性能问题、热点、等待状态或低利用率的位置。在架构图上叠加性能指标：调用图表或代码有助于快速发现问题。
- 定期审核流程：通常，不存在或不完整的性能审核流程会导致架构性能不佳。如果您的架构性能不佳，请实施性能审核流程，以便推动迭代改进。
- 持续优化：采用一种文化，不断优化云工作负载的性能效率。

最佳实践

PERF05-BP01 建立关键性能指标（KPI）来衡量工作负载运行状况和性能

确定用于定量和定性地衡量工作负载性能的 KPI。KPI 有助于您衡量与业务目标相关的工作负载的运行状况和性能。

常见反模式：

- 只监控系统级指标来深入了解工作负载，却不了解这些指标对业务的影响。
- 认为 KPI 已作为标准指标数据发布和共享。
- 没有定义可量化、可衡量的 KPI。
- KPI 与业务目标或策略不符。

建立此最佳实践的好处：确定可反映工作负载运行状况和性能的具体 KPI，有助于调整团队的工作重点，并确定成功的业务成果。与所有部门共享这些指标可让所有人了解并一致认可阈值、期望值和业务影响。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

利用 KPI，业务和工程团队可在衡量目标和策略以及如何将这些因素结合来取得业务成果方面达成共识。例如，网站工作负载可能会将页面加载时间用作总体性能指示。该指标将是用来衡量用户体验的多个数据点之一。除了确定页面加载时间阈值之外，您还应记录未达到理想性能要求时的预期成果或业务风险。较长的页面加载时间会直接影响最终用户的体验，降低他们的用户体验评分，并会导致客户流失。在定义 KPI 阈值时，请结合考虑行业基准和最终用户期望。例如，如果当前行业基准是两秒内加载网页，而您的最终用户希望网页在一秒内加载，那么您在建立 KPI 时应考虑这两个数据点。

您的团队必须使用实时的精细数据和历史数据作为参考来评估工作负载 KPI，并创建控制面板来对 KPI 数据执行指标计算，从而获得运维和利用率方面的洞察。应记录 KPI，包括支持业务目标和策略的阈值，并且应与所监控的指标对应起来。当业务目标、策略或最终用户需求发生变化时，应重新审视 KPI。

实施步骤

- 确定利益相关方：确定并记录关键的业务利益相关方，包括开发和运营团队。
- 定义目标：与这些利益相关方合作，定义并记录工作负载目标。考虑工作负载的关键性能方面（例如吞吐量、响应时间和成本），以及业务目标（例如用户满意度）。
- 查看行业最佳实践：查看行业最佳实践，确定与工作负载目标相一致的相关 KPI。
- 确定指标：确定与工作负载目标一致且有助于衡量绩效和业务目标的指标。根据这些指标建立 KPI。示例指标包括平均响应时间或并发用户数量等衡量指标。
- 定义并记录 KPI：使用行业最佳实践和工作负载目标为工作负载 KPI 设定目标。使用这些信息设置 KPI 阈值的严重性或警报级别。确定并记录未满足 KPI 时带来的风险和影响。
- 实施监控：使用 Amazon CloudWatch 或 AWS Config 等监控工具收集指标并衡量 KPI。
- 直观地传达 KPI：使用 Amazon Quick 等控制面板工具来可视化 KPI，并就此与利益相关方沟通。
- 分析和优化：定期审查并分析 KPI，确定需要从哪些方面改进工作负载。与利益相关方协作实施这些改进。
- 重新审视和完善：定期审查指标和 KPI，评测其有效性，尤其是在业务目标或工作负载绩效发生变化时。

资源

相关文档：

- CloudWatch 文档
- AWS Partner 监控、日志记录和性能
- AWS observability tools
- The Importance of Key Performance Indicators (KPIs) for Large-Scale Cloud Migrations
- How to track your cost optimization KPIs with the KPI Dashboard
- X-Ray 文档
- Using Amazon CloudWatch dashboards
- Quick KPI

相关视频：

- AWS re:Invent 2023 - Optimize cost and performance and track progress toward mitigation
- AWS re:Invent 2023 - Manage resource lifecycle events at scale with AWS Health
- AWS re:Invent 2023 - Performance & efficiency at Pinterest: Optimizing the latest instances
- AWS re:Invent 2022 - AWS optimization: Actionable steps for immediate results
- AWS re:Invent 2023 - Building an effective observability strategy
- AWS Summit SF 2022 - Full-stack observability and application monitoring with AWS
- AWS re:Invent 2023 - Scaling on AWS for the first 10 million users
- AWS re:Invent 2022 - How Amazon uses better metrics for improved website performance
- Creating an Effective Metrics Strategy for Your Business | AWS Events

相关示例：

- Creating a dashboard with Quick

PERF05-BP02 使用监控解决方案了解性能最为关键的方面

了解并确定在哪些方面提高工作负载性能，会对效率或客户体验产生积极的影响。例如，拥有大量客户交互的网站会因为使用边缘服务在距离客户更近的位置向客户分发内容而受益。

常见反模式：

- 认为标准计算指标（例如，CPU 利用率或内存压力）足够捕获性能问题。
- 只使用由自己选定的监控软件记录的默认指标。
- 只在出现问题时审查指标。

建立此最佳实践的好处：了解关键性能领域可以帮助工作负载负责人监控 KPI 并确定具有高影响力的优先改进。

在未建立这种最佳实践的情况下暴露的风险等级：高

实施指导

设置端到端的跟踪，用于确定流量模式、延迟和关键性能领域。针对速度缓慢的查询或性能欠佳的碎片和分区数据，监控数据访问模式。使用负载测试或监控来确定受约束的工作负载领域。

通过了解架构、流量模式和数据访问模式，提高性能效率，并确定延迟和处理时间。确定随着工作负载增长可能会影响客户体验的潜在瓶颈。在研究了这些方面之后，再看看可以通过部署哪项解决方案来解决这些性能问题。

实施步骤

- 设置端到端的监控，用于收集所有工作负载组件和指标。以下是 AWS 监控解决方案的示例。

服务	使用情形
Amazon CloudWatch 真实用户监控（RUM）	收集真实用户客户端和前端会话的应用程序性能指标。
AWS X-Ray	通过应用程序层跟踪流量，并确定组件间的延迟以及依赖关系。使用 X-Ray 服务地图查看工作负载组件之间的关系和延迟。
Amazon Relational Database Service Performance Insights	查看数据库性能指标并确定性能改进机会。
Amazon RDS 增强监控	查看数据库 OS 性能指标。
Amazon DevOps Guru	检测异常运营模式，以便您可以在运营问题影响客户之前发现它们。

- 执行测试以生成指标，确定流量模式、瓶颈和关键性能领域。以下是一些有关如何执行测试的示例：
  - 设置 CloudWatch Synthetics 金丝雀，使用 Linux cron 作业或 rate 表达式，通过编程方式模拟浏览器端的用户活动，从而生成一段时间内的稳定指标。
  - 使用 AWS 分布式负载测试解决方案生成峰值流量，或者在预期增长速率下测试工作负载。
- 评估指标和遥测数据，确定您的关键性能方面。与团队一起审查这些方面，讨论监控和解决方案以避免瓶颈。
- 试验性能改进，并利用数据来衡量这些更改。例如，使用 CloudWatch Evidently 测试新的改进以及对工作负载的性能影响。

资源

相关文档：

- What's new in AWS Observability at re:Invent 2023
- Amazon Builders' Library
- X-Ray 文档
- Amazon CloudWatch RUM
- Amazon DevOps Guru

相关视频：

- AWS re:Invent 2023 - [LAUNCH] Application monitoring for modern workloads
- AWS re:Invent 2023 - Implementing application observability
- AWS re:Invent 2023 - Building an effective observability strategy
- AWS Summit SF 2022 - Full-stack observability and application monitoring with AWS
- AWS re:Invent 2022 - AWS optimization: Actionable steps for immediate results
- AWS re:Invent 2022 - The Amazon Builders' Library: 25 years of Amazon operational excellence
- AWS re:Invent 2022 - How Amazon uses better metrics for improved website performance
- Visual Monitoring of Applications with Amazon CloudWatch Synthetics

相关示例：

- Measure page load time with Amazon CloudWatch Synthetics
- Amazon CloudWatch RUM Web Client
- 适用于 Python 的 X-Ray 开发工具包
- AWS 上的分布式负载测试

PERF05-BP03 制定流程来提高工作负载性能

制定相应流程，对推出的新服务、设计模式、资源类型和配置进行评估。例如，对新实例产品运行现有性能测试，确定其是否有潜力改进工作负载。

常见反模式：

- 认为当前架构是静态的，将来不会更新。
- 不断对架构进行更改，却不提供任何指标方面的依据。

建立此最佳实践的好处：通过制定架构更改流程，您可以使用所收集的数据来影响以后的工作负载设计。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

工作负载的性能会面临一些关键约束。记录这些约束，以便了解哪些创新可以改进工作负载的性能。在知道有新的服务或技术推出时，借助这些信息来确定消除约束或瓶颈的方法。

确定针对工作负载的关键性能约束。记录工作负载的性能约束，以便了解哪类创新可以提高工作负载的性能。

实施步骤

- 确定 KPI：如 PERF05-BP01 建立关键性能指标（KPI）来衡量工作负载运行状况和性能中所述，确定工作负载性能 KPI，为工作负载建立基准。
- 实施监控：使用 AWS 可观测性工具收集绩效指标并衡量 KPI。
- 执行分析：执行深入分析，确定工作负载中性能欠佳的方面（如配置和应用程序代码），如 PERF05-BP02 使用监控解决方案了解性能最为关键的方面中所述。使用分析和性能工具来确定性能改进策略。
- 验证改进：使用沙盒环境或预生产环境来验证策略的有效性。
- 实施变更：在生产环境中实施变更并持续监控工作负载的性能。记录改进内容，并将变更内容传达给利益相关方。
- 重新审视和完善：定期审查绩效改进流程，确定需要改进的领域。

资源

相关文档：

- AWS 博客
- AWS 的新功能
- AWS Skill Builder

相关视频：

- AWS re:Invent 2022 - Delivering sustainable, high-performing architectures

- AWS re:Invent 2023 - Optimize cost and performance and track progress toward mitigation
- AWS re:Invent 2022 - AWS optimization: Actionable steps for immediate results
- AWS re:Invent 2022 - Optimize your AWS workloads with best-practice guidance

相关示例：

- AWS GitHub

PERF05-BP04 对工作负载进行负载测试

对工作负载进行负载测试，从而验证工作负载能否处理生产负载，并找出任何性能瓶颈。

常见反模式：

- 对工作负载的各个部分进行单独负载测试，而不是测试整个工作负载。
- 在与生产环境不同的基础设施上进行负载测试。
- 只对预期负载而不对其他负载进行负载测试，来预测未来可能会出现问题的方面。
- 没有查阅 Amazon EC2 Testing Policy 并提交“模拟事件提交表”，就执行负载测试。这会导致您的测试无法运行，因为它看起来像是拒绝服务事件。

建立此最佳实践的好处：通过负载测试来衡量性能，可说明随着负载的增加，您将在哪些方面受到影响。这样您便可以在变更影响自己的工作负载之前，对所需进行的变更进行预测。

在未建立这种最佳实践的情况下暴露的风险等级：低

实施指导

云端负载测试是在预期用户负载的实际条件下衡量云工作负载性能的过程。这一过程包括：预置类似于生产的云环境，使用负载测试工具生成负载，分析各个指标来评测工作负载处理实际负载的能力。必须使用生产数据的合成或净化版本（删除敏感信息或身份识别信息）运行负载测试。作为交付管道的一部分，自动执行负载测试，并将结果与预定义的 KPI 和阈值进行比较。这一过程有利于您持续实现所需的性能。

实施步骤

- 定义测试目标：确定待评估工作负载的性能方面，例如吞吐量和响应时间。

- 选择测试工具：选择并配置适合工作负载的负载测试工具。
- 设置环境：根据生产环境设置测试环境。您可以使用 AWS 服务来运行生产规模的环境，进而测试架构。
- 实施监控：使用 Amazon CloudWatch 等监控工具，收集架构中各个资源的指标。您还可以收集和发布自定义指标。
- 定义场景：定义负载测试场景和参数（如测试持续时间和用户数量）。
- 执行负载测试：大规模执行负载测试场景。利用 AWS Cloud 来测试工作负载，发现工作负载的哪些部分无法扩展或者是否以非线性方式扩展。例如，您可以使用竞价型实例以很低的成本生成负载，并在投入生产前发现瓶颈。
- 分析测试结果：对结果进行分析，确定性能瓶颈和需要改进的地方。
- 记录和分享调查发现：记录并报告调查发现和建议。与利益相关方共享此信息，协助他们就性能优化策略做出明智的决策。
- 持续迭代：应定期执行负载测试，尤其是在系统更改更新之后。

资源

相关文档：

- Amazon CloudWatch RUM
- Amazon CloudWatch Synthetics
- AWS 上的分布式负载测试

相关视频：

- AWS Summit ANZ 2023: Accelerate with confidence through AWS Distributed Load Testing
- AWS re:Invent 2022 - Scaling on AWS for your first 10 million users
- Solving with AWS Solutions: Distributed Load Testing

- AWS re:Invent 2021 - Optimize applications through end user insights with Amazon CloudWatch RUM
- Demo of Amazon CloudWatch Synthetics

相关示例：

- AWS 上的分布式负载测试

PERF05-BP05 使用自动化技术主动修复与性能相关的问题

使用关键性能指标（KPI）并结合监控和警报系统，主动解决与性能相关的问题。

常见反模式：

- 只允许运营人员对工作负载进行运营更改。
- 通过设置筛选条件将所有没有主动修复行为的警报发送给运营团队。

建立此最佳实践的好处：主动修复警报行为使支持人员能够集中精力处理那些无法自动完成的工作。这样一来，操作人员只需集中精力处理关键警报，从而避免因处理所有警报而变得应接不暇。

在未建立这种最佳实践的情况下暴露的风险等级：低

实施指导

使用警报触发自动操作，以便在可能的情况下修复问题。如果无法实现自动响应，则将警报上报给能够响应的人员。例如，您的系统在关键性能指标（KPI）超出特定阈值时，能够预测预期 KPI 值并发出警报；或者您的工具在 KPI 超出预期值时，能够自动停止或回滚部署。

实施相应流程，让您在工作负载运行期间了解其性能。构建监控控制面板并确定预期性能基准，以确定工作负载的性能是否达到最佳。

实施步骤

- 确定修复工作流程：识别并了解可以自动修复的性能问题。使用 Amazon CloudWatch 或 AWS X-Ray 等 AWS 监控解决方案，帮助您更好地了解问题的根本原因。
- 定义自动化流程：创建可用于自动修复问题的分步修复流程。
- 配置启动事件：将事件配置为自动启动修复流程。例如，您可以定义一个触发器，以便在实例达到特定 CPU 利用率阈值时自动重启实例。
- 自动执行修复：使用 AWS 服务和技术自动执行修复流程。例如，AWS Systems Manager Automation 提供了一种安全且可扩展的方法来自动执行修复流程。如果更改未成功解决问题，请务必使用自我修复逻辑来还原更改。
- 测试工作流程：在预生产环境中测试自动修复流程。

- 实施工作流程：在生产环境中实施自动修复。
- 制定行动手册：制定行动手册并记录相关内容，概述修复计划的步骤，包括启动事件、修复逻辑和采取的行动。确保对利益相关方进行培训，协助他们有效应对自动修复事件。
- 审查和完善：定期评测自动修复工作流程的有效性。必要时调整启动事件和修复逻辑。

资源

相关文档：

- CloudWatch 文档
- AWS Partner Network 合作伙伴监控、日志记录和性能
- X-Ray 文档
- Using Alarms and Alarm Actions in CloudWatch
- Build a Cloud Automation Practice for Operational Excellence: Best Practices from AWS Managed Services
- Automate your Amazon Redshift performance tuning with automatic table optimization

相关视频：

- AWS re:Invent 2023 - Strategies for automated scaling, remediation, and smart self-healing
- AWS re:Invent 2023 - [LAUNCH] Application monitoring for modern workloads
- AWS re:Invent 2023 - Implementing application observability
- AWS re:Invent 2021 - Intelligently automating cloud operations
- AWS re:Invent 2022 - Setting up controls at scale in your AWS environment
- AWS re:Invent 2022 - Automating patch management and compliance using AWS
- AWS re:Invent 2022 - How Amazon uses better metrics for improved website performance
- AWS re:Invent 2023 - Take a load off: Diagnose & resolve performance issues with Amazon RDS
- AWS re:Invent 2021 -{New Launch} Automatically detect and resolve issues with Amazon DevOps Guru
- AWS re:Invent 2023 - Centralize your operations

相关示例：

- CloudWatch Logs Customize Alarms

PERF05-BP06 让工作负载和服务保持最新状态

随时了解新的云服务和功能，积极采用高效的功能，解决出现的问题并提高工作负载的整体性能效率。

常见反模式：

- 认为当前架构是静态的，将来不会更新。
- 没有任何系统或定期安排来评估更新后的软件和软件包是否与工作负载兼容。

建立此最佳实践的好处：通过建立流程来及时了解新服务和产品的最新情况，您可以采用新的特性和功能、解决问题并提高工作负载性能。

在未建立这种最佳实践的情况下暴露的风险等级：低

实施指导

随着新的服务、设计模式和产品功能的推出，评估可提高性能的方法。通过评估、内部讨论或外部分析来确定哪些方法可以提高工作负载的性能或效率。制定相应流程，评估与工作负载相关的更新、新功能和服务。例如，使用新技术构建概念验证或咨询内部团队。在尝试新想法或新服务时，运行性能测试来衡量这些新想法或新服务对工作负载性能的影响。

实施步骤

- 清点工作负载：清点工作负载软件和架构，确定需要更新的组件。
- 确定更新资源：确定与工作负载组件相关的资讯和更新来源。例如，您可以订阅 AWS 的新功能博客，了解与工作负载组件相匹配的产品。您可以订阅 RSS 源或管理电子邮件订阅。
- 制定更新计划：制定计划来评估工作负载的新服务和新功能。
  - 您可以使用 AWS Systems Manager 清单从 Amazon EC2 实例中收集操作系统（OS）、应用程序和实例元数据，并快速了解哪些实例正在运行软件策略所需的软件和配置，以及哪些实例需要更新。
- 评测新更新：了解如何更新工作负载的组件。利用云中的敏捷性，快速测试新功能如何改善工作负载，从而提高性能效率。
- 采用自动化：采用自动化更新流程，减少部署新功能的工作量，并减少手动过程引起的错误。
  - 您可以使用 CI/CD 自动更新 AMI、容器映像以及其他与云应用程序相关的构件。

  - 您可以使用 AWS Systems Manager 补丁管理器等工具来自动执行系统更新流程，并使用 AWS Systems Manager Maintenance Windows 来安排活动。
- 记录流程：记录评估更新和新服务的流程。为负责人提供所需的时间和空间来研究、测试、试验和验证更新及新服务。回顾记录的业务要求和 KPI，帮助确定会对业务产生积极影响的更新的优先级。

资源

相关文档：

- AWS 博客
- AWS 的新功能
- Implementing up-to-date images with automated EC2 Image Builder pipelines

相关视频：

- AWS re:Inforce 2022 - Automating patch management and compliance using AWS
- All Things Patch: AWS Systems Manager | AWS Events

相关示例：

- Inventory and Patch Management
- One Observability 讲习会

PERF05-BP07 定期检查指标

作为例行维护的一部分或为了应对事件或意外事件，请检查收集到了哪些指标。通过这些检查，找出哪些指标对于解决问题至关重要，以及跟踪哪些其他指标会有助于发现、解决或预防问题。

常见反模式：

- 让指标长时间保持警报状态。
- 创建自动化系统无法操作的警报。

建立此最佳实践的好处：不断检查收集的指标，确认这些指标是否有助于正确地发现问题、解决问题或预防问题。如果让指标长时间保持警报状态，这些指标也会过时。

在未建立这种最佳实践的情况下暴露的风险等级：中

实施指导

不断改进指标收集和监控效果。在响应意外事件或事件的过程中，评估哪些指标有助于解决问题、哪些目前没有跟踪的指标会有助于解决问题。通过这种方法，您可以提高收集的指标的质量，从而预防或更快速地解决未来发生的意外事件。

在响应意外事件或事件的过程中，评估哪些指标有助于解决问题、哪些目前没有跟踪的指标会有助于解决问题。这样，您可以提高收集的指标的质量，从而预防或更快速地解决未来发生的意外事件。

实施步骤

- 定义指标：定义为实现工作负载目标而需要监控的关键性能指标，包括响应时间和资源利用率等指标。
- 建立基准：为每个指标设置基准和期望值。基准应提供参考点，用于确定偏差或异常。
- 建立定期机制：建立定期机制（例如每周或每月）来审核关键指标。
- 识别性能问题：在每次审核期间，评测趋势以及与基准值的偏差。找出任何性能瓶颈或异常情况。对于已发现的问题，深入分析根本原因，了解问题背后的主要原因。
- 确定纠正措施：利用分析结果来确定纠正措施。这可能包括调整参数、修复错误和扩展资源。
- 记录调查发现：记录调查发现，包括已确定的问题、根本原因和纠正措施。
- 迭代和改进：持续评测和改进指标审核流程。利用从之前审核中吸取的经验教训，不断改进流程。

资源

相关文档：

- CloudWatch 文档
- 使用 CloudWatch 代理从 Amazon EC2 实例和本地部署服务器中收集指标和日志
- 使用 CloudWatch Metrics Insights 查询您的指标
- AWS Partner Network 合作伙伴监控、日志记录和性能
- X-Ray 文档

相关视频：

- AWS re:Invent 2022 - Setting up controls at scale in your AWS environment

- AWS re:Invent 2022 - How Amazon uses better metrics for improved website performance
- AWS re:Invent 2023 - Building an effective observability strategy
- AWS Summit SF 2022 - Full-stack observability and application monitoring with AWS
- AWS re:Invent 2023 - Take a load off: Diagnose & resolve performance issues with Amazon RDS

相关示例：

- Creating a dashboard with Quick
- CloudWatch Dashboards

结论

实现和维护性能效率需要数据驱动方法。您应积极考虑访问模式和折衷方案，以便通过优化来实现更高的性能。使用基于基准和负载测试的审核流程，您可以选择合适的资源类型和配置。将基础设施视为代码，有助于您快速、安全地改进自己的架构，同时使用数据来制定有关架构的基于事实的决策。结合使用主动监控和被动监控，能够确保架构性能不随着时间推移而降低。

AWS 致力于帮助您构建性能高效且能提供业务价值的架构。使用本文中讨论的工具和技术，确保取得成功。

贡献者

以下个人和组织参与了本文档的编撰：

- Sam Mokhtari，亚马逊云科技高级效率首席解决方案架构师
- Josh Hart，Amazon Web Services 解决方案架构师
- Richard Trabing，Amazon Web Services 解决方案架构师
- Brett Looney，Amazon Web Services 首席解决方案架构师
- Nina Vogl，Amazon Web Services 首席解决方案架构师
- Eric Pullen，Amazon Web Services 解决方案架构师
- Julien Lépine，Amazon Web Services 专家级 SA 经理
- Ronnen Slasky，Amazon Web Services 解决方案架构师

延伸阅读

如需更多帮助，请查阅以下资源：

- AWS Well-Architected Framework
- AWS 架构中心

文档修订

如需获取有关该白皮书更新的通知，请订阅 RSS 信息源。

变更	说明	日期
对最佳实践进行了小幅更新	PERF03-BP04 已更新，增加了新的服务建议。	2024 年 11 月 6 日
更新了最佳实践指南	对整个支柱进行了多处小更新。	2024 年 6 月 27 日
重大更新和结构重组	支柱重组为五个最佳实践领域（原来为八个）。内容已整合成五个领域并进行了相应更新。 重组后的最佳实践领域分为架构选择、计算和硬件、数据管理、联网和内容分发以及流程和文化。	2023 年 10 月 3 日
次要更新	删除非包容性用语。	2023 年 4 月 13 日
针对新框架进行了更新	为最佳实践更新了规范性指南并增加了新的最佳实践。	2023 年 4 月 10 日
已更新白皮书	为最佳实践更新了新的实施指导。	2022 年 12 月 15 日
已更新白皮书	扩展了最佳实践并增加了改进计划。	2022 年 10 月 20 日
次要更新	删除了非包容性用语。	2022 年 4 月 22 日
次要更新	更新了链接。	2021 年 3 月 10 日
次要更新	将 AWS Lambda 超时时间更改为了 900 秒，并更正了	2020 年 10 月 5 日

	Amazon Keyspaces（Apache Cassandra 兼容）的名称。	
次要更新	修复了失效链接。	2020 年 7 月 15 日
针对新框架进行了更新	重大审核和内容更新	2020 年 7 月 8 日
已更新白皮书	语法问题的细微更新	2018 年 7 月 1 日
已更新白皮书	更新了白皮书来反映 AWS 中的更改	2017 年 11 月 1 日
初次发布	发布了性能效率支柱 – AWS Well-Architected Framework。	2016 年 11 月 1 日

版权声明

客户有责任对本文档中的信息进行单独评测。本文档：(a) 仅供参考，(b) 代表当前的 AWS 产品和实践，如有更改，恕不另行通知，以及 (c) 不构成 AWS 及其附属公司、供应商或许可方的任何承诺或保证。AWS 产品或服务“按原样”提供，不附带任何明示或暗示的保证、陈述或条件。AWS 对其客户承担的责任和义务受 AWS 协议制约，本文档不是 AWS 与客户直接协议的一部分，也不构成对该协议的修改。

AWS 术语表

有关最新的 AWS 术语，请参阅 AWS 词汇表 参考中的 AWS 词汇表。