



优化 Microsoft 工作负载的成本 AWS

AWS 规范性指导



AWS 规范性指导: 优化 Microsoft 工作负载的成本 AWS

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon 的商标和商业外观不得用于任何非 Amazon 的商品或服务，也不得以任何可能引起客户混淆、贬低或诋毁 Amazon 的方式使用。所有非 Amazon 拥有的其他商标均为各自所有者的财产，这些所有者可能附属于 Amazon、与 Amazon 有关联或由 Amazon 赞助，也可能不是如此。

Table of Contents

简介	1
概述	1
受众	1
如何使用本指南	1
目标业务成果	3
成本优化之旅	4
优化成本的首要建议	6
概述	6
热门推荐	6
AWS 优化和许可评估	8
概述	8
评估选项	8
全面评估	9
范围工作负载	9
收集数据	9
分析数据	10
计划下一步行动	12
评估影响	13
后续步骤	14
其他 资源	14
亚马逊 EC2 上的 Windows	15
自动执行停止和启动计划	15
概述	15
案例研究	16
成本优化场景	16
成本优化建议	18
其他 资源	29
大小合适的 Windows 工作负载	29
概述	29
成本优化场景	30
成本优化建议	30
建议	38
其他 资源	38
为 Windows 工作负载选择正确的实例类型	39

概述	39
成本优化建议	39
后续步骤	47
其他 资源	48
为 Windows 和 SQL Server 工作负载提供许可证	49
概述	49
Amazon EC2 专属主机	49
AWS 许可选项	52
带上 Windows 服务器许可证	53
成本优化场景	54
成本优化建议	59
其他 资源	59
优化 Windows 在亚马逊 EC2 上的支出	60
概述	60
了解节省计划	60
成本优化场景	65
成本优化建议	68
其他 资源	70
使用 AWS 工具监控成本	70
概述	70
成本优化建议	71
其他 资源	74
SQL Server	75
选择高可用性和灾难恢复解决方案	76
概述	76
SQL Server 始终开启可用性组	77
SQL Server 始终开启故障转移群集	78
SIOS DataKeeper	80
Always On 可用性组	82
分布式可用性组	83
日志传送	84
AWS Database Migration Service	86
AWS Elastic Disaster Recovery	86
成本比较	87
成本优化建议	90
其他 资源	91

了解 SQL 服务器许可	91
概述	91
AWS 许可选项	91
携带许可证对成本的影响	92
许可证优化	93
成本优化建议	90
其他 资源	38
为 SQL Server 工作负载选择正确的 EC2 实例	99
概述	99
成本比较	99
成本优化场景	100
成本优化建议	101
其他 资源	104
整合实例	105
概述	105
成本优化场景	105
成本优化建议	106
其他 资源	107
比较 SQL 服务器版本	107
概述	107
成本影响	108
成本优化建议	110
其他 资源	114
评估 SQL Server 开发者版	114
概述	114
成本影响	114
其他 资源	38
在 Linux 上评估 SQL 服务器	117
概述	117
成本影响	118
成本优化建议	119
其他 资源	120
优化 SQL 服务器备份策略	120
概述	120
使用支持 VSS 的快照进行服务器级备份	121
使用 SQL 服务器备份 AWS Backup	123

数据库级备份	125
成本优化建议	133
其他 资源	135
实现 SQL 服务器数据库的现代化	135
概述	135
数据库产品	136
亚马逊 RDS 和 Aurora 的比较	136
成本优化建议	138
其他 资源	142
优化 SQL 服务器的存储	142
概述	142
Amazon EBS 的固态硬盘存储类型、性能和成本	143
亚马逊 EBS 的通用固态硬盘成本优化	144
其他 资源	146
使用 Compute Optimizer 优化 SQL Server 许可	146
概述	146
成本优化建议	147
配置 Compute Optimizer	147
其他 资源	148
使用 Compute Optimizer 优化 SQL Server 的大小	149
概述	149
配置 Compute Optimizer	149
其他 资源	150
查看 SQL Server 工作负载 Trusted Advisor 的建议	150
概述	150
成本优化建议	150
配置 Trusted Advisor	151
其他 资源	151
容器	153
将 Windows 应用程序移至容器	153
概述	153
成本收益	154
成本优化建议	155
后续步骤	158
其他 资源	158
优化 Amazon ECS 上 AWS Fargate 任务的成本	159

概述	159
成本收益	159
成本优化建议	159
后续步骤	165
其他 资源	165
了解您的 Amazon EKS 成本	165
概述	165
成本收益	165
成本优化建议	166
后续步骤	169
其他 资源	169
使用 App2Container 对 Windows 应用程序进行平台化	169
概述	169
成本收益	170
成本优化建议	171
后续步骤	171
其他 资源	171
存储	172
Amazon EBS	172
将亚马逊EBS卷从 gp2 迁移到 gp3	172
修改亚马逊EBS快照	176
删除未连接的 Amazon 卷 EBS	178
Amazon FSx	181
选择合适SMB的文件存储空间	181
在 Amazon 中启用重复数据删除 FSx	185
了解 Windows 文件服务器中的FSx数据分片	187
了解 Amazon 的HDD容量使用情况 FSx	191
使用单个可用区	193
AWS Storage Gateway	195
亚马逊 S3 文件网关	195
Amazon FSx 文件网关	195
成本影响	195
成本优化建议	197
其他 资源	200
Active Directory	201
亚马逊 EC2 上自行管理的活动目录	201

概述	201
成本影响	201
成本优化建议	202
其他 资源	205
AWS Managed Microsoft AD	206
概述	206
成本影响	206
成本优化建议	206
其他 资源	207
AD Connector	208
概述	208
成本影响	208
成本优化建议	208
其他 资源	209
.NET	210
重构为现代。NET然后迁移到 Linux	211
概述	211
成本影响	211
成本优化建议	212
其他注意事项和资源	212
容器化。NET应用程序	213
概述	213
成本影响	213
成本优化建议	214
其他 资源	216
使用 Graviton 实例和容器	217
概述	217
成本影响	217
成本优化建议	219
其他 资源	219
Support 支持静态的动态缩放。NET框架应用程序	220
概述	220
成本影响	224
成本优化建议	225
其他 资源	226
使用缓存来减少数据库需求	226

概述	226
成本影响	226
成本优化建议	227
其他 资源	232
以无服务器为例。 NET	233
概述	233
成本影响	233
成本优化建议	233
其他 资源	236
考虑专门构建的数据库	237
概述	237
成本影响	240
成本优化建议	242
其他 资源	243
后续步骤	244
文档历史记录	245
术语表	246
#	246
A	246
B	249
C	250
D	253
E	256
F	258
G	259
H	259
I	260
L	262
M	263
O	266
P	269
Q	271
R	271
S	274
T	276
U	278

V	278
W	278
Z	279
.....	cclxxx

优化微软工作负载的成本 AWS

Bill Pfeiffer、Chase Lindeman 和 Amazon Web Services 的凯文·苏汉 ()AWS

2024 年 6 月 ([文件历史记录](#))

概述

本指南提供了建议、最佳实践和策略，以帮助您优化 Microsoft 工作负载的成本 AWS。该指南还包括基础 AWS 知识、成本优化技术和参考架构，可帮助您构建和自动化符合业务目标的高性价比、高性能工作负载。本指南统称为“微软 AWS 成本优化”(MACO)。MACO 指南由行业专家制定，基于真实场景。

本指南涵盖以下 Microsoft 工作负载：

- 亚马逊 Elastic Compute Cloud (亚马逊 EC2) 上的 Windows
- SQL Server
- 容器
- 存储
- Active Directory
- .NET

受众


本指南适用于架构师、工程师、管理员、主管、首席技术官、技术决策者和 AWS 合作伙伴。有 AWS 计费、Microsoft 技术和 AWS 系统管理的经验和基本了解很有帮助，但不是必需的。

如何使用本指南


您可以使用本指南来规划和实施您的 MACO 云之旅。我们建议你从头到尾阅读本指南，以全面了解优化 Microsoft 工作负载成本的选项和方法 AWS。您可以根据组织的需求查看以下工作负载部分：

- [亚马逊 EC2 上的 Windows](#)
- [SQL Server](#)
- [容器](#)

- [存储](#)
- [Active Directory](#)
- [.NET](#)

 Important

本指南中提供的代码示例仅用于演示目的。在生产环境中使用所有代码之前，最好先在开发环境中对其进行测试。在实现任何代码之前，我们建议您对代码进行小批量测试，然后使用查看代码导致的成本变化[AWS Cost Explorer](#)。这可以帮助您解决边缘情况和其他以后可能会出现的问题。

 Important

本指南中的定价示例基于发布时的价格。价格可能会发生变化。此外，您的费用可能会有所不同 AWS 区域，具体取决于您的 AWS 服务 配额以及与您的云环境相关的其他因素。

目标业务成果

本指南可以帮助您和您的组织实现以下业务成果：

- 了解如何使用 AWS 优化和许可评估 (AWS OLA) 根据资源利用率、第三方许可和应用程序依赖关系来评估和优化您当前的本地和云环境。
- 使用适用于 Microsoft 工作负载的 AWS 现代化计算器制定成本优化的商业案例。
- 优化特定微软工作负载的成本，包括亚马逊弹性计算云 (Amazon EC2)、SQL Server、容器、存储、Active Directory 和 .NET 上的 Windows 工作负载。

成本优化之旅

云迁移之旅的范围、时间和具体路径取决于您的业务目标、技术要求和其他因素。本节举例说明了以云财务管理为重点的云迁移之旅，[AWS并遵循了MAC O](#)的建议和最佳实践。您可以使用此示例来了解如何为 Microsoft 工作负载设计云迁移之旅。

以下高级任务说明了组织在实施 MACO 建议和最佳实践时可以采用的方法：

- 制定标签策略并启用用户定义的成本分配标签。有关更多信息，请参阅 AWS 白皮书《[为资源添加标签 AWS 的最佳实践](#)》。
- 根据应用程序、团队或部门定义预算。有关更多信息，请参阅《AWS 账单和[成本管理](#)用户指南》AWS Budgets中的“使用管理成本”。
- 执行 AWS 优化和许可评估 (AWS OLA) 以加快节省开支。有关更多信息，请参阅 AWS 文档中的[AWS 优化和许可评估](#)。
- 使用自带适用于 Windows 和 SQL Server 工作负载的许可证 (BYOL)。Amazon Elastic Compute Cloud Dedicated Hosts有关更多信息，请参见本指南的[为 Windows 和 SQL Server 工作负载提供许可证](#)部分。
- 在上优化您的 SQL Server 许可 AWS。有关更多信息，请参见本指南的[了解 SQL 服务器许可](#)部分。
- 为 Windows 工作负载选择正确的实例类型。有关更多信息，请参见本指南的[为 Windows 工作负载选择正确的实例类型](#)部分。
- 为 SQL 工作负载选择正确的实例类型。有关更多信息，请参见本指南的[为 SQL Server 工作负载选择正确的 EC2 实例](#)部分。
- 将亚马逊 Elastic Block Store (Amazon EBS) 从 gp2 迁移到 gp3。有关更多信息，请参见本指南的[将亚马逊EBS卷从 gp2 迁移到 gp3](#)部分。
- 在 EC2 实例调度器开启的情况下控制工作负载。AWS有关更多信息，请参见本指南的[自动执行停止和启动计划](#)部分。
- 使用 SQL Server 开发者版，消除非生产工作负载的 SQL Server 成本。有关更多信息，请参见本指南的[评估 SQL Server 开发者版](#)部分。
- 使用适用于 Windows File Server 的 Amazon FSx 的单一可用区域来处理开发和测试工作负载。有关更多信息，请参见本指南的[使用单个可用区](#)部分。
- 使用 AWS Compute Optimizer来调整你的 Windows 工作负载的大小。有关更多信息，请参见本指南的[大小合适的 Windows 工作负载](#)部分。

- 使用 Savings Plans 优化亚马逊 EC2 上的 Windows 支出。有关更多信息，请参见本指南的 [优化 Windows 在亚马逊 EC2 上的支出](#) 部分。
- 在 FSx for Windows File Server 上启用重复数据删除。有关更多信息，请参见本指南的 [在 Amazon 中启用重复数据删除 FSx](#) 部分。
- 在 FSx for Windows File Server 上对文件系统使用数据分片。有关更多信息，请参见本指南的 [了解 Windows 文件服务器中的 FSx 数据分片](#) 部分。
- 优化您的 SQL 服务器备份策略。有关更多信息，请参见本指南的 [优化 SQL 服务器备份策略](#) 部分。
- 使静态 .NET 框架应用程序支持动态缩放。有关更多信息，请参阅 [Support 支持静态的动态缩放。NET 框架应用程序](#) 本指南的。
- 使用无服务器的 .NET 微服务。有关更多信息，请参见本指南的 [以无服务器为例。NET](#) 部分。
- 将你的 Windows 应用程序移到容器中。有关更多信息，请参见本指南的 [容器化。NET 应用程序](#) 部分。
- [AWS Compute Optimizer](#) 用于调整在亚马逊弹性容器服务 (Amazon ECS) 上 AWS Fargate 运行的 Windows 容器的大小。有关更多信息，请参见本指南的 [启用 Compute Optimizer](#) 部分。
- 重构到现代 .NET 并迁移到 Linux。有关更多信息，请参见本指南的 [重构为现代。NET 然后迁移到 Linux](#) 部分。
- 利用 Graviton 实例和容器。有关更多信息，请参见本指南的 [使用 Graviton 实例和容器](#) 部分。
- 实现 SQL 服务器数据库的现代化。有关更多信息，请参见本指南的 [实现 SQL 服务器数据库的现代化](#) 部分。
- 设计活动目录基础架构。有关更多信息，请参见本指南的 [Active Directory](#) 部分。

有关以云财务管理为重点的客户旅程的更多信息 AWS，请参阅 AWS 白皮书 [云财务管理功能](#)。

优化成本的首要建议

概述

成本优化是 Well-Architected Framework 的支柱之一，它在您的云迁移计划中起着至关重要的作用。您将在本指南中找到成本优化的建议，但本节列出了影响力最大的建议。您可以快速实施这些建议，它们将对您的组织产生重大影响。这些建议可以帮助您为整个成本优化工作奠定基础。

热门推荐

下表列出了影响力最大的成本优化的首要建议。“难以实现”列根据最容易实现的内容 (1) 到最难实现的范围 (5) 对每项优化进行评级。“预计节省额”列显示基于百分比的估算值，即您的组织可以为每项建议的优化节省多少资金。

优化	实施困难	预估节省
大小合适的 Windows 工作负载	3	25%
为 Windows 和 SQL Server 工作负载提供许可证	3	30%
评估 SQL Server 开发者版	2	20%
了解 SQL 服务器许可	2	高达 50%
自动执行停止和启动计划	3	高达 40%
为 Windows 工作负载选择正确的实例类型	1	10—30%
重构为现代。NET 然后迁移到 Linux	5	10—20%
优化 Windows 在亚马逊 EC2 上的支出	3	高达 20—40%
将亚马逊 EBS 卷从 gp2 迁移到 gp3	4	最高 20%

⚠ Important

上表中估计的节省额适用于每个单独的技术领域，而不是账户内的总 AWS 支出。例如，您可以在各种环境类型和大小中实施实例调度器，这可能会改变潜在的节省。这些估算值专门适用于 Amazon EC2 实例成本，并不意味着其他实例可以节省任何总体成本 AWS 服务。这些估计值仅作为衡量标准提供，而不是保证。

MACO专家可以更深入地讨论成本优化。要安排会议深入了解您的用例，请联系您的客户团队或发送电子邮件至 optimize-microsoft@amazon.com。

AWS 优化和许可评估

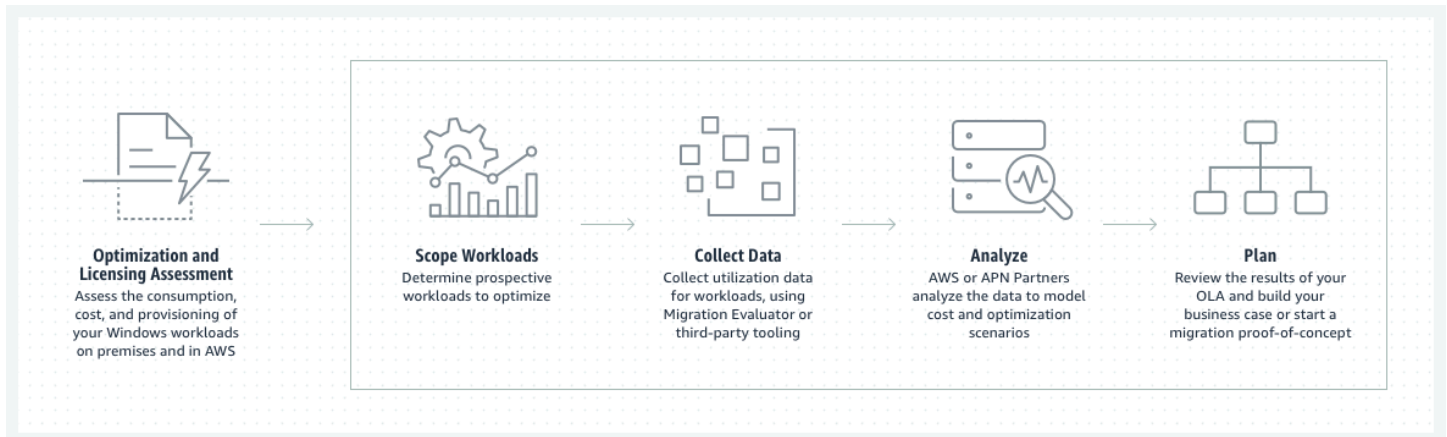
概述

[AWS 优化和许可评估 \(AWS OLA\)](#) 可以帮助您根据资源利用率、第三方许可和应用程序依赖关系来评估和优化当前的本地和现有云环境。您可以使用 AWS OLA 来帮助您的组织制定迁移和许可策略，以便在您迁移到现有 Microsoft 工作负载 AWS 或评估现有工作负载时节省成本。AWS OLA 还可以帮助您实现以下目标：

- 了解现有部署、应用程序性能和合同。
- 合理调整资源规模。
- 制定路线图 AWS Cloud.
- 使用现有投资并仅为实际用量付费，从而降低或消除成本。

我们建议您将 AWS OLA 作为[成本优化之旅](#)的第一步。您可以与合作完成 AWS Partner Network OLA AWS A。他们将帮助您收集评估数据，并为您提供优化许可和实例成本的建议。

下图概述了评估过程。



评估选项

你可以从 Microsoft 工作负载的两个 AWS OLA 选项中进行选择 AWS：

- **精简版** — 在此用例中，您的所有工作负载都在 VMware 上。你可以提供 [rv AWS Tools](#) 的输出。然后，AWS 可以提供 1-5 天的周转时间。这种方法使用直接从 VMware vCenter 中提取的 point-in-time 信息来制定规模建议并提供按需定价选项。

- 完整版 — 在此用例中，您的混合环境运行在不同的云提供商、物理服务器和虚拟服务器中。AWS 使用操作系统代理收集 14 到 30 天的使用数据。这 AWS 允许根据您的应用程序使用模式做出明智的实例规模决定。AWS 使用多种第三方工具（例如 Cloudamize）来完成分析。AWS 与其合作，通过多种定价选项 AWS Partner Network 来帮助提供最终的总拥有成本 (TCO) 评估，这些选项将定价模型和不同的架构考虑在内。

全面评估

完整的 AWS OLA 评估以一小时的电话拉开序幕。在本次电话会议中，AWS 帮助您确定支持迁移的最佳 AWS 基础架构，选择数据收集方法，并确定完成的时间表。在组织中实施发现工具取决于数据收集方法、组织规模以及组织用于管理其服务器群的工具。收集使用情况数据通常需要两周时间。

完整的 AWS OLA 流程需要 30-45 天，包括以下几个阶段：

- 范围工作负载
- 收集数据
- 分析数据
- 计划下一步行动

范围工作负载

首先，AWS 与您和您的团队合作确定评估的范围。这通常按环境类型（例如，非生产环境和生产环境）进行细分。范围包括工作负载的位置。这可能是您要迁移到的工作负载 AWS、已经在运行的工作负载 AWS（例如 Amazon EC2 的 AWS OLA），或者是在其他云提供商中运行的工作负载。

收集数据

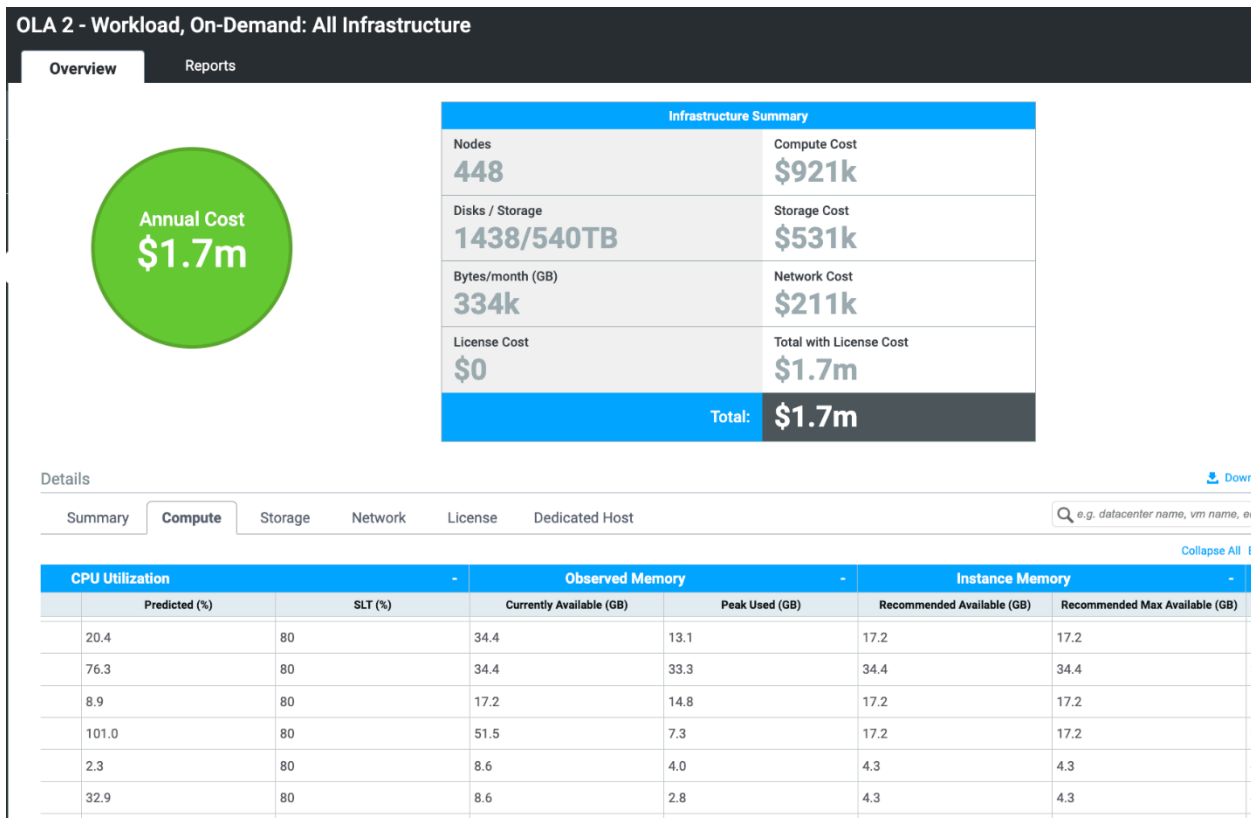
接下来，AWS 部署工具来帮助发现资源并从服务器收集性能数据。该工具有四个部署选项：

- 可以查询虚拟机管理程序的工具（仅需要 VMware vCenter 或 Hyper-V 凭据）
- 可以部署在物理机或虚拟机上的代理
- 根据您的环境和操作系统，使用 SSH、Windows 远程管理 (WinRM) 或 Windows 管理工具 (WMI) 进行无代理发现
- 平面文件数据收集和分析

对于您的工具部署，您可以混合搭配每个选项并整合结果。确保无论您选择哪种选项，都不会给您的 IT 资源带来压力，这一点至关重要。AWS 努力使评估过程尽可能交钥匙。除了简短的电话协助设置外，O AWS LA 团队和 Microsoft 专业解决方案架构师还将准备总拥有成本 (TCO) 分析和建议以供审查。

分析 CPU 利用率、RAM 利用率、存储吞吐量、IOPS 和网络吞吐量时，数据收集通常需要两到三周的时间。理想情况下，此收集发生在工作月的高峰时段（例如，在 end-of-month 财务报告期间）。AWS 想要捕捉峰值使用情况，因为这可以为大小合适的 AWS 实例提供良好的统计样本，同时仍然可以保证性能可以超过本地可用水平。AWS 将利用率指标与各代处理器的性能启发式方法相结合，以准确确定给定工作负载需要多少 CPU 和 RAM。这些目标通常低于内部分配的目标。这不仅可以降低实例大小的计算成本，还可以优化许可成本。

以下仪表板视图显示了可通过评估获取的基础架构成本示例。

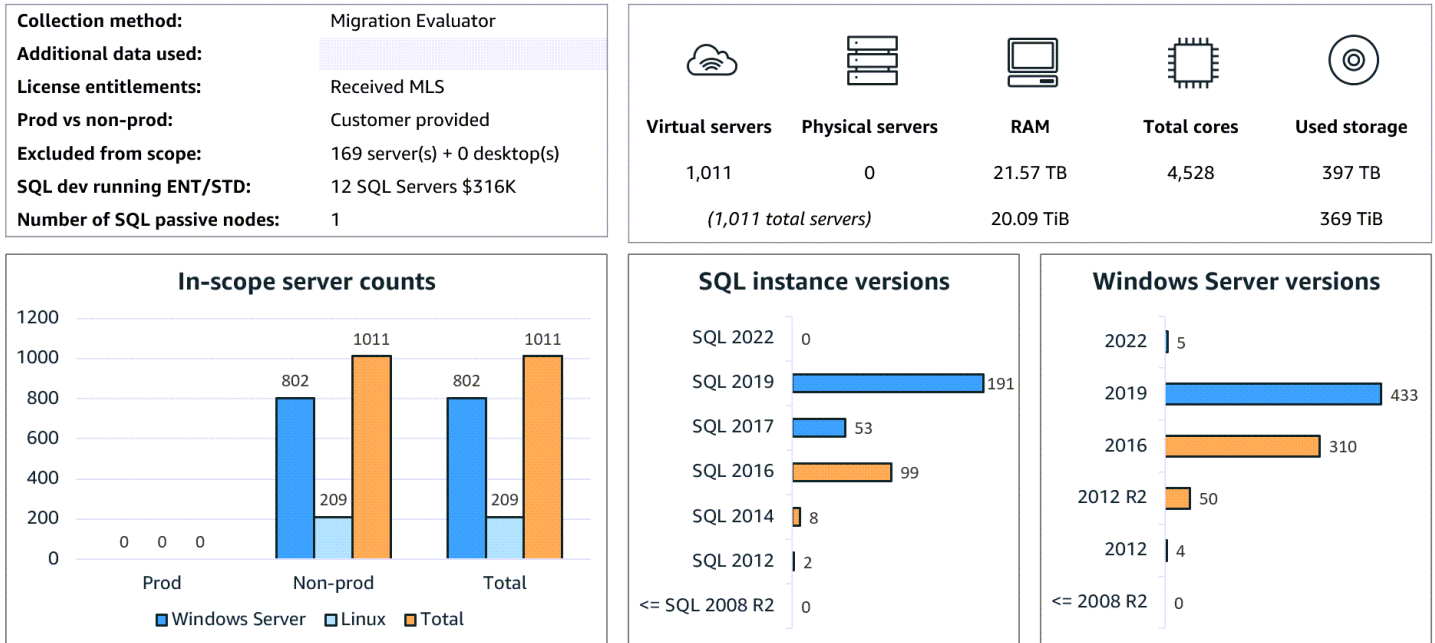


分析数据

AWS 在数据收集完成后提供汇报演示文稿。AWS 审查数据，总结调查结果，然后就本地使用和云迁移提出建议。您可以通过研究整合机会、弹性增益（可以关闭或按季节调整工作负载）、正确的 SKU 机会（例如，正在使用 SQL Server Enterprise 版，但资源要求和功能使用情况表明 SQL Server 标准版是合适的）来降低计算和许可成本。对于像 SQL Server 这样由内核许可的产品，将工作负载放在更

昂贵的计算实例中通常具有经济意义。也就是说，如果CPU配置和RAM与vCPU的比例对减少包含许可证和自带许可 (BYOL) 用例的许可内核数量产生净影响。

以下是基于评估收集的数据的示例分析。



常见的优化方案包括确定 AWS 资源优化机会和第三方许可证节省。

AWS 资源优化机会的示例：

- 避免为高峰使用量进行过度配置。
- 避免过度指定和未充分利用资源。
- 调整您的实例大小并迁移到最新一代的 EC2 实例。
- 通过迁移到托管数据库来节省运营成本。

第三方许可证节省的示例：

- 减少运行相同工作负载所需的内核。
- 删除不必要的 SQL Server 企业版和附加包。
- 移除僵尸服务器并更换过时的硬件。
- 使用 BYOL 和包含许可证的选项来减少 future 的商业协议。
- 实现开源和云原生解决方案的现代化。

计划下一步行动

最后，AWS 使用收集的性能数据来估算具体的工作负载规模和成本。AWS 还可以汇总查看您的范围环境并提供定量分析。这可以帮助您确定最佳选择是本地刷新还是迁移到 AWS。您可以使用 OLA 末尾提供的总拥有成本分析摘要（如以下示例所示）来构建云经济业务案例 AWS。

	Option 1: Amazon EC2 shared	Option 1a: Amazon EC2 shared + power management	Option 2: Amazon EC2 mixed	Option 2a: Amazon EC2 mixed + power management
<i>Option details: compute</i>	100% Reserved Instances (RIs)	RIs + on-demand power management	100% RIs	RIs + on-demand power management
<i>Option details: Microsoft licenses</i>	WS LI and SQL BYOL	WS LI and SQL BYOL	WS BYOL or LI+SQL BYOL	WS BYOL or LI+SQL BYOL
Compute costs¹				
Year 1 compute cost	\$414,546	\$482,623	\$504,019	\$513,941
Year 1 vendor license included cost	\$392,858	\$244,415	\$9,804	\$4,783
	\$807,404	\$727,038	\$513,823	\$518,724
<i>Total compute savings in year 1, compared to Option 1</i>	—	10% (\$80,366)	36% (\$293,581)	36% (\$288,680)
Storage and networking costs²				
Annual estimated storage cost	\$336,494	\$336,494	\$336,494	\$336,494
Annual estimated networking cost	\$41,455	\$41,455	\$41,455	\$41,455
	\$377,949	\$377,949	\$377,949	\$377,949
Microsoft license costs**				
WS/CIS annual Software Assurance (SA) + current SPLA/Subs cost	\$0	\$0	\$0	\$0
WS/CIS license + SA + SPLA/Subs true-up cost	\$0	\$0	\$0	\$0
SQL annual SA + current SPLA/Subs cost	\$0	\$0	\$0	\$0
SQL license SA + current SPLA/Subs true-up cost	\$0	\$0	\$0	\$0
	\$0	\$0	\$0	\$0
Total estimated costs	\$1,185,353	\$1,104,987	\$891,772	\$896,673
<i>Annual TCO savings in year 1, compared to Option 1</i>	—	7% (\$80,366)	25% (\$293,581)	24% (\$288,680)

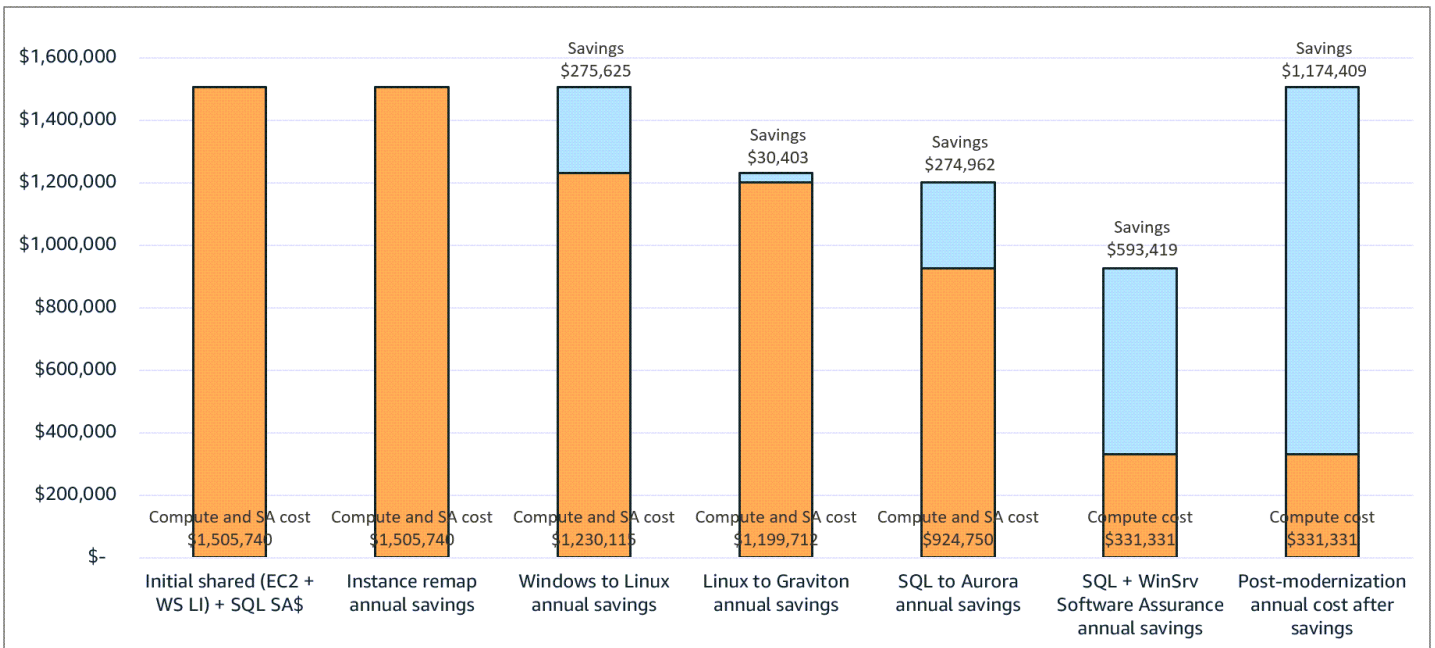
¹ Pricing model used: 3-year, no upfront RI

² Software Assurance and true-up costs provided by Microsoft

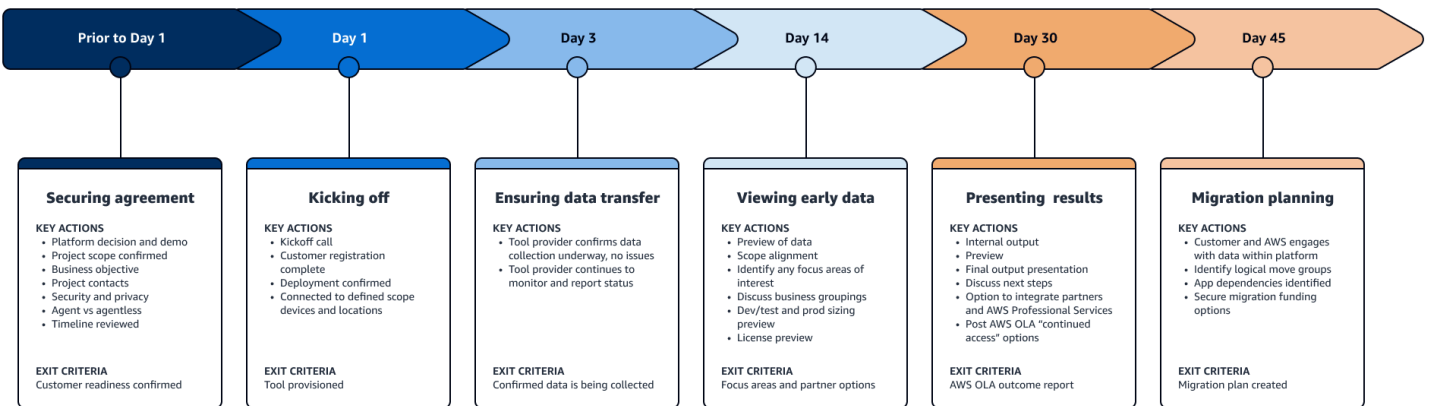
AWS OLA 还通过提出以下建议，深入了解现代化可能对现有工作负载产生的影响：

- 迁移到 Linux 操作系统。
- 添加对 ARM 处理器 (AWS Graviton) 的应用程序支持。
- 将 SQL Server 工作负载转移到亚马逊 Aurora。
- 通过将 Windows 和 SQL Server 工作负载迁移到开源技术来消除软件保障。

下图显示了通过现代化技术可以节省的成本，例如从 Windows 迁移到 Linux 或从 SQL Server 迁移到 Aurora。



完整的 AWS OLA 流程从开始到结束大约需要 45 天。下图显示了一个示例时间轴。



如果你有一个纯粹的 VMware 环境并且可以提供 RVTools 的输出，那么你可以将这个时间表缩短到一个工作周。此外，AWS 还可以分析包含资产和利用率数据的平面文件，例如 CPU 平均值、CPU 峰值、RAM 平均值和 RAM 峰值。

评估影响

通过调整规模，普通客户的成本通常会降低 20-30%。根据使用情况数据，正确调整大小可将源工作负载与大小最佳的 AWS 实例相匹配。这些适当规模的调整不仅可以降低每月的 AWS 环境成本，而且经常可以节省组织内其他部门的开支。例如，Windows 或 SQL Server 许可的增长 20-30% 可以减少微软的下次调整，或者可以腾出许可来使用其他应用程序。line-of-business 整合和适当调整 SQL Server 工作负载通常是实现最显著财务收益的地方。

AWS 可以帮助您将系统归类为现代化存储桶。有些系统是传统系统，在财务上不可行，而另一些系统则可能被现代化为容器或无服务器应用程序，从而实现最显著的节省。与您的 AWS 团队的对话从云支持什么的笼统话题转变为关于如何以及为何对特定工作负载进行现代化改造的更具体的讨论。AWS 还可以帮助您探索潜在的创新机会。

后续步骤

如果你要开始针对在本地环境或本地环境中运行的 Microsoft 工作负载进行成本优化 AWS，请与你的 AWS 客户团队联系并申请 O AWS LA。AWS 团队成员可以回答您的问题，并帮助您决定 O AWS LA 最终是否是您和您的组织的正确选择。或者，您可以[在线申请 AWS OLA](#)。

其他资源

- [AWS 优化和许可评估 \(AWS 文档 \)](#)
- [AWS re: Invent 2022-如何节省成本和优化微软的工作负载 AWS \(ENT205\) \(\)](#) YouTube

亚马逊 EC2 上的 Windows

[Amazon Elastic Compute Cloud \(Amazon EC2\)](#) 是一个高度灵活且可扩展的云计算平台，非常适合运行 Windows 工作负载。您可以使用 Amazon EC2 在安全、可靠、高度可用且适应性强的基础设施上部署、管理和扩展您的 Windows 服务器工作负载。AWS Cloud 考虑一下在 Amazon EC2 上运行 Windows 工作负载的以下主要好处：

- **可扩展性** — Amazon EC2 使您能够轻松扩展 Windows 工作负载以适应不断变化的需求。您可以快速创建新的 EC2 实例来应对不断增长的需求，并且在不再需要实例时同样轻松地将其终止。您只需为实际使用的资源付费。
- **灵活性** — Amazon EC2 上的 Windows 支持各种实例类型，这些实例类型旨在满足各种工作负载要求，从通用实例到内存或计算优化型实例。这种灵活性可确保您可以为基于 Windows 的特定应用程序选择最佳实例类型，从而最大限度地提高性能并最大限度地降低成本。
- **安全** - AWS 为您的 Windows 工作负载提供多层安全保护，包括网络防火墙、数据加密和安全访问控制。这意味着您可以信任自己的应用程序和数据受到保护，同时仍然可以完全控制您的安全设置和配置。
- **成本效益** — 按 pay-as-you-go 价模式使您只需为所使用的资源付费，无需对硬件和软件进行前期投资。该模型还使您能够优化成本、减少资本支出并提高运营效率。对于各种规模的企业来说，这是一种理想的定价模式。

本指南的这一部分涵盖以下主题：

- [自动执行停止和启动计划](#)
- [大小合适的 Windows 工作负载](#)
- [为 Windows 工作负载选择正确的实例类型](#)
- [为 Windows 和 SQL Server 工作负载提供许可证](#)
- [优化 Windows 在亚马逊 EC2 上的支出](#)
- [使用 AWS 工具监控成本](#)

自动执行停止和启动计划

概述

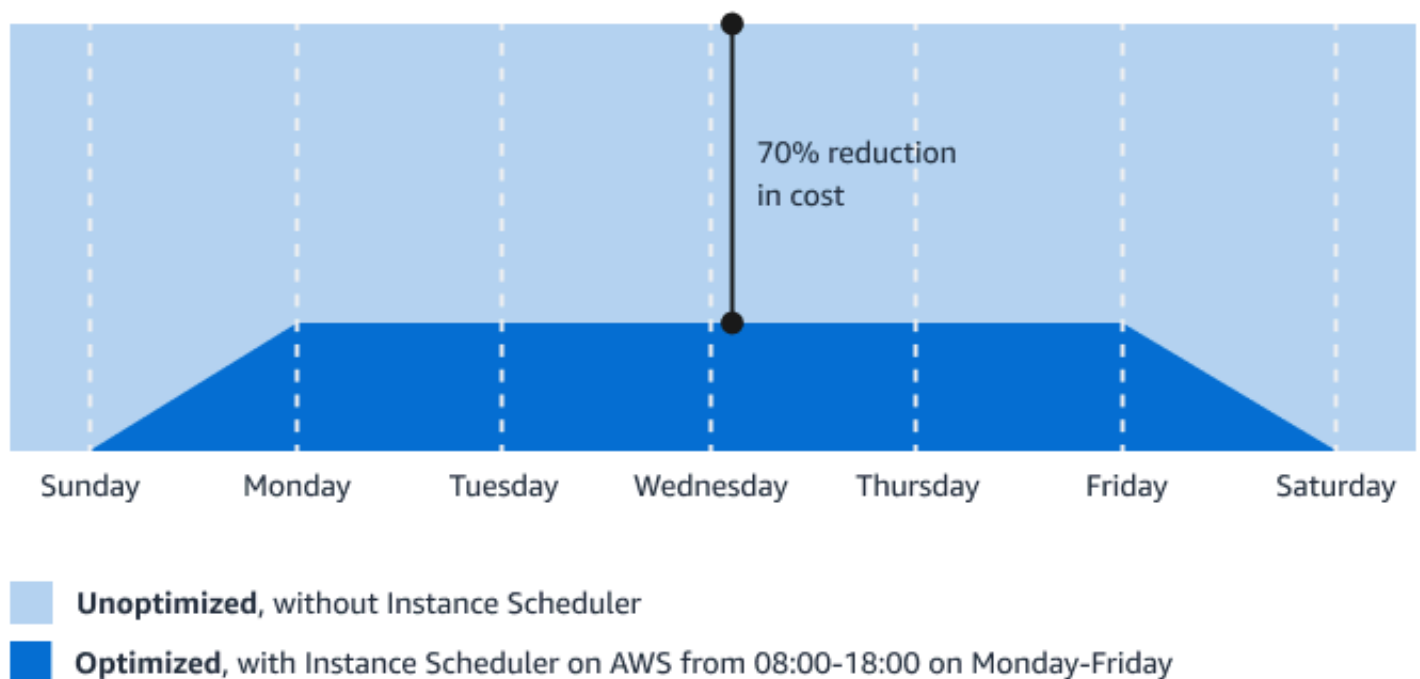
开启的[实例调度器 AWS](#)可以自动启动和停止 [Amazon EC2 和亚马逊关系数据库服务 \(Amazon RDS\)](#) 实例，从而帮助您降低运营成本。如果您让所有实例持续以最大利用率运行，则最终可能会为未使用的

资源付费。开启的实例调度器 AWS 使您能够在不需要实例的时间关闭实例，例如在非工作时间、周末或其他使用率较低的时段。随着时间的推移，这可以显著节省成本。

上的 Instance Scheduler AWS 还提供跨账户实例调度、自动标记以及使用命令行界面或维护窗口配置计划或周期的功能。[AWS Systems Manager](#) 这些功能可以帮助您更有效、更准确地管理实例，在不同的项目或团队之间跟踪和分配成本。

案例研究

以一家公司为例，该公司在生产环境 AWS 中使用实例调度器在每天工作时间以外自动停止实例。如果该公司让所有实例都保持满负荷运行，那么对于那些仅在正常工作时间内才需要的实例，他们可以节省高达 70% 的成本。下图显示了如何将每周利用率从 168 小时减少到 50 小时。



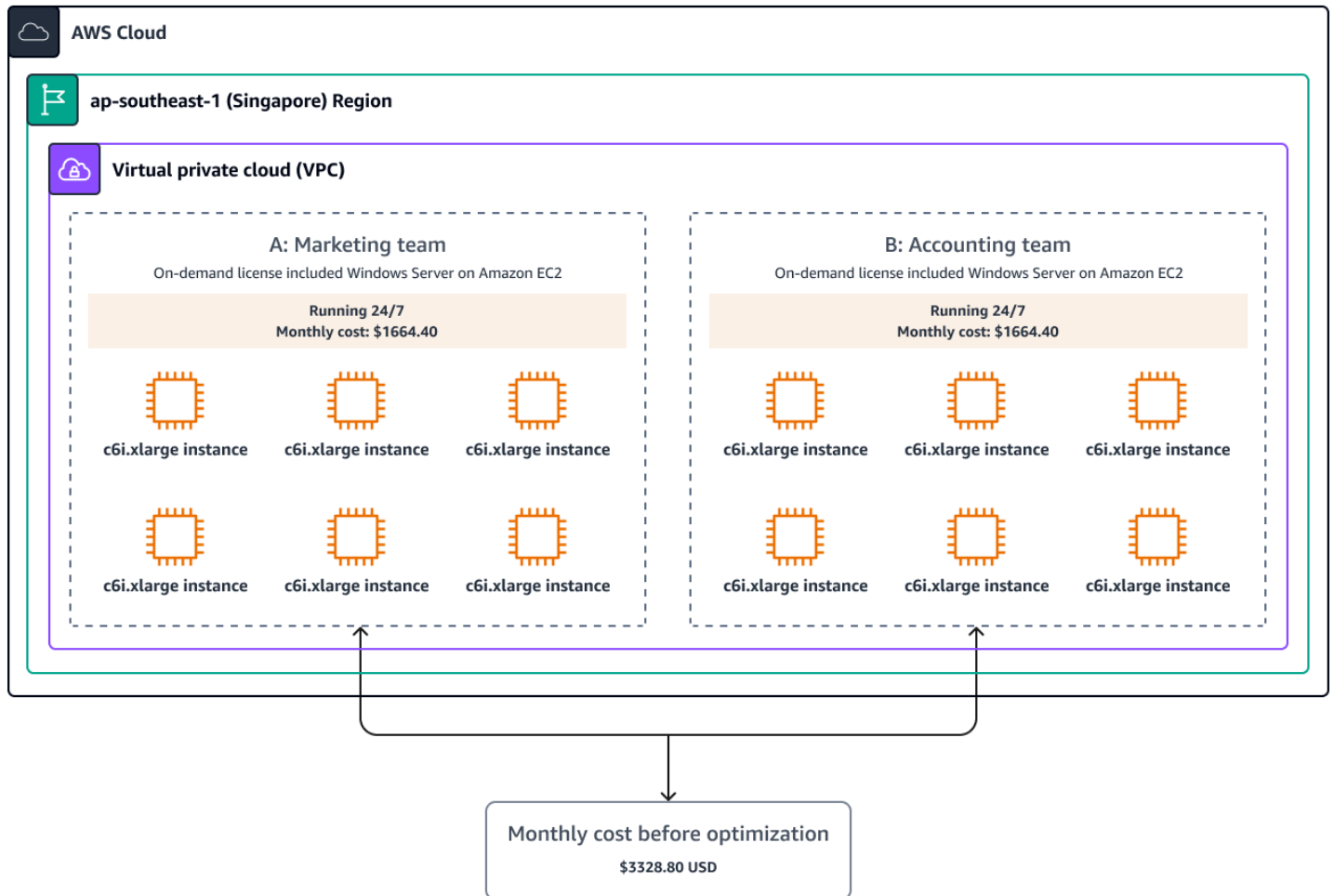
再举一个例子。电力公司牙买加公共服务有限公司 (JPS) 将其数据库迁移到 Amazon RDS。现在，JPS 使用 Amazon EC2 来托管 API 服务并运行其他应用程序。对于 JPS 来说，开启的实例调度器 AWS 成为管理非生产环境的关键工具。JPS 使用了实例调度器 AWS 来降低开发成本，并根据团队需求和工作计划管理 EC2 实例。这帮助 JPS 将成本降低了 40%。有关更多信息，请参阅 AWS 案例研究 [《牙买加公共服务高效迁移到云端，使用 AWS 实例调度器将成本降低 40%》](#)。

成本优化场景

以下示例场景有助于说明在上 AWS 使用实例调度器的成本优势。在这种情况下，新加坡一家大型零售公司在 Amazon EC2 上部署了两个 Windows 环境。营销团队利用第一个环境，即工作负载 A，在门店

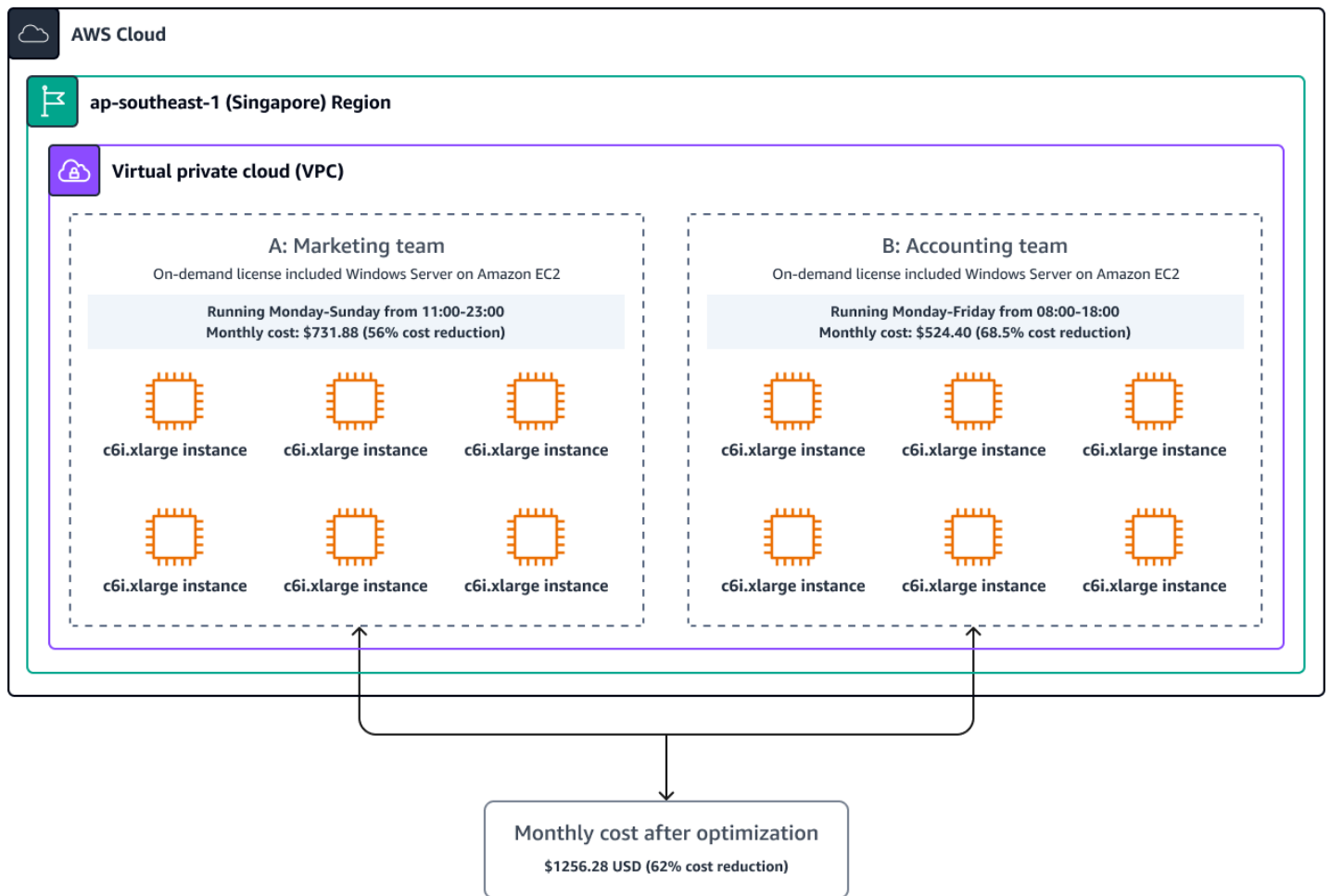
开业时分析实时店内交易。第二种环境称为工作负载 B，是为会计团队保留的，该团队仅在正常工作时间工作。考虑到当前的使用模式，这两种环境的当前运行计划（24/7）并不理想，需要进行优化以降低公司的运营成本。

下图显示了优化前的每月成本。



例如，三月份有 31 天，其中 23 天是工作日。如果营销团队使用实例调度器 AWS 并仅在需要时才运行其实例（即每月 321 小时而不是每月 730 小时），则他们每月有可能节省 932.52 美元。这相当于运营成本降低了 56%。会计团队也可以体验到显著的优势，他们的实例使用时间从每月 730 小时减少到 230 小时。这导致减少了 1,140 美元，即 68.5%。该公司每月总共可以节省 2,072.52 美元（相当于减少 62%），相当于每年 24,870.24 美元。

下图显示了优化后的每月成本。



Note

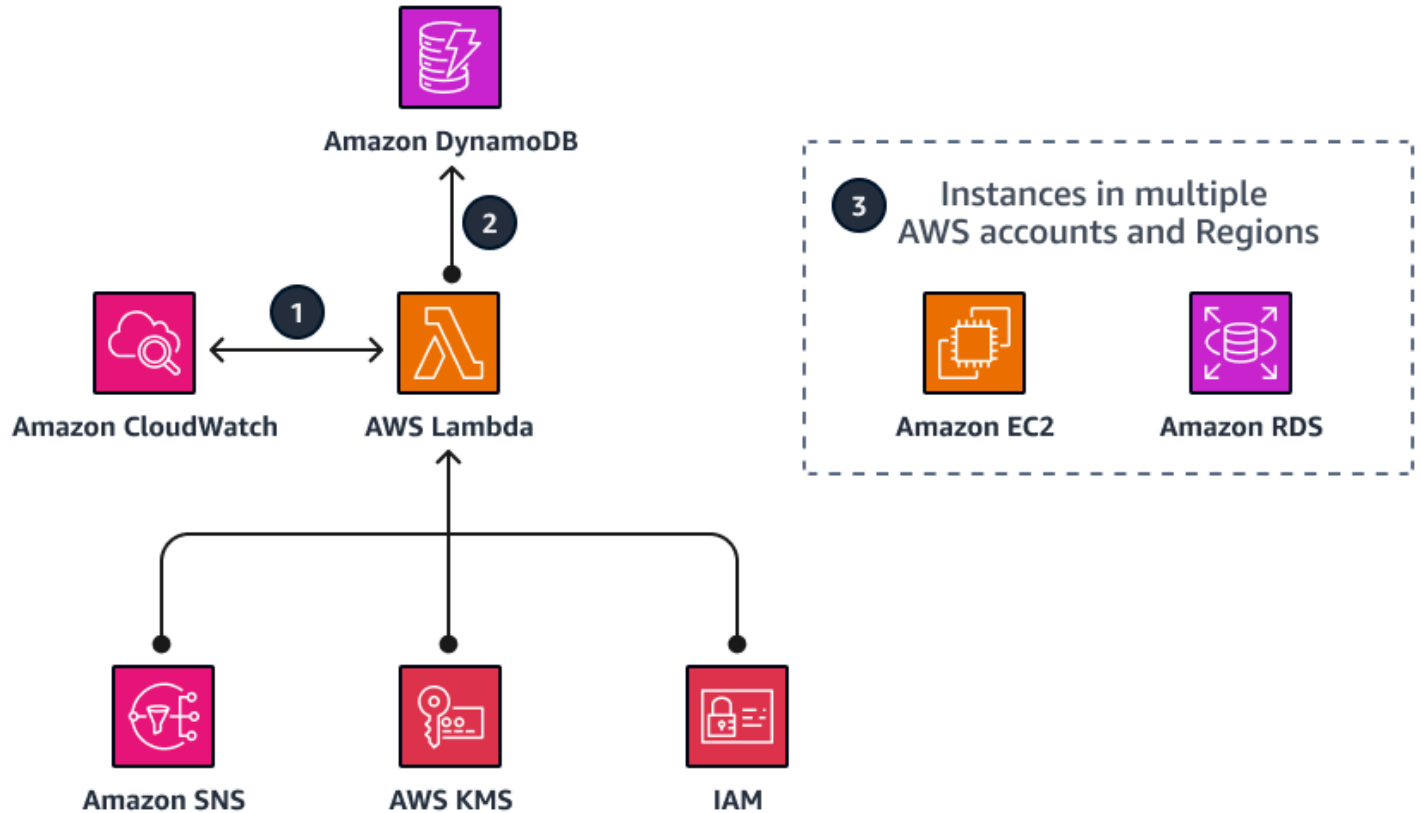
此示例的定价是使用 2023 [AWS Pricing Calculator](#) 年 3 月确定的。

成本优化建议

本节 AWS 根据上一节“成本优化方案”一节中介绍的示例场景，说明如何部署和配置实例计划程序。我们建议您采取以下后续步骤，通过在上 AWS 使用实例计划程序来优化成本：

1. 启动实例调度器堆栈
2. 配置周期
3. 配置日程安排
4. 标记实例。

以下架构图显示了实例调度器堆栈中创建的内容。AWS Cloud



该图显示了以下工作流程步骤：

1. AWS CloudFormation 模板按照您定义的时间间隔设置 Amazon CloudWatch 事件。此事件调用一个函数。AWS Lambda 在配置过程中，您可以定义 AWS 区域 和帐户。您还可以定义一个自定义标签，实例调度器 AWS 使用该标签将计划与适用的 Amazon EC2 实例、Amazon RDS 实例和集群关联起来。
2. 计划配置值存储在 Amazon DynamoDB 中，Lambda 函数在每次运行时都会检索这些值。然后，您可以将自定义标签应用于适用的实例。
3. 在实例计划程序的初始配置过程中，您可以定义一个标签密钥来识别适用的 Amazon EC2 和 Amazon RDS 实例。创建计划时，您指定的名称将用作标记值，用于标识要应用于已标记资源的计划。

启动实例调度器堆栈

本节介绍如何在上 AWS 启动实例调度程序的 CloudFormation 堆栈。

Note

在上运行实例调度器时 AWS 服务 使用的费用由您承担。AWS截至 2023 年 1 月，us-east-1在该地区使用默认设置运行此解决方案的费用约为每月 9.90 美元，如果您有 Lambda 免费套餐的月度使用额度，则费用更低。有关更多信息，请参阅 AWS 解决方案库中《[实例计划程序 AWS 实施指南](#)》中的“成本”部分。

要启动实例调度程序堆栈，请完成以下步骤。

1. 登录[AWS Management Console](#)并选择 [Launch 解决方案](#)（可下载的模板）以启动instance-scheduler-on-aws.template CloudFormation模板。

Note

您也可以[下载模板](#)作为自己实施的起点。

2. 默认情况下，该模板在美国东部（弗吉尼亚州北部）区域启动。要在其他区域启动实例计划程序，请使用控制台导航栏中的区域选择器。

Note

此示例使用亚太地区（新加坡）区域。

3. 在“创建堆栈”页面的“先决条件-准备模板”部分，确认已选中“模板已准备就绪”选项。在“模板来源”部分，确认已选择 Amazon S3 网址选项。
4. 确认 Amazon S3 网址文本框中的模板 URL 是否正确，然后选择“下一步”。
5. 在指定堆栈详细信息页面上，为您的解决方案堆栈分配一个名称。有关命名字符限制的信息，请参阅 AWS Identity and Access Management ([IAM](#)) 文档中的 [IAM 和 STS 限制](#)。本指南中示例的堆栈名称被调用MyInstanceScheduler。

Note

堆栈名称不能超过 28 个字符。

6. 在“参数”下，查看模板的参数并根据需要对其进行修改。
7. 选择 Next(下一步)。在 [配置堆栈选项](#) 页面上，请选择 下一步。

- 在“查看”页面上，查看并确认设置。选中确认模板将创建 IAM 资源的复选框。
- 选择创建以部署堆栈。

配置周期

部署 CloudFormation 模板后，该解决方案会创建一个 DynamoDB 表，其中包含示例周期规则和计划，您可以将其用作创建自己的自定义周期规则和计划的参考。有关时间段配置的示例，请参阅 AWS 文档中的实例计划程序中的[示例计划](#)。

要完成此场景中的步骤，您必须生成与每个工作负载相对应并满足其特定需求的周期。例如：

```
Period 1 (Workload A):
  Name: retail-hours
  Days: Monday to Sunday
  Hours: 1100 - 2300
Period 2 (Workload B):
  Name: office-hours
  Days: Monday to Friday
  Hours: 0800 - 1800
```

要配置周期，请完成以下步骤：

- 登录 [DynamoDB](#) 控制台，确保您所在的区域与启动实例计划程序模板 CloudFormation 的区域相同。AWS
- 在导航窗格中，选择表，然后选择名为的表ConfigTable。
- 选择“浏览表格项目”。
- 要为办公时间创建时段，请为办公时间项目选择时段。
- 在编辑项目页面上，将开始时间的值更改为 0800，将结束时间的值更改为 1800。将工作日的默认值保留不变。

Note

开始时间和结束时间值决定何时启动和停止实例，而工作日值决定该计划适用于一周中的哪几天（本例为星期一至星期五）。

- 选择保存更改。
- 要复制办公时间段并使用它来创建新的零售时间段，请为办公时间项目选择时段。然后，从“操作”菜单中选择“复制项目”。

8. 修改属性以满足您的需求。以下属性用于满足示例场景的要求：

```
type: period
name: retail-hours
begintime: 11:00
description: Retail hours
endtime: 23:00
weekdays: mon-sun
```

9. 选择创建项目。

10. 在 Dynam ConfigTableoDB 中，确定项目列表中列出的您刚刚创建的两个时段。

配置日程安排

在“开启实例调度器”的上下文中 AWS，计划是指应用一个或多个时段以及相关时区。然后，这些计划将作为标签分配给您的实例。本节介绍如何创建两个计划（如下所示）以适应两个示例工作负载的不同时间模式，然后将这些计划与您在上一节中创建的时间段相关联。

```
Schedule 1:
  Name: singapore-office-hours
  Period: office-hours
  Timezone: Asia/Singapore
Schedule 2:
  Name: singapore-retail-hours
  Period: retail-hours
  Timezone: Asia/Singapore
```

要创建和配置计划，请完成以下步骤：

1. 登录 [DynamoDB](#) 控制台，确保您所在的区域与启动实例计划程序模板 CloudFormation 的区域相同。AWS
2. 在导航窗格中，选择表，然后选择名为的表ConfigTable。
3. 选择“浏览表格项目”。
4. 要复制英国办公时间表并使用它为您的办公时间（本示例为新加坡办公时间）创建新的时间表，请为该uk-office-hours 项目选择时间表。然后，从“操作”菜单中选择“复制项目”。
5. 修改属性以满足您的需求。以下属性用于满足示例场景的要求：

```
type: schedule
```



```
name: singapore-office-hours
description: Office hours in Singapore
periods: office-hours
timezone: Asia/Singapore
```

6. 选择创建项目。

7. 重复步骤 4—6，使用以下属性值为新加坡零售时间创建时间表：

```
type: schedule
name: singapore-retail-hours
description: Retail hours in Singapore
periods: retail-hours
timezone: Asia/Singapore
```

8. 在 Dynam ConfigTableoDB 中，确定您创建的两个计划和两个周期。

标记实例。

建立计划后，必须使用标签将计划分配给要使用的特定实例。您可以使用其中的[AWS Resource Groups](#)标签编辑器生成标签并将其分配给您的 Amazon EC2 实例。

1. 登录[AWS Management Console](#)并确保您位于之前启动 CloudFormation 模板的同一区域。
2. 打开 Res [ource Groups 控制台](#)。在导航窗格中，展开“添加标签”，然后选择“标签编辑器”。
3. 在查找要标记的资源部分中，对于区域，选择您的区域。对于资源类型，请选择 Amazon EC2 或 Amazon RDS。此场景侧重于工作负载 A 中的 Amazon EC2 实例。营销团队正在新加坡地区使用工作负载 A。用于此工作负荷的资源已经标有部门密钥和营销值。您可以使用此标签来搜索实例。
4. 选择搜索资源。
5. 从搜索结果列表中选择要包含在计划中的实例，然后选择管理所选资源的标签。
6. 在编辑所有选定资源的标签部分中，选择添加标签，将实例计划程序计划标签添加到您的 EC2 实例。您可以使用与 s chedulea (之前在 DynamoDB 中创建) 匹配的标签键和值。
7. 对于标签密钥，请添加计划。在“标签值”中，输入singapore-retail-hours。
8. 选择检查并应用标签更改。
9. 要将标签应用于您选择的所有 EC2 实例，请选择将更改应用于所有选定实例。
- 10对于您想要应用的任何其他计划，请重复步骤 3—9。

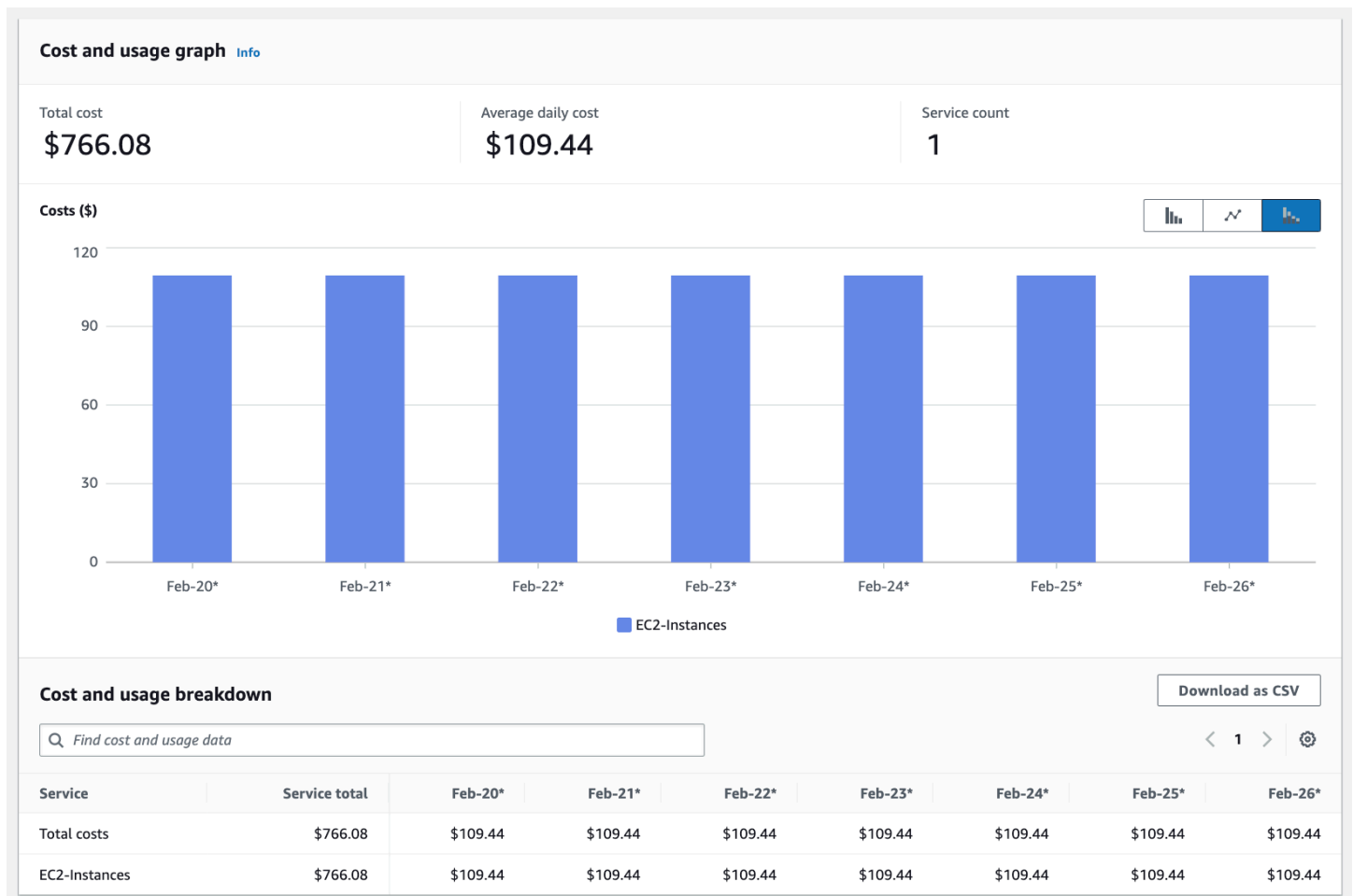
验证结果

我们建议您使用[AWS Cost Explorer](#)来衡量在上 AWS 使用实例调度程序的成本收益。您可以使用 Cost Explorer 执行以下操作：

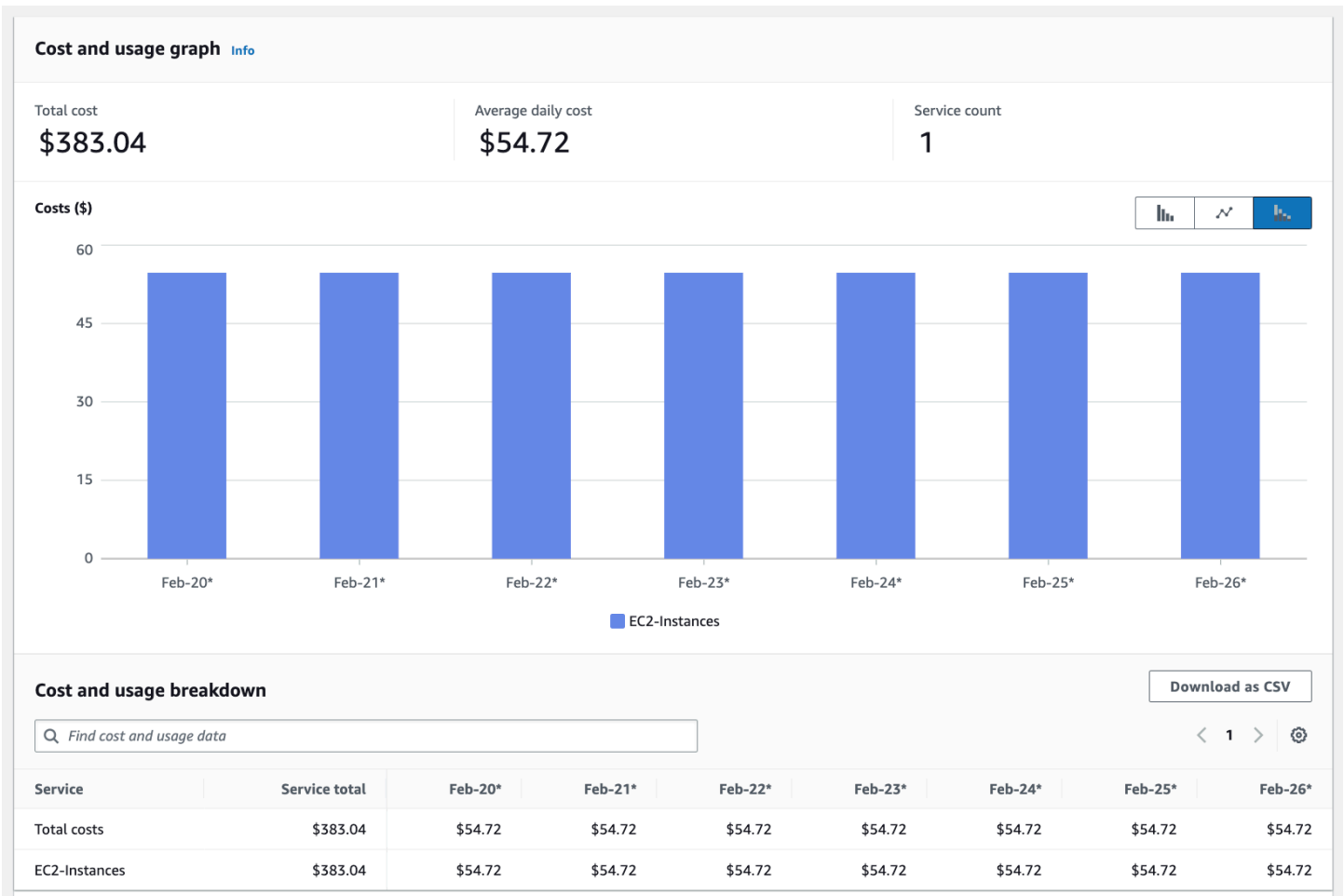
- 查看和分析与您的 EC2 实例相关的成本，包括由实例计划程序管理的实例。
- 按标签筛选您的 Cost Explorer 视图，以便您可以专注于特定的工作负载，并详细了解使用实例调度器实现的成本节约。
- 深入了解使用实例调度程序的财务影响。
- 寻找进一步优化成本的机会，并做出以数据为依据的决策以优化 AWS 支出。

下图说明了在使用实例调度器进行优化前的七天内（星期一至星期日）运行工作负载 A 和工作负载 B 的成本。

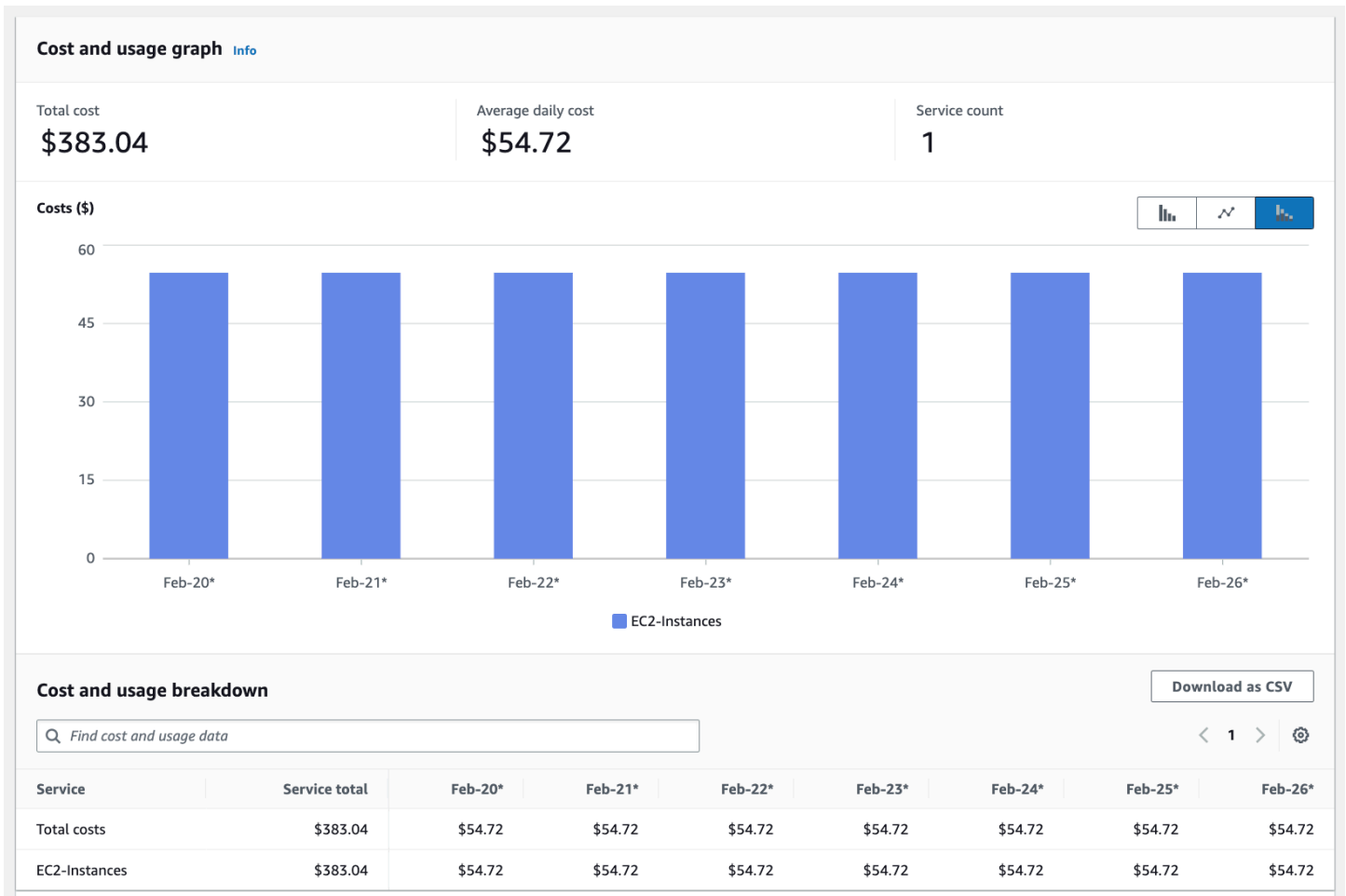
工作负载 A 和 B 的总开支



工作量 A 费用

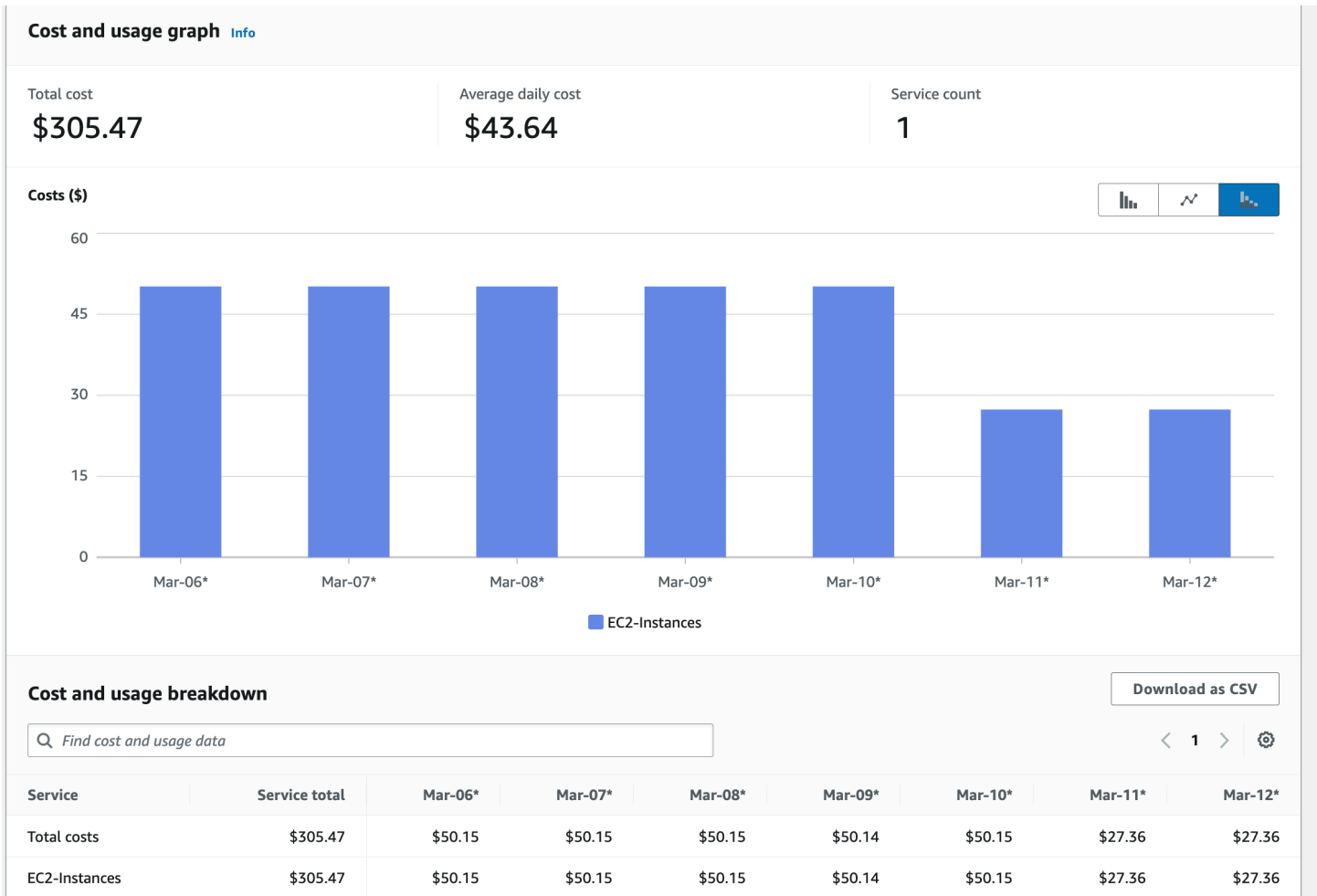


工作量 B 费用

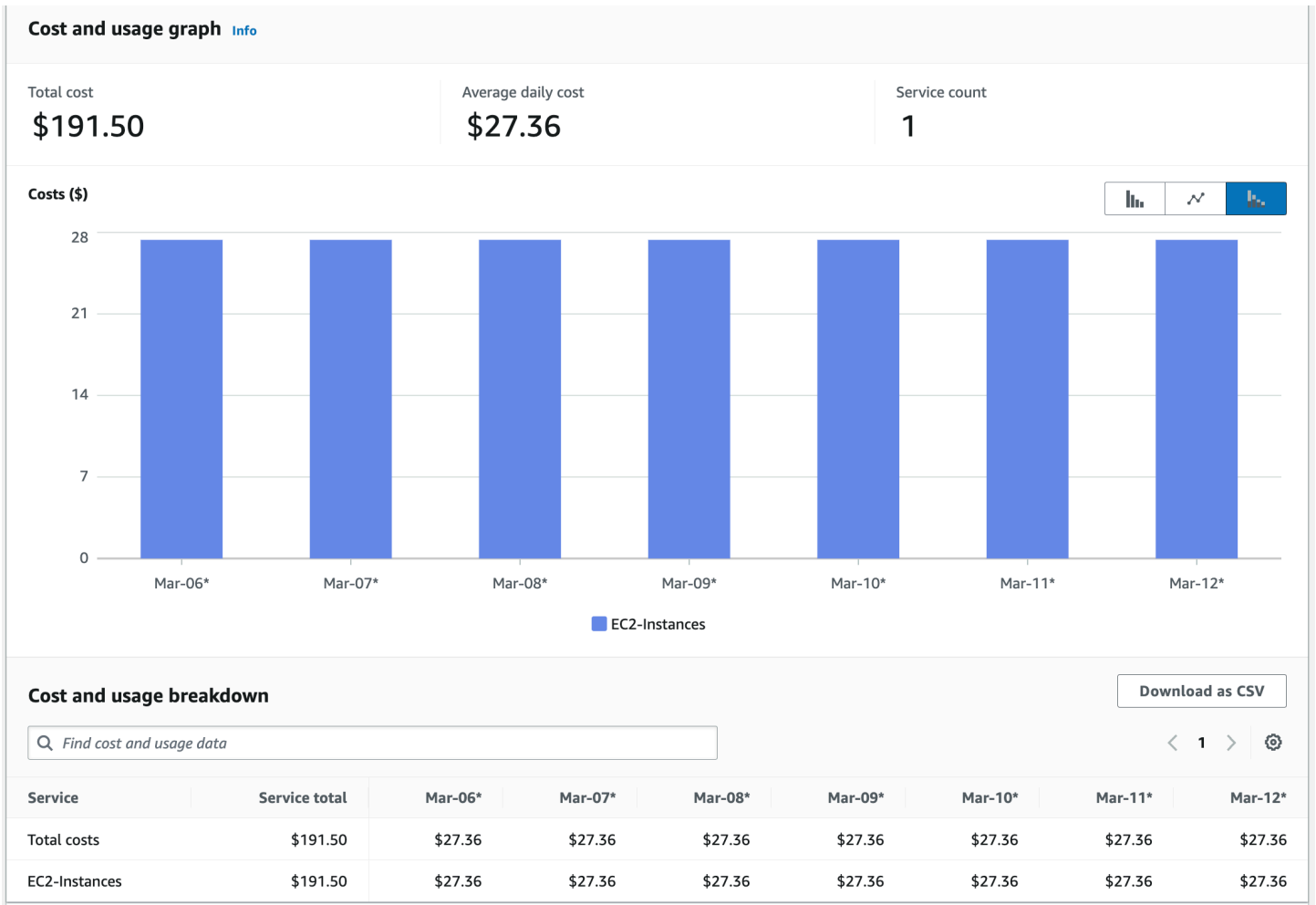


在此场景中，Cost Explorer 显示了通过实施实例调度器而降低的成本。AWS 下图描述了优化后七天（星期一至星期日）内工作负载 A 和工作负载 B 的运营成本。

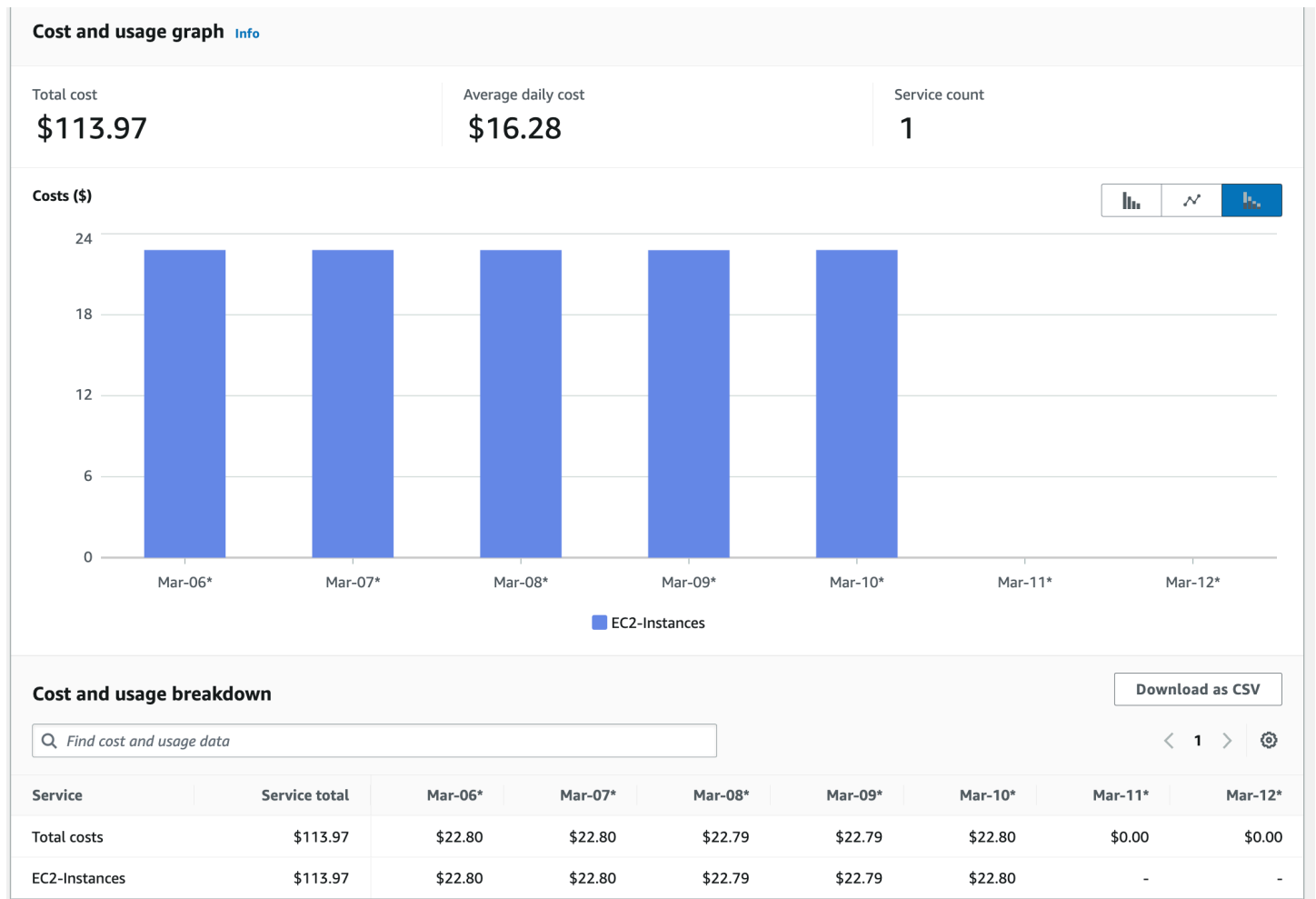
工作量 A 和 B 的总开支



工作量 A 费用



工作量 B 费用



其他资源

- [自动启动和停止 AWS 实例](#) (AWS 文档中的实例调度器)
- [回归基础知识：使用实例计划程序控制 Amazon EC2 和 Amazon RDS 资源成本](#) () YouTube
- [为@@ AWS 资源添加标签](#) (标记 AWS 资源用户指南)
- [使用 AWS Cost Explorer](#) ([AWS Billing and Cost Management 文档](#)) 分析您的成本

大小合适的 Windows 工作负载

概述

合适的尺寸是最有效的节省成本的工具之一。AWS 提供了各种收集正确规模信息的方法，从使用 [AWS 优化和许可评估 \(O AWS LA\)](#) 审查潜在的工作负载，到使用查看现有工作负载 [AWS Cost Explorer](#)。

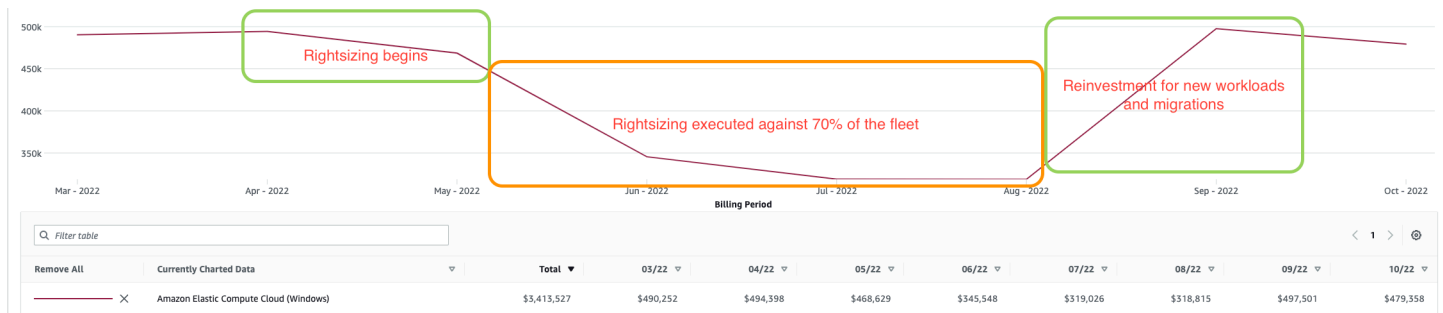
本部分向您展示如何使用 [AWS Compute Optimizer](#) 来识别 Amazon EC2 规模合适的机会。Compute Optimizer 有助于防止以下类型的资源出现过度配置和配置不足的情况：AWS

- [亚马逊弹性计算云 \(Amazon EC2\) 实例类型](#)
- [亚马逊 Elastic Block Store \(亚马逊 EBS\) Volumes](#)
- [亚马逊弹性容器服务 \(Amazon ECS\) Container Service](#) 服务在 AWS Fargate
- [AWS Lambda](#) 基于 [Amazon](#) 提供的使用率数据的函数 CloudWatch

成本优化场景

衡量正确规模的有效性可能具有挑战性，因为可以将正确规模的工作引向特定的应用程序、团队或整个组织。例如，假设一个组织将数千个实例迁移到 AWS，其中 90% 的队列包含 Windows 工作负载。该组织可以使用 Compute Optimizer 来分析其机群，发现其账户中存在严重的超额预留空间，以及 AWS 区域然后，他们可以使用 [AWS Systems Manager Automation](#) 通过多个维护窗口调整车队规模。因此，该组织设法为其 70% 的队列调整了大小合适的实例类型，并节省了 35% 的成本。

以下仪表板说明了本示例组织战略性地实施了 Compute Optimizer 的正确规模建议时在几个月内实现的节省。他们的目标是尽可能高效地操作现有的工作负载，以便在合同快要到期时，恢复从托管数据中心停滞不前的迁移。



成本优化建议

我们建议您采取以下后续步骤，使用 Compute Optimizer 来优化成本：

- 启用 Compute Optimizer
- 为 Windows 节点启用内存指标收集
- 使用 Compute Optimizer 推荐
- 为实例添加标签以调整大小
- 启用成本分配标签以使用 AWS 计费工具

- 使用 AWS Systems Manager 自动化实施合适的尺寸建议
- 考虑其他调整大小的方法
- 在 Cost Explorer 中查看之前和之后的成本

启用 Compute Optimizer

您可以在组织级别或单个账户级别启用 [Compute Optimizer](#)。AWS Organizations 组织范围的配置为所有成员账户的整个队列中的新实例和现有实例提供持续报告。这样一来，正确调整大小就可以成为重复的活动，而不是 point-in-time 活动。

组织级别

对于大多数组织来说，使用 Compute Optimizer 的最有效方式是在组织层面。这为您的组织提供了多账户和多区域可见性，并将数据集中到一个来源供审查。要在组织级别启用此功能，请执行以下操作：

1. 使用具有[所需权限](#)的角色登录您的 Organizations [管理账户](#)，然后选择加入该组织内的所有账户。您的组织必须[已启用所有功能](#)。
2. 启用管理账户后，您可以登录该账户，查看所有其他成员账户，并浏览他们的推荐。

Note

最佳做法是为 Compute Optimizer 配置[委托管理员帐户](#)。这使您能够行使最小权限原则。这样，您就可以最大限度地减少对组织管理帐户的访问权限，同时仍提供对组织范围服务的访问权限。

单一账户级别

如果您的目标账户成本高昂但无权访问，您仍然可以为该账户和 AWS Organizations 地区启用 Compute Optimizer。要了解选择加入流程，请参阅 [Compute Optimizer 文档中的入门](#)。

为 Windows 节点启用内存指标收集

内存指标为 Compute Optimizer 提供了在组织中提出明智的正确规模建议所需的基本指标。这是因为在提供建议之前要对 CPU、内存、网络和存储进行分析。

要将内存指标从 Windows EC2 实例传递给 Compute Optimizer，您必须启用 CloudWatch 代理并将内存指标配置为每 60 秒收集一次。将内存指标与一起使用不会产生额外费用 CloudWatch。

启用 CloudWatch 代理并配置内存指标

下载 [ComputeOptimize.yml 文件](#)。您可以使用此文件为账户中的所有实例启用内存收集。模板文件生成以下组件：

- [AWS Systems Manager 参数存储](#)-用于存储收集内存指标所需的 CloudWatch 代理配置。
- AWS Identity and Access Management (IAM) 角色 AWS Systems Manager 附带 [AWS 托管策略 — 此角色适用于 Systems Manager Automation](#) 文档。
- [AWS Systems Manager documents](#) — 这将安装和配置 CloudWatch 代理（替换任何现有 CloudWatch 配置）。
- [AWS Systems Manager 状态管理器](#) 关联 — 这使得 Systems Manager 文档能够在您账户中的所有实例上运行。

Important

运行此模板会覆盖实例上的任何现有 CloudWatch 配置。

接下来执行以下操作：

1. 登录 AWS Management Console 并打开 [CloudFormation 控制台](#)。
2. 在导航窗格中，选择 Stacks（堆栈）。
3. 选择创建堆栈，然后选择使用现有的资源（导入资源）。
4. 选择下一步。
5. 对于模板资源，选择上传模板文件。
6. 选择文件，然后上传 ComputeOptimize.yml 文件。
7. 选择下一步。
8. 在指定堆栈详细信息页面上，在堆栈名称中输入堆栈的名称，然后选择下一步。
9. 在识别资源页面上，输入要导入的资源的标识符值。
10. 选择“导入资源”。
11. 部署堆栈后，选择 Outputs 选项卡以查找关联的密钥、值和描述。

监控协会的进展

1. CloudFormation 堆栈部署完成后，打开 [Systems Manager 控制台](#)。

2. 在导航窗格的“节点管理”部分，选择状态管理器。
3. 在关联页面上，选择您的协会的关联 ID。
4. 选择 Execution history (执行历史记录) 选项卡。
5. 在“执行 ID”列中，选择您的协会的关联 ID。状态应为“成功”。

在中查看指标 CloudWatch

我们建议您至少等待五分钟，让指标填充。CloudWatch

1. 打开[CloudWatch控制台](#)。
2. 在导航窗格中，展开“指标”部分，然后选择“所有指标”。
3. 确认指标显示在 CWAgent 命名空间下。

Note

要将设置应用于任何新实例，请重新运行关联。

使用 Compute Optimizer 推荐


举一个例子，该示例侧重于在单个账户和单个区域内进行正确的规模更改。在此示例中，Compute Optimizer 已在组织级别为所有账户启用。请记住，正确调整大小是一个破坏性的过程，在大多数情况下，该过程是由应用程序所有者在几周的计划维护窗口内精确执行的。

如果您从组织的管理帐户中导航到 Compute Optimizer (如以下步骤所示)，则可以选择要调查的帐户。在此示例中，该us-east-1地区的单个账户中有六个实例在运行。所有六个实例都已超额配置。目标是根据 Compute Optimizer 的建议调整实例的大小。

识别过度配置的实例并导出建议的详细信息

1. 登录 AWS Management Console 并打开 [Compute Optimizer 控制台](#)。
2. 在导航窗格中，选择控制面板。
3. 在控制面板页面的搜索框中，输入 region=美国东部 (弗吉尼亚北部)。然后，输入 findings=OverProvisioned。这些筛选器允许您查看该地区所有超额配置的us-east-1实例。
4. 要查看针对超额配置的 EC2 实例的详细建议，请向下滚动到 EC2 实例卡片，然后选择查看建议。
5. 选择“导出”并保存文件以备将来使用。

6. 对于 S3 存储桶，输入要作为导出文件目标的 Amazon S3 存储桶的名称。

 Note

要保存建议以备将来查看，您必须有一个可用的 S3 存储桶供每个区域的 Compute Optimizer 写入内容。有关更多信息，请参阅 Compute Optimizer AWS Compute Optimizer 文档中的[亚马逊 S3 存储桶策略](#)。

7. 在“导出筛选条件”部分中，选中“包括组织中所有成员账户的推荐”复选框。

8. 对于资源类型，选择 EC2 实例。


9. 在“要包括的列”部分中，选中“全选”复选框。

10. 选择导出。

根据建议选择实例

实例建议基于 Compute Optimizer 收集和分析的性能指标。必须了解实例上运行的工作负载，以确保您选择最佳实例。[此示例假设您可以从最新一代的 Amazon EC2 R6i、R5 和 T3 实例中进行选择](#)。T3 实例具有突发性，并且具有较低的网络带宽能力。R5 和 R6 实例的每小时成本相同，而且几乎相同。但是，R6 实例具有更高的网络带宽容量，采用最新一代的英特尔处理器，并提供与 R5 相同的计算占用空间。在此示例中，R6 是调整大小的最佳选择。

1. 在 [Compute Optimizer 控制台](#) 中，从导航栏中选择适用于 EC2 实例的建议。本页显示了当前实例类型与替换该实例的推荐选项的比较。
2. 要获取您想要调整大小的实例的 ID，请从中的管理账户打开 [Amazon S3 控制台](#) AWS Organizations。
3. 在导航窗格中，选择 Buckets，然后选择用于存储导出结果的存储桶。
4. 在对象选项卡上，从对象列表中选择您的导出文件，然后选择下载。
5. 要从文件中提取实例信息，可以使用 Microsoft Excel 中“数据”选项卡上的“文本到列”按钮。

 Note

实例 ID 用亚马逊资源名称 (ARN) 表示。请务必将分隔符设置为 “/” 并提取实例 ID。或者，您可以编写脚本或使用集成开发环境 (IDE) 来修剪 ARN。

6. 在 Excel 中，筛选查找结果列以仅显示 OVER_PROVISIONED 实例。这些是你为正确调整大小而瞄准的实例。

7. 将实例 ID 保存在文本编辑器中，便于日后访问。

为实例添加标签以调整大小

为工作负载添加标签是组织资源的强大工具。AWS 标签使您能够精细地了解成本并促进退款。有关向资源添加标签的策略和方法的更多信息，请参阅 AWS 白皮书为 AWS 资源添加[标签 AWS 的最佳实践](#)。在此示例中，您可以使用[AWS 标签编辑器](#)对您希望在维护时段内调整大小的超额配置实例进行标签调整。您也可以使用此标签来查看变更前后的费用。

1. 登录 AWS Management Console 并打开包含要调整大小的实例的账户的[AWS Resource Groups 控制台](#)。
2. 在导航栏的“添加标签”部分，选择“标签编辑器”。
3. 对于区域，请选择您的目标区域。
4. 对于资源类型，请选择 AWS::EC2::Instance。
5. 选择搜索资源。
6. 在资源搜索结果页面上，选择要调整大小的所有实例，然后选择管理所选资源的标签。
7. 选择 Add tag (添加标签)。
8. 对于“标签密钥”，输入“调整大小”。在“标签值”中，输入“启用”。然后，选择“查看”并应用标签更改。

Note

您可以添加其他元数据，例如团队或业务部门，以便日后在 Cost Explorer 中进行筛选。

在创建用户定义的标签并将其应用于资源后，这些标签最多可能需要 24 小时才能显示在成本分配标签页面上进行激活。选择要激活的标签后，标签可能还需要 24 小时才能生效。

对于高级用户，您可以在目标账户和区域[AWS CloudShell](#)内使用来标记多个实例。例如：

```
bash
#!/bin/bash
# Set variables
TAG_KEY="rightsizing"
TAG_VALUE="type-m5"
# Get a list of instance IDs
INSTANCE_IDS=$(aws ec2 describe-instances --query
"Reservations[].Instances[].InstanceId" --output text)
```

```
# Loop through each instance ID and add the tag
for INSTANCE_ID in $INSTANCE_IDS; do
  aws ec2 create-tags --resources $INSTANCE_ID --tags Key=$TAG_KEY,Value=$TAG_VALUE
done
```

启用成本分配标签以使用 AWS 计费工具

我们建议激活用户定义的成本分配标签。这样就可以在 AWS 计费工具（例如，Cost Explorer 和）中识别和筛选 Right sizing 标签。AWS 成本和使用情况报告如果您不启用此功能，则标签筛选选项和数据将不可用。有关使用成本分配标签的信息，请参阅 AWS Billing and Cost Management 文档中的[激活用户定义的成本分配标签](#)。

1. 登录 AWS Management Console 并打开[AWS Billing 控制台](#)。
2. 在导航窗格的账单部分，选择成本分配标签。
3. 在用户定义的成本分配标签选项卡上，输入 Rightsizing。
4. 选择 Rightsizing 标签密钥，然后选择“激活”。

24 小时后，标签应显示在 Cost Explorer 中。

使用 Systems Manager Automation 实施合适的规模建议

调整大小是一种需要停止和启动实例的场景。在这种情况下，您可能需要在维护窗口中处理这种中断，并且需要不同的团队来处理自己的大小调整。在更改实例类型之前，请查看 Amazon EC2 文档中[兼容实例类型的注意事项](#)。

本节中的示例步骤使用名为 [AWS-](#) 的 Systems Manager Automation 文档，为每个账户和地区实施合适的规模建议 ResizeInstance。这种方法对于大多数组织来说都是典型的，因为大多数组织出于不同的目的需要不同的实例类型。您也可以使用相同的 AWS-ResizeInstance 自动化文档来定位单账户和多账户部署。

1. 登录 AWS Management Console 并打开 [Systems Manager 控制台](#)。
2. 在导航窗格的“共享资源”部分，选择“文档”。
3. 在搜索栏中输入 AWS-ResizeInstance，然后 ResizeInstance 从搜索结果中选择 AWS-。
4. 选择执行自动化。
5. 在执行自动化运行手册页面上，选择简单执行。
6. 在输入参数部分中，输入 InstanceId 和 InstanceType。保留其余的默认值。
7. 选择 Execute，然后等待自动化完成更改实例类型的步骤。

考虑其他调整大小的方法

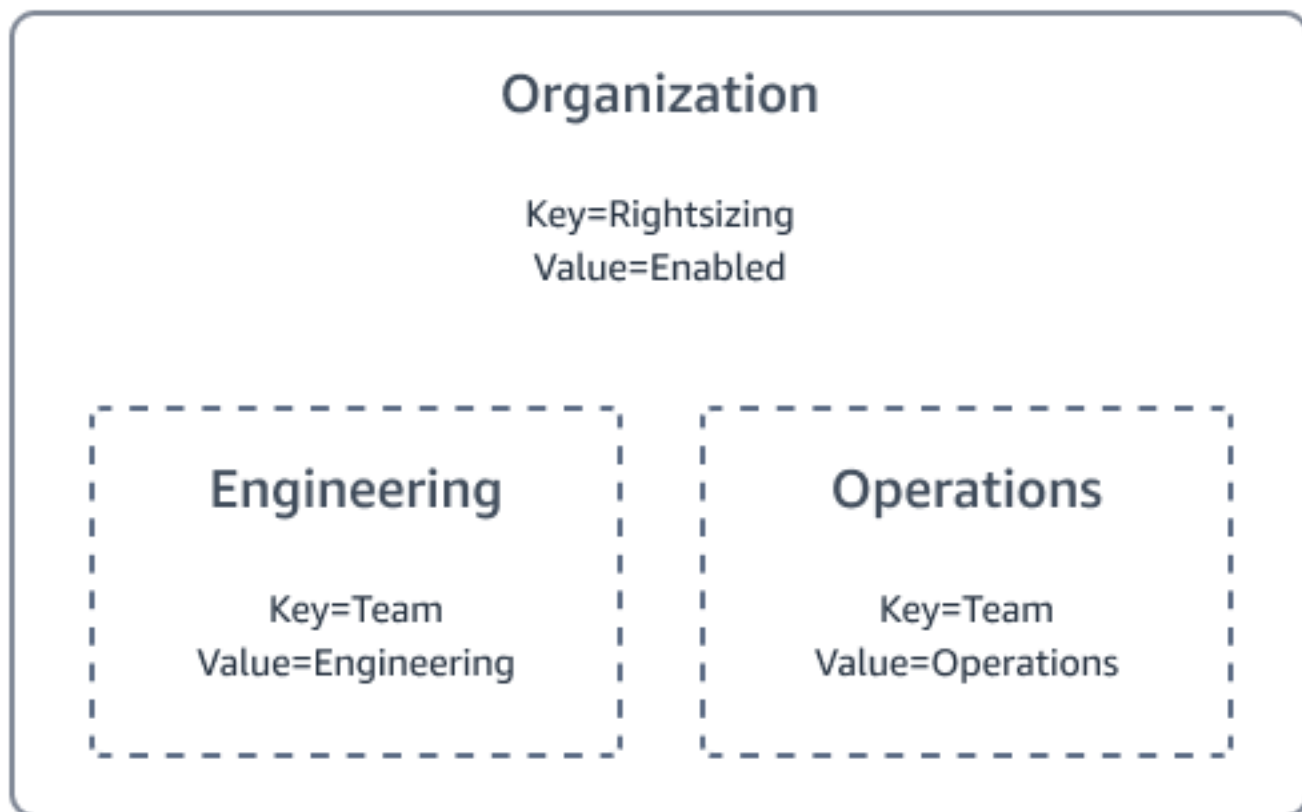
如果您使用启动模板部署实例，则可以将启动模板更新为大小合适的实例类型，然后刷新实例，用大小合适的版本替换实例。

如果您计划在多个账户和区域中使用正确的规模调整流程，则必须创建自定义 Systems Manager Automation 文档。本文档允许您作为参数输入多个实例，并使目标实例移动到相同的目标实例类型（例如，无论源实例类型如何，所有实例都将过渡到 t3a.medium）。

在 Cost Explorer 中查看之前和之后的成本

正确调整资源大小后，您可以使用 Right sizing 标签使用 Cost Explorer 来显示之前和之后的成本。回想一下，您可以使用[资源标签](#)来跟踪成本。通过使用多层标签，您可以精细地了解成本。在本指南中介绍的示例中，Rightsizing 标签用于将通用标签应用于所有目标实例。然后，使用团队标签来进一步组织资源。下一步是引入应用程序标签，以进一步显示运营特定应用程序的成本影响。

下图显示了组织的标签结构。



以运营团队拥有的生产 Web 服务器规模合适的企业为例。在 Cost Explorer 中，Right sizing 标签设置为启用，团队标签设置为操作。在此示例中，正确调整规模的工作可将运营成本从每小时 0.89 美分降低到 0.28 美分。假设每月 744 小时，则在正确调整之前，每年的费用为 7,945.92 美元。调整好尺

寸后，年度成本降至2,499.84美元。这意味着每年的工作量成本降低了68.5%。想象一下这对大型组织的影响。请记住，这是在示例环境中完成的，实例大多处于空闲状态。在生产环境中，您可以看到节省10-35%。

现在，考虑一下正确调整工程团队拥有的生产堡垒主机的规模会产生什么影响。在 Cost Explorer 中，Right sizing 标签设置为启用，团队标签设置为工程。在此示例中，正确调整大小的工作可将成本从每小时 0.75 美分降低到 0.44 美分。假设每月744小时，则在进行正确调整之前，每年的费用为 6,696.00美元。调整好尺寸后，年度成本降至3,928.32美元。

如果您使用多个标签，则可以将数据筛选为精细的成本明细。在此示例中，Team 标签降低了噪音，因此您可以查看团队级别的冲击。由于 Right sizing 标签已启用，因此您还可以筛选具有该标签且值为启用或不存在任何值的任何实例。这可以让您全面了解您的正确规模调整工作，尤其是在Cost Explorer级别的管理账户（付款人）中查看时。通过此视图，您可以查看所有账户和实例。

举一个单一账户级别的示例，其中 Rightsizing 标签设置为启用。运营成本从每小时1.64美元降至每小时0.72美分。假设每月744小时，则在进行正确调整之前，每年的费用为14,641.92美元。调整好尺寸后，年度成本降至6,428.16美元。这意味着该账户的计算成本降低了56%。

在开始合适的尺码之旅之前，请考虑以下几点：

- AWS 提供了许多降低成本的选项。这包括 [AWS OLA](#)，AWS 它会在迁移到本地实例之前先检查您的本地实例 AWS。O AWS LA 还为您提供合适的规模建议和许可指南。
- [在购买 Savings Plans 之前，请完成所有合适的规模。](#) 这可以帮助您避免超额购买 Savings Plans 承诺。

建议

我们建议采取以下后续步骤：

1. 查看您的现有格局，并考虑将 Amazon EBS gp2 卷转换为 gp3 卷。
2. 查看 [Savings Plans](#)。

其他资源

- [AWS Compute Optimizer](#) (AWS 文档)
- [标记 AWS 资源的最佳实践](#) (AWS 白皮书)
- [如何从 AWS Compute Optimizer 您的 AWS Organizations\(YouTube\) AWS Trusted Advisor 中收集数据](#)

- [优化性能和降低许可成本：利用 AWS Compute Optimizer Amazon EC2 SQL Server 实例](#) (AWS 博客上的 Microsoft 工作负载)

为 Windows 工作负载选择正确的实例类型

概述

与本地环境相比，在云端运行的工作负载有一个显著的区别，那就是过度配置的做法。在购买供本地使用的物理硬件时，您需要的资本支出预计将持续预先确定的期限，通常为 3-5 年。为了适应硬件生命周期内的预期增长，购买硬件时使用的资源超过了工作负载当前所需的资源。因此，物理硬件的过度配置往往远远超出了实际工作负载的需求。

虚拟机 (VM) 技术已成为利用剩余硬件资源的有效手段。管理员过度配置了 vCPU 和 RAM 的虚拟机，从而使虚拟机管理程序能够通过向每台虚拟机分配未使用的资源来管理繁忙和空闲服务器之间的物理资源使用情况。在管理虚拟机时，分配给每个虚拟机的 vCPU 和 RAM 资源更多地起到资源调控器的作用，而不是实际使用情况的指标。虚拟机资源过度分配很容易超过可用计算资源的三倍。

[Amazon Elastic Compute Cloud \(Amazon EC2\)](#) 避免在底层硬件上过度配置虚拟机，因为这是不必要的。云计算是一种运营费用，而不是资本支出，您只需为实际使用的资源付费。如果您的工作负载将来需要更多资源，请在实际需要时进行配置，而不是先发制人地进行配置。

有数百种选项可供选择正确的 [Amazon EC2 实例类型](#)。如果您计划将 Windows 工作负载迁移到云端，请 AWS 提供 O [AWS L A](#)，以帮助您更好地了解当前的工作负载并举例说明其运行性能 AWS。AWS OLA 分析旨在将合适的 EC2 实例类型和大小与您的实际本地使用量相匹配。

如果您已经有工作负载在 Amazon EC2 上运行，并且正在寻求成本优化策略，那么本指南的这一部分将帮助您确定 Amazon EC2 实例之间的区别及其对典型 Windows 工作负载的适用性。

成本优化建议

要优化 EC2 实例类型的成本，我们建议您执行以下操作：

- 为您的工作负载选择合适的实例系列
- 了解处理器架构之间的价格差异
- 了解各代 EC2 的性价比差异
- 迁移到较新的实例
- 使用可突发实例

为您的工作负载选择合适的实例系列

为您的工作负载选择正确的实例系列非常重要。

Amazon EC2 实例分为以下几组：

- 通用型
- 计算优化型
- 内存优化型
- 加速计算型
- 存储优化
- HPC 已优化

大多数 Windows 工作负载都属于以下类别：

- 通用型
- 计算优化型
- 内存优化型

为了进一步简化这一点，可以考虑在每个类别中使用基准 EC2 实例：

- 计算优化 — C6i
- 一般用途 — m6i
- 内存优化 — R6i

上一代 EC2 实例在处理器类型上存在细微差异。例如，C5 计算优化型实例的处理器比 M5 通用实例或 R5 内存优化型实例更快。最新一代的 EC2 实例 (C6i、m6i、R6i、C6a、m6a 和 R6a) 在不同的实例系列中都使用相同的处理器。由于处理器在最新一代实例中保持一致，因此实例系列之间的价格差异现在更多地取决于 RAM 的大小。实例的 RAM 越多，其成本就越高。

以下示例说明了在该地区 us-east-1 运行的基于英特尔的 4 vCPU 实例的每小时定价。

实例	vCPU	RAM	每小时价格
c6i.xlarge	4	8	0.17 美元

实例	vCPU	RAM	每小时价格
m6i.xlarge	4	16	0.19 美元
r6i.xlarge	4	32	0.25 美元

Note

定价基于该us-east-1地区的按需每小时定价。

可突发实例

虽然关闭未使用的计算资源以避免产生费用是云计算中的最佳实践，但并非所有工作负载都可以在每次需要时关闭和打开。有些工作负载会长时间处于闲置状态，但必须全天 24 小时可以访问。

Burstable 实例 (T3) 提供了一种在保持低计算成本的同时，将尖峰或低利用率的工作负载全天保持在线。可突发的 EC2 实例拥有该实例在短时间内可以使用的最大 vCPU 资源量。这些实例使用基于[可突发的 CPU 积分](#)的系统。这些积分是在全天闲置期间累积的。Burstable 实例提供不同的 vCPU 与 RAM 的比率，这使得它们在某些情况下可以替代计算优化实例，而在另一些情况下，它们可以替代其他通用实例。

以下示例说明了在该地区运行的 T3 实例（即可突发实例）的每小时定价。us-east-1

实例	vCPU	内存 (GB)	每小时价格
t3.nano	2	0.5	0.0052
t3.micro	2	1	0.0104
t3.small	2	2	0.0208
t3.medium	2	4	0.0416
t3.large	2	8	0.0832
t3.xlarge	4	16	0.1664 美元
t3.2xlarge	8	32	0.3328

Note

定价基于该us-east-1地区的按需每小时定价。

了解处理器架构之间的价格差异

自 EC2 实例问世以来，[英特尔](#)处理器一直是 EC2 实例的标准。前几代 EC2 实例（例如 C5、M5 和 R5）并未将英特尔指定为处理器架构（因为它是默认架构）。新一代的 EC2 实例，例如 C6i、M6i 和 R6i，都包含一个“i”，表示使用英特尔处理器。

处理器架构注释的变化是由于引入了其他处理器选项。与英特尔最具可比性的处理器是 [AMD](#)（用“a”表示）。AMD EPYC 处理器使用相同的 x86 架构，性能与英特尔处理器相似，但价格更低。如以下定价示例所示，与英特尔同类实例相比，AMD EC2 实例的计算成本折扣约为 10%。

英特尔实例	每小时价格	AMD 实例	Price	% 差异
c6i.xlarge	0.17 美元	c6a.xlarge	0.153 美元	10%
m6i.xlarge	0.192	m6a.xlarge	0.1728 美元	10%
r6i.xlarge	0.252 美元	r6a.xlarge	0.2268 美元	10%

Note

定价基于该us-east-1地区的按需每小时定价。

第三个主要的处理器架构选项是 EC [AWS 2 实例上的 Graviton 处理器](#)（用“g”表示）。Graviton 处理器由设计 AWS，在 Amazon EC2 上具有最佳的性价比。目前的 Graviton 处理器不仅比英特尔处理器便宜 20%，而且性能也提高了 20% 或更高。预计下一代 Graviton 处理器将进一步扩大这种性能差异，测试显示性能将再提高 25%。

Windows 服务器无法在基于 ARM 架构的 Graviton 处理器上运行。实际上，Windows 服务器只能在 x86 处理器上运行。尽管在 Windows Server 上使用基于 Graviton 的实例无法将价格性能提高 40%，但你仍然可以在特定的微软工作负载上使用 Graviton 处理器。例如，[较新版本的 .NET 可以在 Linux 上运行](#)。这意味着这些工作负载可以使用 ARM 处理器，并受益于更快、更实惠的 Graviton EC2 实例。

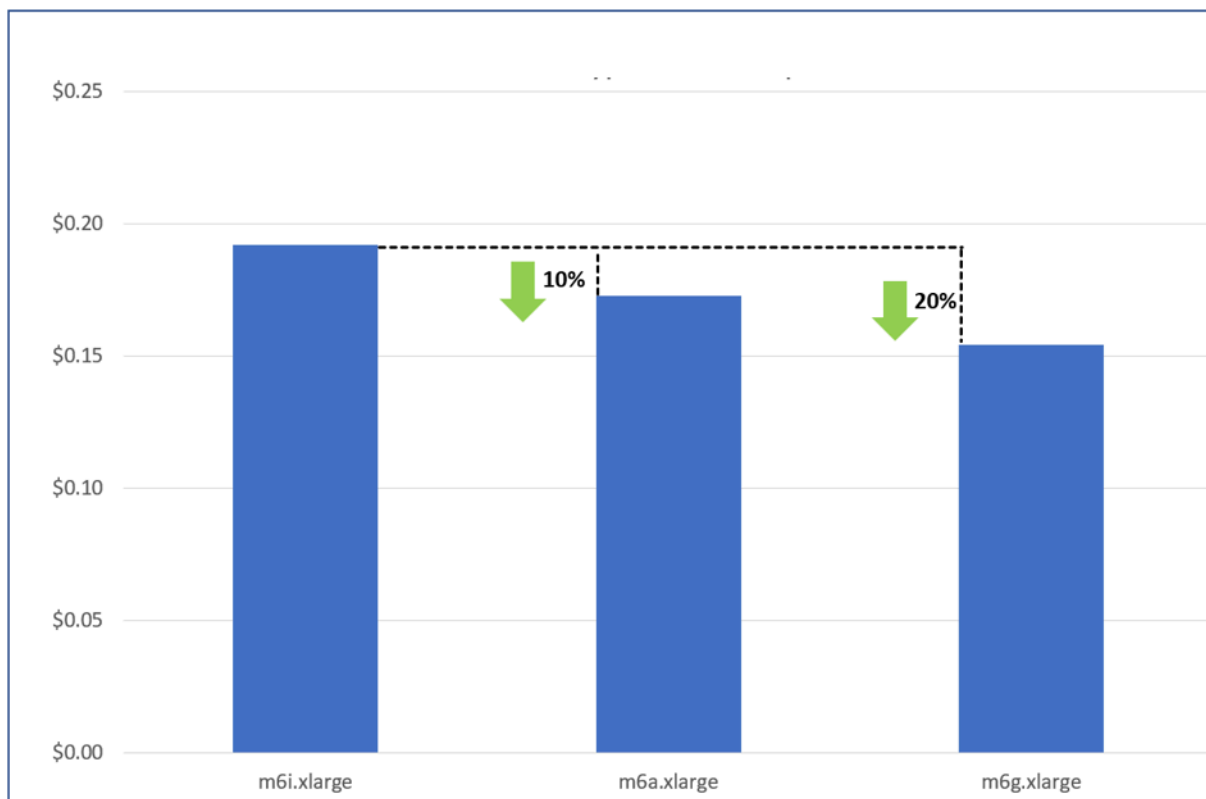
以下示例说明了在该地区运行的 Graviton 实例的每小时定价。us-east-1

英特尔实例	每小时价格	Graviton 实例	每小时价格	% 差异
c6i.xlarge	0.17 美元	c6g.xlarge	0.136	20%
m6i.xlarge	0.192	m6g.xlarge	0.154	20%
r6i.xlarge	0.252 美元	r6g.xlarge	0.2016	20%

Note

定价基于该us-east-1地区的按需每小时定价。

下图比较了 M 系列实例的价格。



了解 EC2 各代之间的性价比差异

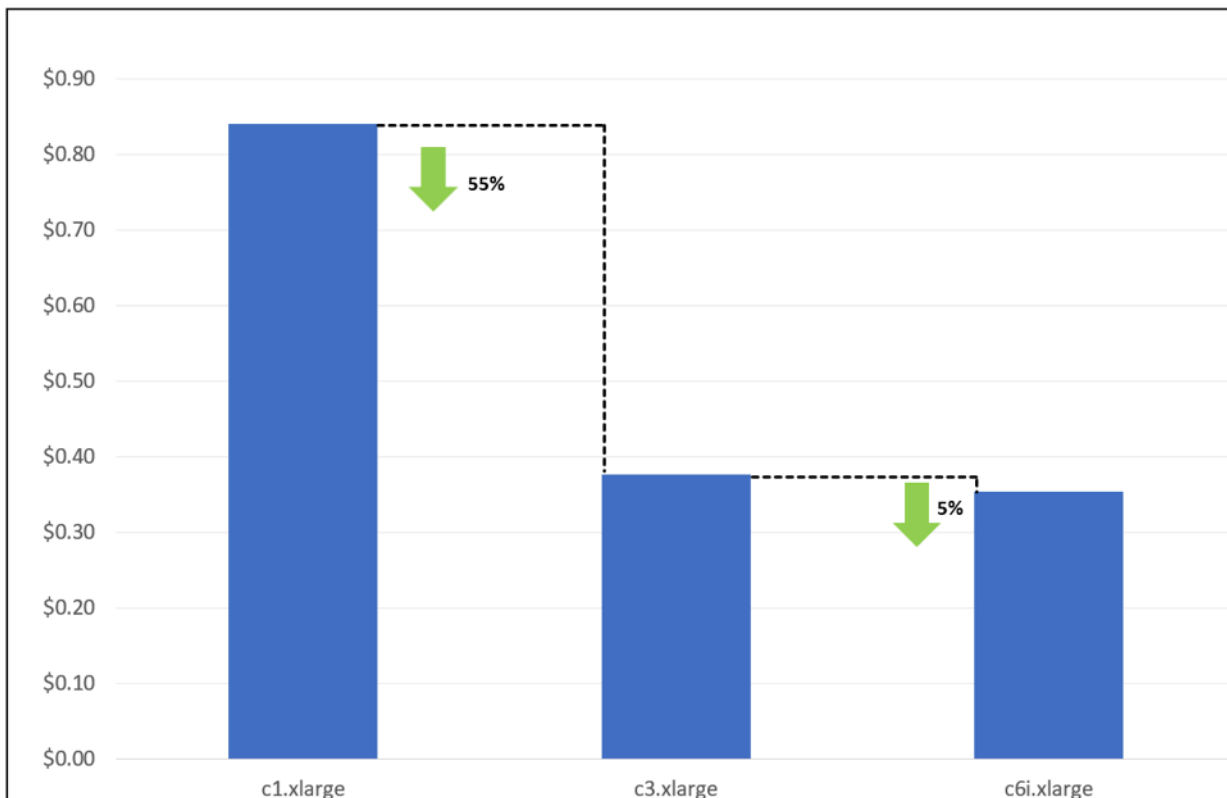
Amazon EC2 最稳定的特征之一是，每款新一代产品都比其前代产品具有更好的性价比。如下表所示，新一代 EC2 实例的价格会随着后续版本的增加而降低。

计算优化型实例	每小时价格	通用实例	每小时价格	内存优化实例	每小时价格
c1.xLarge	0.52 美元	m1.xLarge	0.35 美元	r1.xlarge	不适用
c3.xLarge	0.21 美元	m3.xLarge	0.266 美元	r3.xlarge	0.333 美元
c5.xLarge	0.17 美元	m5.xLarge	0.192	r5.xlarge	0.252 美元

Note

定价基于该us-east-1地区的按需每小时定价。

下图比较了不同代 C 系列实例的成本。



但是，第 6 代实例的价格与第 5 代实例的价格相同，如下表所示。

计算优化型实例	每小时价格	通用实例	每小时价格	内存优化实例	每小时价格
c5.xLarge	0.17 美元	m5.xLarge	0.192	r5.xlarge	0.252 美元
c6i.xLarge	0.17 美元	m6i.xLarge	0.192	r6i.xlarge	0.252 美元

Note

定价基于该us-east-1地区的按需每小时定价。

尽管成本相同，但由于处理器更快、网络吞吐量更高，以及亚马逊弹性区块存储 (Amazon EBS) Elastic Block Store (Amazon EBS) 吞吐量和IOPS的增加，因此具有卓越的性价比。

最显著的性价比改进之一是 [X2i](#) 实例的增强。与上一代实例相比，这一代实例的性价比最高可提高 55%。如下表所示，x2iedn 在各个性能方面都得到了改进（价格均与上一代产品相同）。

实例	每小时价格	vCPU	RAM	处理器速度	实例存储	联网	亚马逊 EBS 吞吐量	EBS IOPS
x1e.2xlarge	1.66 美元	8	244	2.3 GHz	237GB 固态硬盘	10 Gbps	125 Mb/s	7400
x1iedn.2xlarge	1.66 美元	8	256	3.5 GHz	240GB NVMe 固态硬盘	25 Gbps	2500 Mb/s	65000

Note

定价基于该us-east-1地区的按需每小时定价。

示例 应用场景

以一家分析公司为例，该公司跟踪送货车辆并希望提高其 SQL Server 性能。在一家 MACO 中小企业审查了该公司的性能瓶颈后，该公司从 x1e.2xlarge 实例过渡到 x2iedn.xlarge 实例。新的实例大小较小，但是 x2 实例的增强功能可通过使用缓冲池扩展来提高 SQL Server 性能和优化。这使该公司能够从 SQL Server 企业版降级到 SQL Server 标准版。它还使该公司能够将其 SQL Server 许可从 8 个 vCPU 减少到 4 个 vCPU。

优化前：

Server	EC2 实例	SQL Server 版本	月度成本
proddb1	x1e.2xlarge	企业	3,918.64 美元
proddb2	x1e.2xlarge	企业	3,918.64 美元
总计			7,837.28 美元

优化后：

Server	EC2 实例	SQL Server 版本	月度成本
proddb1	x2iedn.xlarge	Standard	1,215.00 美元
proddb2	x2iedn.xlarge	Standard	1,215.00 美元
总计			2,430.00 美元

总而言之，从 x1e.2xlarge 实例更改为 x2iedn.xlarge 实例使公司在生产数据库服务器上每月可节省 5,407 美元。这使工作负载的总成本降低了 69%。

Note

定价基于该us-east-1地区的按需每小时定价。

迁移到较新的实例

[老一代的 Amazon EC2 在 Xen 虚拟机管理程序上运行，而新一代则在 Nitro 系统上 AWS 运行。](#) Nitro System 将主机硬件的几乎所有计算和内存资源都提供给您的实例。这样可以提高整体性能。[从 Xen 迁移到基于 Nitro 的实例时，需要考虑一些特殊因素。](#) 例如，[AWS Windows AMI 使用微软安装媒体使用的默认设置和自定义项进行配置。](#) 自定义项包括支持最新一代实例类型（在 [Nitro System 上构建的实例](#)）的驱动程序和配置。

如果您要从 2018 年 8 月之前创建的自定义 Windows AMI 或亚马逊提供的 Windows AMI 启动实例，我们建议您完成亚马逊 EC2 文档中从[迁移到最新一代实例类型的](#)步骤。

使用可突发实例

虽然突发实例是节省计算成本的好方法，但我们建议您在以下情况下避免使用这些实例：

- 具有桌面体验@@ [的 Windows 服务器的最低规格](#)要求有 2 GB 的内存。避免在 Windows 服务器上使用 t3.micro 或 t3.nano 实例，因为它们缺少最少的内存。
- 如果您的工作负载激增，但闲置时间不足以积累突发积分，则使用普通 EC2 实例比使用可突发实例更高效。我们建议[您监控您的 CPU 积分](#)以验证这一点。
- 在大多数情况下，我们建议您避免在 SQL Server 中使用突发性能实例。SQL Server 的许可基于分配给实例的 vCPU 数量。如果 SQL Server 一天中的大部分时间都处于闲置状态，则需要为未充分利用的 SQL 许可证付费。在这些情况下，我们建议您将多个 SQL Server 实例整合到一个更大的服务器上。

后续步骤

我们建议您采取以下后续步骤来优化 Amazon EC2 Windows 实例的成本：

- 使用最新一代的 EC2 实例以获得最佳性价比。
- 使用配备 AMD 处理器的 EC2 实例可将计算成本降低百分之十。
- 选择与您的工作负载相匹配的 EC2 实例类型，最大限度地提高资源利用率。

下表显示了 Windows 工作负载的典型起点示例。还有其他选项可供选择，例如用于增强 SQL Server 工作负载的实例存储卷或具有更大的 vCPU 与 RAM 比率的 EC2 实例。我们建议您对工作负载进行全面测试，并使用诸如此类的监控工具 AWS Compute Optimizer 来帮助进行必要的调整。

工作负载	典型的	可选
Active Directory	T3、m6i	R6i
文件服务器	T3、m6i	C6i
Web 服务器	T3、c6i	m6i、R6i
SQL Server	R6i	x2iedn、x2iezn

如果您必须更改 EC2 实例类型，则该过程通常只需要简单的服务器重启即可。有关更多信息，请参阅 Amazon EC2 文档中的[更改实例类型](#)。

在更改实例类型之前，我们建议您考虑以下几点：

- 您必须先停止由 Amazon EBS 支持的实例，然后才能更改其实例类型。请务必为实例停止时的停机时间做好计划。停止实例并更改其实例类型可能需要几分钟时间，重新启动实例所用的时间则由应用程序的启动脚本决定。有关更多信息，请参阅 Amazon EC2 文档中的[停止和启动您的实例](#)。
- 当您停止并启动实例时，AWS 会将该实例移至新硬件。如果您的实例有公有 IPv4 地址，则 AWS 释放该地址并为您的实例提供新的公有 IPv4 地址。如果您需要一个不变的公有 IPv4 地址，请使用[弹性 IP 地址](#)。
- 如果在实例上启用了[休眠](#)功能，则无法更改实例类型。
- 您无法更改[竞价型实例](#)的实例类型。
- 如果您的实例位于 Auto Scaling 组中，Amazon EC2 Auto Scaling 会将已停止的实例标记为运行状况不佳，并且可能会将其终止并启动替代实例。为防止出现此情况，您可以在更改实例类型时，为组暂停扩展流程。有关更多信息，请参阅 Amazon EC2 [Auto Scaling 文档中的暂停和恢复 Auto Scaling 组的流程](#)。
- 当您更改具有 NVMe 实例存储卷的实例类型时，更新的实例可能会有额外的实例存储卷，因为即使未在 Amazon 系统映像 (AMI) 或实例块储存设备映射中指定，所有 NVMe 实例存储卷都可用。否则，已更新的实例通常具有您在启动原始实例时指定的相同实例存储卷数。

其他资源

- [亚马逊 EC2 实例类型](#) (AWS 文档)
- [AWS 优化和许可评估](#) (AWS 文档)

为 Windows 和 SQL Server 工作负载提供许可证

概述

如果你在 Microsoft 工作负载和现有企业许可协议方面有大量投资，则可以从多个 AWS 选项中进行选择来支持这些工作负载，包括[随附许可证（由提供 AWS）](#)和[自带许可（BYOL）](#)选项。您可以使用[Amazon EC2 专用主机](#)来充分利用现有的微软许可协议，将 Windows Server 引入 AWS。这可以为您节省高达 50% 的 Amazon EC2 实例成本。由于 Windows 许可证约占实例成本的一半，因此将 Windows Server 置于 AWS 于专用主机上可以节省大量成本。由于无法将 Windows Server 置于[默认（共享）租期](#)，因此，如果您想在上 AWS 面使用现有的 Windows Server 许可证，专用主机是理想的选择。

专用主机不仅适用于 Windows 服务器 BYOL 实例。它们还为您提供了与现有 SQL Server 工作负载的本地许可相匹配的灵活性。专用主机公开底层服务器的物理内核，使您能够在物理核心级别上许可 SQL Server。在默认（共享）租赁中，这是不可能的，其中 SQL Server 许可基于分配给实例的虚拟 CPU 数量。此功能使您能够以符合本地许可策略 AWS 的方式对 SQL Server 工作负载进行许可。因此，与默认（共享）租赁相比，使用符合条件的 Windows 许可可以节省高达 50% 的 SQL Server 许可成本，此外还可以节省实例成本。有关此场景的更多信息，请参阅本指南的“[了解 SQL Server 许可](#)”部分。

Amazon EC2 专属主机

Amazon EC2 专用主机与用于运行其 EC2 计算产品的 EC2 主机基本相同。AWS 不同之处在于，这些主机完全专用于单个客户，并提供对底层物理基础架构的专有访问权限。您可以使用专用主机在完全专供您使用的硬件上运行您的实例，而不必与其他 AWS 客户共享资源。这使您可以更好地控制云资源，并通过将自己的软件许可证（例如 Windows Server 和 SQL Server）带到云端来降低成本 AWS。

记住以下内容：

- 专用主机是完全专用于单个客户的物理服务器。您可以查看专用主机的插槽和物理内核，从而满足许可合规性要求，例如每插槽、每内核或每虚拟机的软件许可协议。
- 可以支持同一实例系列的多个实例大小的专用主机被称为异构专用主机。这些[实例系列](#)包括 T3、A1、C5、M5、R5、c5n、r5n 和 m5n。相比之下，其他实例系列仅支持同一专用主机上的一个实例大小。它们被称为同构专用主机。
- 专用主机按每台主机计费。这意味着，无论在专用主机上运行多少实例，您都要按每台专用主机付费。专用主机的定价因所选实例系列、地区和付款选项而异。您可以为工作负载选择最佳配置，以实现所需的性能和成本结果。

此图说明了共享租赁实例和专用主机之间的区别。



同质专用主机

以使用 M6i 专用主机的场景为例。M6i 和 R6i 专用主机有两个插槽、64 个物理内核，并支持相同大小的实例类型。它们被称为同质专用主机。这意味着您可以在单个 M6i 专用主机上启动的实例数量取决于实例大小。

例如：

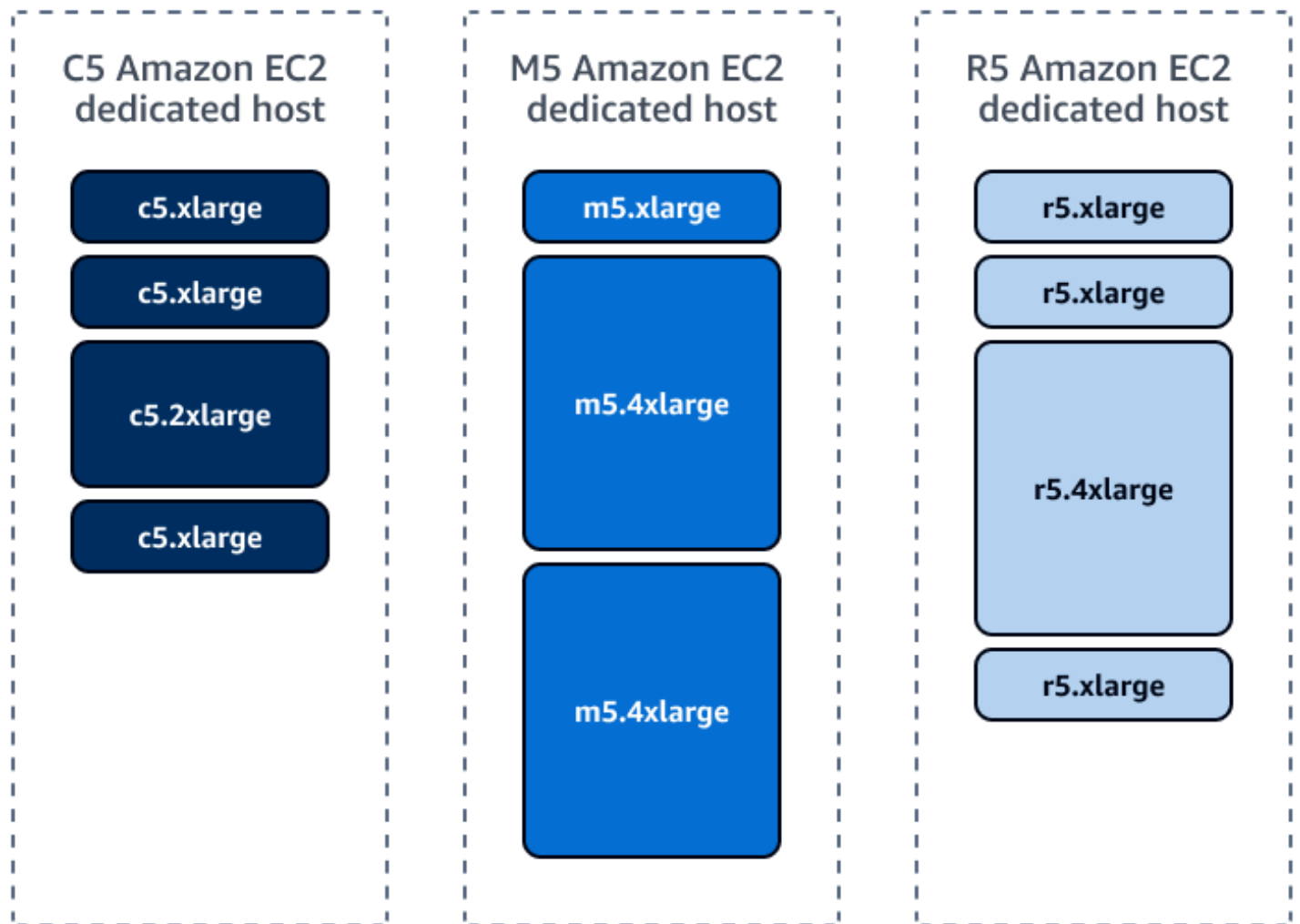
- 对于超大型（4 个 vCPU），您最多可以在此专用主机上启动 32 个 m6i.xlarge 实例。
- 对于 8xlarge（32 个 vCPU），您最多可以在此专用主机上启动 4 个 m6i.8xlarge 实例。
- 对于金属（128 个 vCPU），您最多可以在此专用主机上启动 1 个 m6i.metal 实例。

下图显示了 M6 实例的专用主机选项。



异构专用主机

在同一主机上支持多种实例大小的专用主机被称为异构 Amazon EC2 专用主机。下图显示了具有各种实例大小 (例如 2xlarge、xlarge 和 4xlarge) 的 C5、M5 和 R5 专用主机的示例。



专用主机管理

我们建议您在管理 Amazon EC2 专用主机时考虑以下几点：

- 要充分利用专用主机，您可以在[组织内的多个账户之间共享一台主机](#)。主机共享可实现资源优化，并可通过使用主机上的每个可用插槽来节省成本。通过在业务部门之间共享专用主机，您可以集中您的 IT 基础架构并提高资源利用率，同时仍能保持工作负载之间的分离。如果您是组织中的一员，AWS Organizations 并且您的组织内启用了共享，则会自动授予组织中的消费者访问共享专用主机的权限。否则，使用者会收到加入资源共享的邀请，并在接受邀请后获得对共享专用主机的访问权限。

- 您可以在包含许可证的模式下在专用主机上运行 Windows Server 2022，因为 Windows Server 2019 是您可以自带设备的最新版本。如果你想在专用主机上使用 Windows Server 2022，则必须使用包含 Windows Server 2022 许可证的实例。
- [AWS License Manager](#) 是一款全面的解决方案，用于管理来自不同供应商的跨本地环境 AWS 的软件许可证。如果您 [使用 License Manager](#)，则可以更好地了解和控制软件许可证的使用方式，从而节省成本并提高合规性。您可以使用 License Manager 来设置规则来模拟您的独特许可条款。这使您能够强制执行这些规则并防止许可证滥用。这可以降低不合规的风险并改善许可证管理流程。
- 您可以使用 License Manager 通过 [主机资源组](#) 自动放置、释放和恢复主机。这可以提高工作效率并减少管理开销。License Manager 还提供基于许可规则的跨 AWS 本地环境的许可证使用情况的集中视图，从而可以轻松管理整个组织的增量许可购买、合规性和供应商审计。此外，License Manager 还与 AWS Organizations 和 AWS Resource Access Manager (AWS RAM) 集成，可跨账户和地区共享许可证配置。这使您能够根据计划为整个环境创建报告，并在其中集中管理许可规则 AWS 账户。最终，这可以改善治理并降低复杂性。
- 在为单个区域内的专用主机设计高可用性时，请确保在至少两个可用区中为生产关键型工作负载分配了至少两台专用主机。有关更多信息，请参阅有关 AWS 参考部署的 [微软 Windows 版 Amazon EC2 专用主机](#)。
- 对于每个专用主机实例系列，对于每种实例大小，您可以运行的实例数量都有限制。有关更多信息，请参阅 Amazon EC2 文档中的 [专用主机配置表](#)。

AWS 许可选项

许可证主要分为以下几类：

- 包括许可证 — 此许可选项允许您按需购买和使用许可证，只需为使用的内容付费。它非常适合在许可使用方面寻求灵活性并希望避免前期成本的用例。你可以从各种 Windows Server、SQL Server 和其他微软产品中进行选择。
- 具有许可证移动性的 BYOL 产品 — 如果您已经拥有现有许可证并想在云中使用时，则此许可选项允许您通过 [Microsoft 许可证移动性计划将自己的许可证](#) 带到云端。具有许可证移动性的产品，例如具有软件保障 (SA) 的 SQL Server，可以采用共享或专用租约。这降低了 AWS 实例成本。
- 没有许可证移动性的 BYOL 产品 — 对于缺少许可证移动性的微软产品，如 Windows Server，AWS 提供了在云中使用这些产品的专用选项。此外，专用主机还支持物理核心级别的许可，从而可以在运行工作负载所需的许可证上节省 50% 或更多。对于大部分时间运行的稳定且可预测的工作负载，专用主机是绝佳的选择。

带上 Windows 服务器许可证

自带Windows许可证是最有效的许可证优化策略之一，因为它使您能够利用现有投资并减少 AWS 开支。特定的 BYOL 场景不需要 SA 或许可证移动性权益，但是 Amazon EC2 的专用基础设施始终是必需的。要获得资格，您必须在 2019 年 10 月 1 日之前购买永久许可证，或者在 2019 年 10 月 1 日之前生效的有效企业注册下将其添加为调整许可证。在这些特定的 BYOL 场景中，您只能将许可证升级到 2019 年 10 月 1 日之前可用的版本。例如，如果你在 2017 年放弃了 SA，那么您只能部署到 Windows Server 2016，而不是 2019 年。但是，2019 年是最后一个有资格使用 BYOL 的 AWS 版本。有关更多信息，请参阅 AWS 文档中的[许可-Windows 服务器](#)。

获得许可证可能会显著影响运行 Microsoft 工作负载的成本 AWS。当您自带许可证时，您无需为在云中运行的实例支付额外的许可费用，这样可以节省大量成本。

下表显示了在各种配置下全天候运行单个 c5.xlarge 实例的按需每月成本。

配置	每月费用 (美元)
Windows 服务器 + SQL Server 企业版	1,353.00 美元 (LI)
Windows 服务器 + SQL Server 标准版	609.00 美元 (LI)
仅限 Windows 服务器	259.00 美元 (LI)
仅限计算 (Linux)	127.00 美元

您可以使用现有的许可证来降低许可成本并节省总 AWS 账单。

要获得亚马逊 EC2 专用主机上的 BYOL 资格，您必须自带软件许可证，例如 Windows Server 和 SQL Server 的软件许可证。BYOL 允许您在上使用现有许可证 AWS，从而节省成本。要自带许可证，您必须获得软件供应商的许可授权，还必须提供软件的安装媒体或映像。安装媒体或映像可用于在专用主机上启动实例。要了解有关创建 BYOL AMI 的更多信息，请参阅博客上的 Microsoft 工作负载中的[如何使用虚拟机导入/导出从本地创建 Windows Server Bring-Your-Own-License AMI](#)。AWS

Note

设置为“自动”的许可证类型等同于[AWS 包含许可证](#)的选项。此选项可能会导致不必要的按需支出。您将需要切换[许可类型](#)。

成本优化场景

正确调整和优化许可证是成本优化的关键组成部分 AWS。如果您实施了正确的策略，则可以使用 Amazon EC2 专用主机和 BYOL 选项来降低许可成本、保持合规性并从许可投资中获得最大价值。

本节涵盖以下示例场景：

- 使用 T3 专用主机节省成本
- 将共享租赁与使用 SQL Server BYOL 的专用主机进行比较
- 高可用性的 SQL 服务器部署

使用 T3 专用主机节省成本

T3 专用主机与传统上提供固定 CPU 资源的其他 Amazon EC2 专用主机不同。相比之下，T3 专用主机支持可突发实例，这些实例能够共享 CPU 资源，提供基准 CPU 性能，并在需要进行爆发。共享 CPU 资源（也称为超额订阅）使单个 T3 专用主机能够支持比同类通用专用主机多四倍的实例。

与任何其他 Amazon EC2 专用主机相比，T3 专用主机提供的实例密度更高，从而降低了总拥有成本。Burstable T3 实例使您能够在比以往任何时候更少的主机上整合 low-to-moderate 更多具有平均 CPU 利用率的实例。与其他 Amazon EC2 专用主机相比，T3 专用主机还能在更多的 vCPU 和内存组合中提供更小的实例大小。较小的实例大小可以降低总拥有成本，并有助于提供等同于或大于本地主机的整合率。

T3 专用主机最适合运行具有 low-to-moderate CPU 利用率和符合条件的每插槽、每内核或每虚拟机软件许可证的 BYOL 软件，包括微软 Windows 桌面、Windows Server、SQL Server 和 Oracle 数据库。

使用 T3 专用主机减少 Windows 服务器数据中心许可证（每核）

在本地环境中，您可以利用这样一个事实，即您可以轻松地在 VMware 主机上超额订阅物理 CPU 并实现高级别整合。

考虑以下示例。你目前在本地环境中使用的是 10x36 内核、384 GB RAM 的 VMware 主机。此外，每台主机都运行 96x2 vCPU、4 GB RAM 的 Windows Server 虚拟机，平均 CPU 利用率较低。

现在，您可以通过将虚拟机迁移到 T3 专用主机来实现更高的整合级别，该主机的 RAM 量是当前本地 VMware 主机的两倍。您可以在 T3 专用主机上运行相同数量的服务器，同时将主机成本降低 50%。这可以帮助你降低 Windows 服务器的许可成本 33%。下表重点介绍了使用 T3 专用主机可以节省的费用。

	本地 VMware	T3 专用主机	节省成本
物理服务器	10	5	
每台主机的物理内核	36	48	
每台主机的内存 (GB)	384	768	
每台主机 2 个 vCPU、4 GB RAM 虚拟机	96	192	
虚拟机总数	960	960	
Windows Server 数据中心许可证总数 (每个内核) = (服务器数量 x 物理核心数量)	10 * 36 = 360	5 * 48 = 240	33%

将共享租赁与使用 SQL Server BYOL 的专用主机进行比较

举一个实际的例子来演示 Amazon EC2 专用主机的价值。在这种情况下，组织在具有 240 个内核的本地环境中运行 SQL Server 工作负载，并希望经济高效地在上 AWS 部署相同的工作负载。如果该组织自带许可证 (BYOL)，他们将继续为 SA 付费，减少内核数量会直接影响他们的成本。

下图比较了微软授权和 SQL Server 之间 AWS 节省的费用。

Microsoft entitlements (Enterprise Agreements)		SQL Server savings with AWS	
	Number of cores	AWS shared vCPUs	AWS BYOL/Dedicated Hosts cores
SQL Server Enterprise edition	208	120	96
SQL Server Standard edition	32	20	-
Total SA cost	\$341,000	\$197,418	\$151,355

通过在 AWS 共享租赁上调整实例的大小，您可以将 SQL Server 许可证减少到 140 个内核。这导致南澳的成本为 197,000 美元。

Amazon EC2 专用主机允许您在物理核心级别上许可 SQL Server。在 SQL Server 许可基于分配给实例的 vCPU 数量的共享租赁中，这是不可能的。因此，通过使用两台 R5 专用主机，每台 48 个内核，您只需要覆盖 96 个内核，而不是共享租赁所需的 140 个 vCPU。通过部署 R5 专用主机并在物理层面

许可工作负载，您可以将所需数量的 SQL Server Enterprise 版许可证减少到 96 个内核。这意味着您可以部署多达 192 个内核（考虑到超线程）的 SQL Server 工作负载，同时仍能满足许可要求并节省大量成本。

在这种情况下，该组织每年支付约 34.1 万美元的 SA 费用。在合理调整共享租赁规模后，他们使用 140 个 vCPU 将成本降低到 197,000 美元。Amazon EC2 专用主机将成本进一步降低至 15.1 万美元（减少了约 56%）。

高可用性的 SQL 服务器部署

此示例分析了成本如何影响 SQL Server 的部署，AWS 并考虑了各种许可注意事项。假设一个组织需要部署六台 SQL Server Enterprise 服务器 AWS 来支持三个应用程序。这些服务器需要高可用性，每台都有 16 个 vCPU 和 256 GB 内存。请参阅以下场景详细信息：

- 服务器 — SQL 服务器
- 操作系统版 — 2019 年 Windows 服务器数据中心
- SQL Server 版 — SQL Server E
- vCPU — 16
- 内存 (GB) — 256
- 数量 — 6

为了在 AWS 不牺牲性能的情况下优化成本，我们建议您根据 CPU、内存、网络 and 磁盘 (IOPS/BW) 利用率调整实例的大小。正确调整工作负载大小后，将其放在提供 16 个 vCPU 的 x2iedn.4xlarge 实例类型上。但是，此实例类型还包括工作负载所需的两倍内存。进一步的优化仍然是可能的。

方案 1

一个组织使用适用于 Windows 和 SQL Server 的“包含许可证”选项，在 AWS 共享租约上部署了六台 SQL Server Enterprise 服务器。使用此选项，Windows 和 SQL Server 许可证的成本将包含在实例价格中。请参阅以下场景详细信息：

- 共享租赁 (实例) — x2iedn.4xlarge
- 每小时成本 (美元) — 10.0705 美元
- 每月每单位成本 (美元) — 7,351.47 美元
- 服务器数量 — 6
- CPU — 16
- 内存 — 512

- 6 台服务器的每月成本 — 44,108 美元

方案 2

一个组织在共享租约上拥有 SQL Server 的 SA 和 BYOL。这意味着该组织使用适用于 Windows 的“包括许可证”选项，但会根据分配给实例的 vCPU 数量提供自己的 SQL Server 许可。由于该组织有六台 SQL Server Enterprise 服务器，每台服务器有 16 个 vCPU，因此总共需要 96 个 vCPU。请参阅以下场景详细信息：

- 共享租赁 (实例) — x2iedn.4xlarge
- 每小时成本 (美元) — 4.0705 美元
- 每月每单位成本 (美元) — 2971.47 美元
- 服务器数量 — 6
- CPU — 16
- 内存 — 512
- BYOL 内核 — 96
- 6 台服务器的月度成本 — 17,828 美元

通过将自己的 SQL Server 许可证与 SA 一起使用 SQL Server 许可证，与使用 SQL Server 附带许可证的选项相比，这种情况下的组织可以节省成本。确切的成本节省取决于具体许可协议的定价和条款。在这种情况下，将 SQL Server Enterprise 许可证引入后，每月 AWS 的成本将减少 26,280 美元。

AWS

方案三

一个组织在 Amazon EC2 专用主机上同时拥有 Windows 和 SQL Server 的 BYOL。这意味着组织将在物理核心级别分配许可证，使他们能够仅许可主机的物理内核。物理核心级别的许可允许您在不影响所需许可证的情况下部署最大数量的实例。这种许可模式通常用于 Windows 服务器数据中心和 SQL Server 企业版。

此场景使用两台 x2iezn Amazon EC2 专用主机。每台主机有 24 个物理内核和 48 个 vCPU。这为六台 SQL Server Enterprise 服务器提供了足够的容量，每台服务器有 16 个 vCPU 和 256 GB 内存。请参阅以下场景详细信息：

- 专用主机数量 — 2
- 实例系列 — x2iezn
- 每小时成本 (美元) — 11.009 美元

- 每月每单位成本 (美元) — 8,036 美元
- 物理核心 — 48
- 可用的 vCPU — 96
- 需要 Windows 服务器核心许可证 — 24
- SQL Server 企业版核心所需的许可证 — 24
- 每月费用 — 16,073

两台 x2iezn 系列亚马逊 EC2 专用主机的总费用为每月 16,073 美元。有关定价的更多信息，请参阅此场景的 AWS Pricing Calculator [估算值](#)。在这种情况下，组织通过携带 Windows 许可证，每月可以节省 1,755.65 美元。如果他们使用 Amazon EC2 专用主机，他们还可以减少所需的 SQL Server 许可证数量。在共享租赁中，他们需要 96 个 SQL Server Enterprise 许可证才能覆盖六台 SQL Server Enterprise 服务器，每台服务器有 16 个 vCPU。但是，通过使用 Amazon EC2 专用主机和物理核心级别的许可，他们可以将所需的许可证数量减少到 48 个内核。

以下详细信息比较了示例 3 中的成本，并显示了与其他方案相比，使用 BYOL 选项在 Amazon EC2 专用主机上部署工作负载可以节省多少成本。

- 本地服务器 — SQL Server
- vCPU — 16
- 内存 — 256
- 服务器数量 — 6
- 场景 1 的每月费用：Windows (LI) + SQL Server Enterprise (LI) — 44,108 美元
- 场景 2 的月度成本：Windows (LI) + SQL Server Enterprise (BYOL) — 17,828 美元
- 方案 3 的月度费用：亚马逊 EC2 专用主机上的 Windows (LI) + SQL Server Enterprise (BYOL) — 16,073 美元

Note

费用基于按需定价。您可以使用 Savings Plans 或专用预留实例进一步降低成本。与按需定价相比，这些选项提供了灵活的定价模式，可节省大量成本。有了这些计划，你可以承诺一到三年的任期。有关更多信息，请参阅本指南的[优化 Windows 在 Amazon EC2 上的支出](#)部分。

请考虑以下适用于 Amazon EC2 专用主机的付款方式：

- [专用主机](#) (亚马逊 EC2 文档)
- [专用主机预留](#) (Amazon EC2 文档)
- S@@ [avings](#) Plans (亚马逊 EC2 文档)

[AWS Pricing Calculator](#)现在支持专用主机定价。这可以帮助您选择合适的底层专用主机。

成本优化建议

我们建议您采取以下后续步骤来优化成本，方法是 AWS Cost Explorer：

1. [启用 Cost Explorer](#)。
2. 使用 Cost Explorer [查看和分析您的 Amazon EC2 专用主机部署的成本和使用情况](#)。
3. 验证您是否正在运行 BYOL。您可以在 Amazon EC2 控制台的实例或 AMI 页面上，或者在 describe-images 或 describe-instances 命令返回的响应中显示以下平台详情和使用操作值。
 - 平台详情：Windows，使用操作:: 0002 RunInstances (包括许可证)
 - 平台详情：Windows BYOL，使用操作:: 0800 RunInstances

其他资源

- [符合许可证类型转换条件的许可证类型](#) (AWS License Manager 文档)
- [AWS License Manager 和专用主持人研讨会](#) (AWS License Manager 研讨会)
- [Amazon EC2 专用主机常见问题解答](#) (AWS 文档)
- [如何使用虚拟机导入/导出从本地创建 Windows Server Bring-Your-Own-License AMI \(博客上的微软工作负载 \)](#) AWS
- [虚拟机导入/导出 \(文档 \)](#) AWS
- [亚马逊 Web Services 和微软：常见问题](#) (AWS 文档)
- [License Manager 中的许可证类型转换](#) (AWS License Manager 文档)
- 在 [Amazon EC2 专用主机上部署高度可用的 SQL Server](#) (AWS 云运营和迁移博客)

优化 Windows 在亚马逊 EC2 上的支出

概述

将服务器迁移到的首要问题之一 AWS 是基础架构成本。的确，云的好处之一是按需支付资源费用，但是有些生产工作负载需要全天候可用。Savings Plans 旨在为您在 EC2 实例的 AWS 稳定状态下节省资金 AWS Lambda、和。 AWS Fargate

Savings Plans提供灵活的定价模式，可以帮助您降低亚马逊EC2、Fargate、Lambda和Amazon SageMaker 使用量的定价，以换取对稳定的使用量（例如每小时10美元）的承诺。您承诺在一到三年内保持稳定的每小时计算支出，作为交换，您将获得该使用量的折扣。

通过 Savings Plans，您可以从三种不同的付款方式中进行选择：

- “无预付款”选项不需要任何预付款，并且您的承诺完全按月收取。
- 部分预付选项为Savings Plans提供更低的价格。您需要预先支付至少一半的承诺费用，其余部分按月收费。
- All Up front 选项提供最低的价格，您的全部承诺均通过一次付款收取。

您可以在中追踪您的储蓄计划到期日和即将排队的储蓄计划。AWS Cost Explorer您可以使用 Savings Plans 提醒在计划到期日前 1、7、30 或 60 天或在套餐排队等候购买时收到电子邮件提醒。这些通知还会提醒您注意到期日期。您最多可以向 10 个电子邮件收件人发送通知。

了解节省计划

每种类型的计算使用都有按需费率和 Savings Plans 费率。如果您承诺每小时计算使用10美元，则按储蓄计划费率计算，所有使用量最高可获得10美元的Savings Plans价格。任何超出计算支出承诺的使用量均按常规按需费率收费。你可以使用中的 Cost Explorer 开始使用储蓄计划 AWS Management Console。

通过使用Cost Explorer 中提供的建议，您可以轻松地对Savings Plans做出承诺，从而实现最大的节省。建议的每小时承诺使用量取决于您的历史按需使用量以及您选择的计划类型、期限和付款选项。Savings Plans 首先应用于购买该计划的账户，然后将其共享给整合账单系列中的其他账户。

Note

中的 Savings Plans 共享选项默认 AWS Organizations 处于启用状态。您可以在付款人账户的 AWS Billing 控制台中拒绝此选项。您可以访问 [“推荐”](#) 页面，查看推荐的 Savings P AWS

Plans，以帮助您节省符合条件的使用量。这些建议可以随时更新，以便您轻松购买最佳的 Savings Plans。

计算类节省计划

Compute Savings Plans 提供了最大的灵活性，有助于降低成本。无论实例系列、大小、可用区、区域、操作系统或租约如何，这些计划都会自动适用于 EC2 实例的使用情况。它们也适用于 Fargate 或 Lambda 的使用。例如，使用 Compute Savings Plans，您可以随时从 C4 实例更改为 M5 实例，将工作负载从欧洲（爱尔兰）转移到欧洲（伦敦），或者随时将工作负载从 EC2 转移到 Fargate 或 Lambda。您将自动继续支付 Savings Plans 的价格。

EC2 实例节省计划

EC2 Instance Savings Plans 提供最高的折扣，以换取承诺在某个地区使用单个实例系列（例如，承诺在弗吉尼亚北部保持一致的 M5 使用水平）。无论可用区、大小、操作系统或租约如何，这都会自动为您提供该区域所选实例系列的按需价格折扣。EC2 Instance Savings Plans 允许您在该地区某个系列中的实例之间更改使用量。例如，你可以从运行 Windows 的 c5.xlarge 转移到运行 Linux 的 c5.2xlarge，然后自动从 Savings Plans 的价格中受益。

计算和 EC2 实例储蓄计划都适用于属于亚马逊 EMR、亚马逊 Elastic Kubernetes Service (Amazon EKS) 和亚马逊弹性容器服务 (Amazon ECS) Elastic Container Service (Amazon ECS) 集群的 EC2 实例。Savings Plans 不涵盖亚马逊 EMR、Amazon EKS 和 Amazon ECS 的费用，但底层 EC2 实例在内。EC2 实例节省计划在计算类节省计划之前应用，因为计算类节省计划适用范围更广。

Note

做出承诺后，您无法轻易更改 Savings Plan。我们建议您在承诺使用任一 Savings Plans 选项之前仔细规划。与按需定价相比，Savings Plans 提供的价格更低，以换取承诺，并且在期限内无法取消。

按小时承诺示例

如果您购买了 Savings Plan，则需要在计划期限内按小时支付款项。如果您承诺每小时 10 美元的计算使用量，Savings Plan 定价将自动应用于所有使用量，每小时不超过 10 美元。任何超出承诺的使用量均按常规按需费率收费。您可以使用 Cost Explorer 中的 Savings Plans 购买推荐工具来获取推荐的承诺，从而最大限度地节省开支。在计划期限内，无法修改特定计划的每小时财务承诺。如果您想在分析使用量后增加使用量，则可以额外购买 Savings Plan 来支付超额使用量。

Savings Plans 的好处

与预留实例相比，Savings Plans 提供了更灵活的定价模式，可以为您节省资金，同时您可以充分利用 Savings Plans 提供的更广泛的计算选项。即使您的计算需求发生变化，Savings Plans 也提供折扣。这可以帮助您跟上不断变化的动态环境，而不会产生任何额外的管理开销。以下是使用 Savings Plans 的其他一些好处：

- 易于使用 — 获得自动折扣以换取金钱承诺。
- 灵活性-适用于多种使用类型的单一承诺。
- 潜在的节省 — 有多种储蓄方式。考虑以下示例：
 - 使用 Compute Savings Plans 在 Windows 服务器工作负载上节省 60% ([d2.8xlarge](#) , 3 年 , 全部预付 , Windows , 共享租赁 , us-east-2)
 - 使用 EC2 Instance Savings Plans 在 Windows 服务器工作负载上节省 73% ([d2.8xlarge](#) , 3 年 , 全部预付 , Windows , 共享租赁 , us-east-2)
 - 非特殊实例类型 ([t3 系列](#) , 3 年 , 全部预付 , Windows , 共享租赁 , us-east-2) 可节省 28-41%
 - Windows 服务器平均可节省 25-40%

Note

由于灵活性降低，EC2 Instance Savings Plans 提供的折扣比 Compute Savings Plans 更大。您承诺以折扣价使用。

每种类型的计算使用都有 Savings Plan 费率和按需费率。下表显示了每种操作系统类型的 Savings Plans 和按需费率。您需要根据承诺使用量按照 Savings Plans 费率收费，超出承诺使用量的任何使用量均按常规按需费率收费。

实例名称	Savings Plans 费率	按需储蓄	按需费率	操作系统	区域	付款选项	学期长度
x2iedn.xlarge	0.32	61%	0.83 美元	Linux	美国东部 (弗吉尼亚州北部)	无预付款	3

实例名称	Savings Plans 费率	按需储蓄	按需费率	操作系统	区域	付款选项	学期长度
x2iedn.xlarge	2.01	50%	1.02	Windows	美国东部 (弗吉尼亚州北部)	无预付款	3
x2iedn.xlarge	1.02	20%	2.52 美元	包括 Windows 许可证 + SQL Server 企业版	美国东部 (弗吉尼亚州北部)	无预付款	3
x2iedn.xlarge	0.32	61%	0.83 美元	BYOL	美国东部 (弗吉尼亚州北部)	无预付款	3

Savings Plans 包括操作系统，他们对 BYOL 有单独的折扣。它们都在 [Compute Savings Plans 计算器中进行了细分](#)。

预留实例定价模型

AWS 还有另一种基于承诺的定价模式，称为预留实例。如果您在承诺后计算发生了变化，导致预留实例无法使用，则此模型可能会出现问题。Savings Plans 旨在提供与[标准和可转换预留实例](#)类似的成本降低，但灵活性要大得多。无论实例系列、大小、操作系统、租赁或区域如何，Compute Savings Plans 都提供更低的 EC2 实例使用价格。它们还可实现最大的灵活性。

下表可以帮助您在 Savings Plans 或预留实例之间进行选择。

	Reserved Instance	EC2 实例节省计划	计算类节省计划
1 年平均折扣	高达 38%	高达 29%	高达 29%
3 年平均折扣	高达 58%	高达 73%	高达 60%

	Reserved Instance	EC2 实例节省计划	计算类节省计划
实例系列	Fixed	Fixed	灵活
实例大小	已修复 (不是 Linux)	灵活	灵活
地理位置	1 个区域	1 个区域	灵活
操作系统	Fixed	灵活	灵活
服务	亚马逊 EC2 或亚马逊 RDS	Amazon EC2	亚马逊 EC2、Fargate、Lambda
付款选项	全部、部分、无预付款	全部、部分、无预付款	全部、部分、无预付款
实例限制	每个可用区 20	无限制	无限制

Note

Savings Plans 的工作原理是根据每小时的货币承诺为您提供折扣。在计划有效期内，不能取消或更改每小时财务承诺，但您可以购买额外的 Savings Plans 来支付额外的使用量。这使您能够随着车队的增长保持稳定的每小时承诺。

您可以使用诸如 [AWS Cost Explorer](#) 或 [AWS Cloud 情报仪表盘](#) 之类的工具来跟踪您的承诺。Cost Explorer 提供了保险目标线，可以帮助您的组织规划其 Savings Plans 的保险策略。如果您的 75% 的工作负载处于稳定状态，那么 75% 是一个不错的目标。这样，25% 的支出是根据动态工作负载按需/可变的。如果您需要将其提高到 85% 的承保范围，则可以购买另一份 Savings Plans 承诺以增加每小时的资金承诺。

Note

我们建议您购买 Savings Plans 而不是预留实例，但如果您已经购买了预留实例，则两种承诺模式可以一起使用。

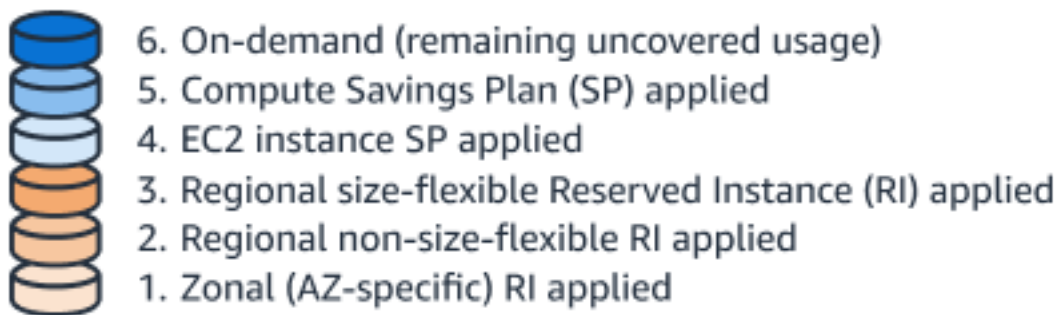
举一个例子，你购买了预留实例，但又想开始试用 Savings Plans 选项。这种组合适用于您的最终账单是合乎逻辑的。以下是你可以应用于你的层次结构 AWS 账户：

1. 区域预留实例适用于拥有该实例的账户。如果预留实例还有几个小时的剩余时间，则适用于该组织的其余部分。
2. 大小不灵活的 Windows 区域预留实例适用于拥有该实例的账户的匹配使用量。任何剩下的东西都会向组织的其他成员推出。
3. 大小灵活的区域预留实例适用于拥有该实例的账户（首先是家族中最小的实例，然后是更大的实例），然后适用于组织的其余部分。
4. 区域预留实例适用于任何未使用的按需容量预留。
5. EC2 Instance Savings Plans 适用于购买该计划的账户。
6. Compute Savings Plans 适用于购买该计划的账户。

Note

折扣从产生最高折扣的使用量开始，然后降至最低折扣。对于大多数常见的实例类型（例如 T3、M6 和 C5），Windows 实例的折扣潜力通常低于 Linux。这意味着，在大多数情况下，Linux 实例比 Windows 实例受益更多。

下图显示了将预留实例与 Savings Plans 分开后的价格。计算和 EC2 Instance Savings Plans 都首先适用于正在运行的实例，然后适用于未使用的按需容量预留。



成本优化场景

本节介绍使用含许可证计费模式的 Amazon EC2 专用主机和 Amazon EC2 实例的成本优化方案。

Amazon EC2 专属主机

考虑一下你要将本地 Windows 工作负载迁移到的场景 AWS。您的数据中心有以下服务器：

- 两台服务器拥有 16 个 vCPU 和 128 GB 内存
- 两台服务器拥有 32 个 vCPU 和 164 GB 内存
- 一台拥有 8 个 vCPU 和 64 GB 内存的服务器
- 16 台服务器配备 vCPU 和 32 GB 内存

此外，假设你可以自带许可证，AWS 因为你有足够的许可证可以带过来。下表显示了您可以在中使用的服务器实例 AWS。

实例类型	CPU	RAM	使用额
r5.4xlarge	16	128	2
r5.8xlarge	32	256	2
r5.2xlarge	8	64	1
r5.xlarge	4	32	16
			21

分析显示，这 21 台虚拟机可以分布在两台带有 R5 实例系列主机的专用主机上。下表显示了这两台专用主机的费用。

专用主机按需场景	预付工资	1 个月	1 年	3 年	AWS Pricing Calculator
按需	无	10,123 美元	121,475 美元	364,392 美元	AWS Pricing Calculator 估计
1 年期 Savings Plan	无	7,447 美元	89,362 美元	–	AWS Pricing Calculator 估计

专用主机按需场景	预付工资	1 个月	1 年	3 年	AWS Pricing Calculator
3 年 Savings Plan	无	5,476 美元	65,712 美元	197,128 美元	AWS Pricing Calculator 估计
3 年 Savings Plan, 含预付款	84,438 美元	2,755 美元	117,499 美元	183,618 美元	AWS Pricing Calculator 估计

如果您有想要迁移到 AWS 的服务器，则为期一年的 Savings Plan 的最终价格为 89,362 美元，而不是按需价格计算的 121,475 美元。这意味着一年后可享受 26.5% 的折扣。如果您正在考虑延长居留时间，则可以选择 AWS 为期 3 年的 Savings Plan 来节省更多成本。三年后，你支付的是 197,128 美元，而不是 364,392 美元。这使得三年后可节省总金额的 46%。

包含许可证的 Amazon EC2 实例

假设您要将单个三层应用程序迁移到 AWS，并希望使用提供的 AWS 许可证。此外，假设您的应用程序适用于以下服务器：

- 两台带有两个 vCPU 和 4 GB 内存的网络服务器
- 两台具有八个 vCPU 和 16 GB 内存的应用服务器
- 两台具有 16 个 vCPU 和 64 GB 内存的数据库服务器（使用 SQL Server 标准版）

下表显示了您可以在中使用的服务器实例 AWS。

实例类型	CPU	RAM	使用额
c5.large	2	4	2
c5.2xlarge	8	16	2
r5.2xlarge	8	64	2
			6 台服务器

下表显示了这些服务器在中的成本 AWS。

包括的许可证 AWS	预付工资	1 个月	1 年	3 年	AWS Pricing Calculator
按需	无	3,912 美元	46,950 美元	140,849 美元	AWS Pricing Calculator 估计
1 年期 Savings Plan	无	3,466 美元	41,952 美元		AWS Pricing Calculator 估计
3 年 Savings Plan，无需预 付款	无	3,189 美元	38,264 美元	114,804 美元	AWS Pricing Calculator 估计
3 年 Savings Plan，含预付 款	112,110 美元	无	无	无	AWS Pricing Calculator 估计

如果您想以按需定价在生产环境（全天候）中运行这些服务器，则每月需要支付 3,912 美元的费用。一年后每月支付的费用相当于 46,950 美元，三年后总共支付 140,849 美元。

如果您选择不预付款的 1 年期 Savings Plan，则每月费用将降至 3,466 美元。在第一年年底，您需要支付 41,952 美元。这相当于 11% 的总折扣。如果您选择不预付款的 3 年 Savings Plan，则每月费用将降至 3,189 美元。三年后，您需要支付 114,804 美元。这为您节省了 18.5%。

成本优化建议

当你计划和预测工作负载时，这两种情况都可以帮助你节省资金 AWS。重要的是要认识到，与第一种情况相比，第二种情况下的 discount 要小。在第二种情况下，许可价格包含在云服务器的价格中。AWS 不提供许可价格折扣，但您可以随时携带许可证（在特定情况下），并且 AWS 可以随时保证最优惠的计算/实例价格。

我们建议您采取以下措施来控制计算和实例资源 AWS 支出：

- 访问推荐

- 根据您的需求定制推荐
- 查看每小时承诺

访问推荐

您可以使用 [Amazon EC2 控制台](#) 访问有关您的 Savings Plan 的建议。您甚至可以以 CSV 格式下载您的推荐以供日后查看。有关更多信息，请参阅 Savings Plans 文档中的 [监控您的储蓄计划](#)。

根据您的需求自定义推荐

打开 [Amazon EC2 控制台](#)，展开“实例”部分，然后选择 Savings Plans。此页面显示了提出建议之前和之后的实例和计算价格。您也可以根据您的建议调整以下因素：

- 期限-例如，1-3 年
- 付款选项 — 例如“预付”、“部分预付”或“不预付”
- 历史记录-例如，过去 7、30 或 60 天

查看每小时承诺

使用同样的示例，假设您有一个全天候运行的实例。建议使用 Savings Plan。根据规模，您的按需价格为每小时 120 美元。您可以选择承诺每小时 90 美元，但这可能会因地区、实例和购买选项而异。在此示例中，与按需成本相比，您可以节省 25%。您还可以跟踪您的利用率和覆盖范围（如果它们低于您定义的阈值），并在预算即将结束时配置提醒。

查看建议

我们建议您仔细查看 Savings Plan 建议。AWS 未经您的许可，不会更改任何内容。这些只是建议，是否应用由您自己决定。

购买套餐

打开 [Amazon EC2 控制台](#)，展开“实例”部分，然后选择 Savings Plans。然后，选择 Purchase Savings Plans。根据您的要求，您可以选择以下选项：期限、区域、实例系列、按小时承诺使用、付款选项，甚至是开始日期。您可以从 Compute Savings 计划、EC2 实例储蓄计划和 SageMaker 储蓄计划中进行选择。有关更多信息，请参阅 [Savings Plans 文档中的购买储蓄计划](#)。

获取利用率报告

购买 Savings Plan 后，您可以获得使用率报告。该报告可帮助您检查使用率，查看购买的计划是否足以覆盖和最大限度地提高折扣，以及取消或添加新的折扣。此报告可以导出为其他格式，例如 CSV。有关更多信息，请参阅 [Savings Plans 文档中的使用利用率报告](#)。

遵循最佳购买实践

我们建议您在购买 Savings Plans 之前遵循以下最佳实践：

- 用于[AWS Trusted Advisor](#)移除闲置的 EC2 资源。
- 在购买 Savings Plans 之前，请进行任何合适的调整。
- 确定持续保持 30-60 天的小时费率。
- 购买一份承诺书，以支付您的组织所满意的尽可能多的固定小时费率。考虑需求或季节的波动。
- 选择季度审查 Savings Plans 预算以保持一致的费率（例如，Savings Plans 承保范围的覆盖率目标为 70%）。如果费率降至所需承保范围以下，请额外购买 Savings Plan 作为补偿，以实现您的保险目标。

其他资源

- [Amazon EC2 预留实例的储蓄计划](#)（AWS 白皮书）
- [了解 Savings Plans 如何适用于您的 AWS 用量](#)（Savings Plans 文档）
- [宣布 EC2 Windows 服务器和 SQL Server 实例的每秒计费](#)（AWS 文档）
- [AWS 成本优化系列：Savings Plans 视频 | Amazon Web Services](#) () YouTube

使用 AWS 工具监控成本

概述

成本可见性是优化成本的关键因素 AWS。AWS 有许多工具可用于可视化成本并针对这些成本创建警报。其中包括可帮助您跟踪和报告支出的工具。AWS Budgets 本节介绍监控 Windows AWS 支出的具体方法，以便您可以跟踪预算要求并相应地做出反应。这包括向您的 Windows EC2 资源添加必要的标签。这些标签使您能够使用正确监控 Windows EC2 和其他微软服务 AWS Budgets。

通过监控支出并使用 AWS 工具创建警报，您可以更详细地了解当前支出、预计支出和支出异常情况。如果您使用 [Savings Plans](#) 来帮助降低 EC2 实例的每小时定价，我们建议您查看储蓄计划的总体使

用率和覆盖范围。这可以帮助您确保持续实现节约。您可以使用 AWS Cost Explorer 查看储蓄计划库存，并根据以前的使用情况获得有关其他储蓄计划的建议。您还可以通过使用[AWS Budgets](#)和设置来跟踪特定支出[AWS Cost Anomaly Detection](#)。

成本优化建议

我们建议您采取以下后续步骤，通过使用 AWS Budgets Cost Explorer 和异常检测来优化成本：

- 标记 Windows EC2 资源
- 使用设置警报 AWS Budgets
- 启用成本异常检测
- 获取实时支出分析
- 使用 Cost Explorer 查看包含许可证的 Windows 支出

标记 Windows EC2 资源

要有效监控 AWS 支出，必须为要监控的工作负载[制定标签策略](#)。这很重要，这样您就可以对资源进行分类分组，并获得有关特定支出的通知，而不是一般使用支出。您可以使用标签资源，这些资源不仅可以降低成本，还可以用于其他目的，例如[AWS Systems Manager 自动化](#)。此外，我们建议您对[必需的标签](#)进行一些管理。

要跟踪您在 AWS Budgets“Cost Explorer”和“成本异常检测”中的支出，必须确保设置了适当的标签。您可以使用标签为与这些标签匹配的项目设置特定的预算，以便在支出增加时提醒您。

例如，您可以使用一个简单的标签，比如 key=OS Value=Windows。这会将你的所有 Windows 实例整合到一个组中，你可以跟踪其支出。您也可以为其他项目使用标签，例如 Systems Manager。创建标签后，必须激活标签以进行成本跟踪。考虑添加[一条AWS Config 规则来监控附加到某些资源的标签](#)。AWS Config 如果正在运行的资源不包含相应标签，则可以提醒您，这些标签可以准确地显示您的 Windows EC2 支出。

设置好标签后，您可以在中创建自定义预算 AWS Billing。这样可以查看你的 Windows EC2 支出。您可以设置每日预算或每月预算。

使用设置警报 AWS Budgets

在此示例场景中，您将为 Windows EC2 创建每日预算。这是一项经常性预算，它使用自动调整选项来跟踪您的支出并相应地调整预算。如果您的环境是静态的，则可以改用固定预算。请务必选择基准时间范围（例如，30 天）。

1. 登录 AWS Management Console 并打开 [AWS Cost Management 控制台](#)。
2. 在导航窗格中，选择预算。
3. 在页面顶部，选择创建预算。
4. 在预算设置下，选择自定义（高级）。
5. 在预算类型下，选择成本预算。然后选择下一步。
6. 在详细信息下，在预算名称中输入您的预算名称。例如，Windows EC2 支出。
7. 在“设置预算金额”下，为“期间”选择“每日”。
8. 对于预算续订类型，请为在预算期之后重置的预算选择经常性预算。
9. 在“开始日期”中，选择开始跟踪预算金额的起始日期或期间。
10. 对于预算方法，请选择自动调整（新建）。
11. 对于基准时间范围，选择自定义范围，然后输入 30 天。
12. 选择下一步。
13. 在预算范围部分，选择筛选特定 AWS 成本维度。这是使用标签来创建适当尺寸的地方。AWS Budgets 在其筛选器中不支持“平台类型”作为选项。因此，您必须应用操作系统标签。
14. 选择“添加过滤器”，然后从“维度”中选择“标签”选项。
15. 选择操作系统标签，然后为此选择 Windows 值，为标签创建预算。
16. 选择下一步。
17. 在配置警报页面上，选择添加警报阈值。在这里，您可以设置两个警报：一个用于 50% 的阈值，另一个用于 100% 的阈值。如果在该月的中点之前突破了 50% 的阈值警报，它将发出警告。这样，您就可以检查支出是否超出预期，并在月底之前做出反应。
18. 在阈值中，输入 50，然后选择预算金额的百分比。
19. 对于触发器，选择实际。
20. 对于电子邮件收件人，请输入电子邮件地址。添加另一个阈值为 100 的警报。

Note

此示例使用简单的电子邮件通知来发送警报，但您也可以使用 [Amazon Chime](#) 或 Slack。

启用成本异常检测

您可以使用成本标签来设置异常的支出提醒。例如，您可以使用 [AWS Cost Anomaly Detection](#) 创建支出监控器，并在系统检测到您的账户中存在异常支出时收到提醒。

要为之前创建的 key=OS 和 Value=Windows 标签设置监视器和警报，请执行以下操作：

1. 登录 AWS Management Console 并打开 [AWS Cost Management 控制台](#)。
2. 在导航窗格中，选择成本异常检测。
3. 选择“成本监控器”选项卡，然后选择“创建监控器”。
4. 在步骤 1 中，选择成本分配标签作为您的显示器类型。
5. 对于成本分配标签密钥，请选择 Windows EC2 支出。
6. 对于成本分配标签值，请选择 Windows。
7. 在“为显示器命名”中，输入 Windows EC2 支出。
8. 选择下一步。
9. 要为警报创建订阅，请选择创建新订阅。如果您有现有的订阅，请选择 Choose an existing subscription (选择现有的订阅)。
10. 在订阅名称中，输入 Windows EC2 支出异常。
11. 对于提醒频率，请选择每日摘要。
12. 对于提醒收件人，请输入您的电子邮件地址。
13. 选择添加阈值。在“阈值”中，输入 10，然后选择高于预期速度的百分比。
14. 选择 Create monitor (创建监控)。

实时查看支出

警报是监控你的 Windows EC2 支出的有用工具，但如果你想实时查看支出，则必须使用 Cost Explorer。观看此视频，了解 Cost Explorer 如何使您能够分析和降低 EC2 成本。有关更多信息，请观看上的 [“AWS 支持您 | 了解并降低您的 EC2 成本”](#) 视频 YouTube。

查看 Windows 中包含许可证的支出

您可以使用 Cost Explorer 查看账户中的 EC2 Windows 支出。要查看 Windows 包含许可证的支出，您必须在 Cost Explorer 中设置以下正确的 [筛选条件](#)：

- 对于平台，请选择 Windows (亚马逊 VPC)。对于 API 操作，请选择 RunInstance:0002。这是包含许可证的 Windows EC2 实例的 AWS Billing 代码。
- 如果您想查看您的 BYOL 实例支出，请将:0002 更改为:0800 RunInstance。RunInstance 这是 Windows EC2 BYOL 的账单代码。

借助 Cost Explorer 中的这种可见性，你可以快速筛选出你在 Windows EC2 上花费的成本。如果您想更深入地了解自己的 AWS 支出，可以使用向下筛选 AWS 成本和使用情况报告 到单个实例级别的支出。您还可以生成可在 Amazon 中可视化的报告，QuickSight 并构建自定义控制面板。

欲了解更多信息，请观看上的 [“AWS 支持您-可视化您的成本和使用情况报告” YouTube 视频](#)。

其他 资源

- [使用 AWS Config \(AWS Config 文档 \) 设置必需的标签](#)
- [AWS Budgets 教程-为 AWS Billing | Amazon Web Services \(YouTube\) 设置提醒](#)
- [AWS 成本和使用情况报告 查询库 \(Well-Archit AWS ected Labs \)](#)

SQL Server

客户运行 Microsoft 工作 AWS 负载的时间已超过 15 年，比任何其他云提供商都要长。这主要是因为云端使用微软应用程序方面 AWS 拥有最丰富的经验，并且在以下领域为 Windows Server 和 Microsoft SQL Server 提供了最佳平台：

- 更高的性能和可靠性
- 更好的安全和身份服务
- 更多迁移支持
- 最广泛、最深层次的能力
- 降低总拥有成本 (TCO)
- 灵活的许可选项

AWS 支持构建和运行依赖于 SQL Server 的 Windows 应用程序所需的一切，包括 Active Directory、.NET、SQL Server、Windows 桌面即服务以及所有支持的 Windows 服务器版本。凭借久经考验的专业知识，AWS 可以帮助您轻松地提升、转移、重构甚至现代化 Windows 工作负载。

本指南的这一部分涵盖以下主题：

- [选择高可用性和灾难恢复解决方案](#)
- [了解 SQL 服务器许可](#)
- [为 SQL Server 工作负载选择正确的 EC2 实例](#)
- [整合实例](#)
- [比较 SQL 服务器版本](#)
- [评估 SQL Server 开发者版](#)
- [在 Linux 上评估 SQL 服务器](#)
- [优化 SQL 服务器备份策略](#)
- [实现 SQL 服务器数据库的现代化](#)
- [优化 SQL 服务器的存储](#)
- [使用 Compute Optimizer 优化 SQL Server 许可](#)
- [使用 Compute Optimizer 优化 SQL Server 的大小](#)
- [查看 SQL Server 工作负载 Trusted Advisor 的建议](#)

选择高可用性和灾难恢复解决方案

概述

我们建议您为 SQL Server 部署设计一种架构 AWS，既要满足您的业务需求，又能满足[灾难恢复 \(DR\) 目标](#)，包括恢复时间目标 (RTO) 和恢复点目标 (RPO)。以下解决方案可以帮助您为亚马逊弹性计算云 (Amazon EC2) 上的 SQL Server 设计正确的架构，同时还可以优化 SQL Server 工作负载的成本。

- SQL Server Always On 可用性组 — SQL Server Always On 可用性组为 SQL Server 数据库提供高可用性和灾难恢复 (HA/DR) 解决方案。可用性组由一组一起故障转移的用户数据库组成。Always On 可用性组还提供数据库级别的冗余，但不需要共享存储，每个副本都有自己的本地存储。您可以将此功能部署为 HA/DR 解决方案。有关更多信息，请参阅[什么是 Always On 可用性组？](#)在微软文档中。
- SQL Server Always On 故障转移群集实例 (FCI) — SQL Server Always On FCI 使用 Windows 服务器故障转移群集 (WSFC) 在 SQL Server 实例级别提供 HA。FCI 需要共享存储来托管数据库。您可以使用共享块存储或共享文件存储。例如，您可以使用适用于 Windows File Server 的 Amazon FSx 或适用 NetApp 于 ONTAP 的 Amazon FSx 作为具有多个可用区的共享存储解决方案。有关更多信息，请参阅 Microsoft 文档中的[始终开启故障转移群集实例 \(SQL Server\)](#)。
- SIOS DataKeeper — SIOS DataKeeper 可以通过启用跨可用区和可用区的 SQL Server FCI 来帮助同时满足高可用性和灾难恢复要求。AWS 区域 SIOS 使用本地 Amazon Elastic Block Store (Amazon EBS) 卷 DataKeeper 创建集群虚拟 SAN，并在可用区之间使用同步复制实现 HA，同时使用区域间的异步复制和灾难恢复。有关更多信息，请参阅 SIOS 文档中的[Windows 应用程序高可用性保护](#)。
- 分布式可用性组- 分布式可用性组是一种特殊类型的可用性组，跨越两个独立的 Always On 可用性组。可用性组可以位于两个不同的区域（例如 us-east-1 和 us-west-1）。您可以将分布式可用性组视为可用性组的可用性组，因为底层 Always On 可用性组是在两个不同的 WSFC 集群上配置的。部署分布式可用性组需要 SQL Server 企业版。有关更多信息，请参阅 Microsoft 文档中的[分布式可用性组](#)。
- 日志传送- 您可以实施日志传送来保护多个区域的数据库，以防某个区域受到影响并变得不可用。根据事务和日志传送频率，您可以在几分钟内实现 RPO 和 RTO。有关更多信息，请参阅 Microsoft 文档中的[关于日志传送 \(SQL Server\)](#)。
- AWS Elastic Disaster Recovery— Elastic 灾难恢复是一种软件即服务 (SaaS) 应用程序，用于管理服务器从任何基础设施复制到 AWS 灾难恢复。您还可以使用弹性灾难恢复跨区域复制 SQL Server。Elastic 灾难恢复是一种基于代理的解决方案，可将整个虚拟机（包括操作系统、所有已安装的应用程序和所有数据库）复制到暂存区。有关更多信息，请参阅[什么是 Elastic 灾难恢复？](#)在 Elastic 灾难恢复文档中。

- AWS Database Migration Service (AWS DMS) — AWS DMS 支持数据进出实时迁移 AWS，包括不同区域。您可以使用此功能在不同的区域中设置单独的 SQL Server 实例，以用作灾难恢复数据库。有关更多信息，请参阅[什么是 AWS Database Migration Service？](#) 在 AWS DMS 文档中。

SQL Server 始终开启可用性组

如果您将 SQL Server Enterprise 版仅用于高可用性 [Always On 可用性组](#)，则可以利用基本可用性组降级到 SQL Server 标准版。使用基本可用性组而不是 Always On 可用性组，可以将成本从 65-75% 降低。

Note

有关不同 SQL Server 版本之间成本差异的更多信息，请参阅本指南的“[比较 SQL Server 版本](#)”部分。

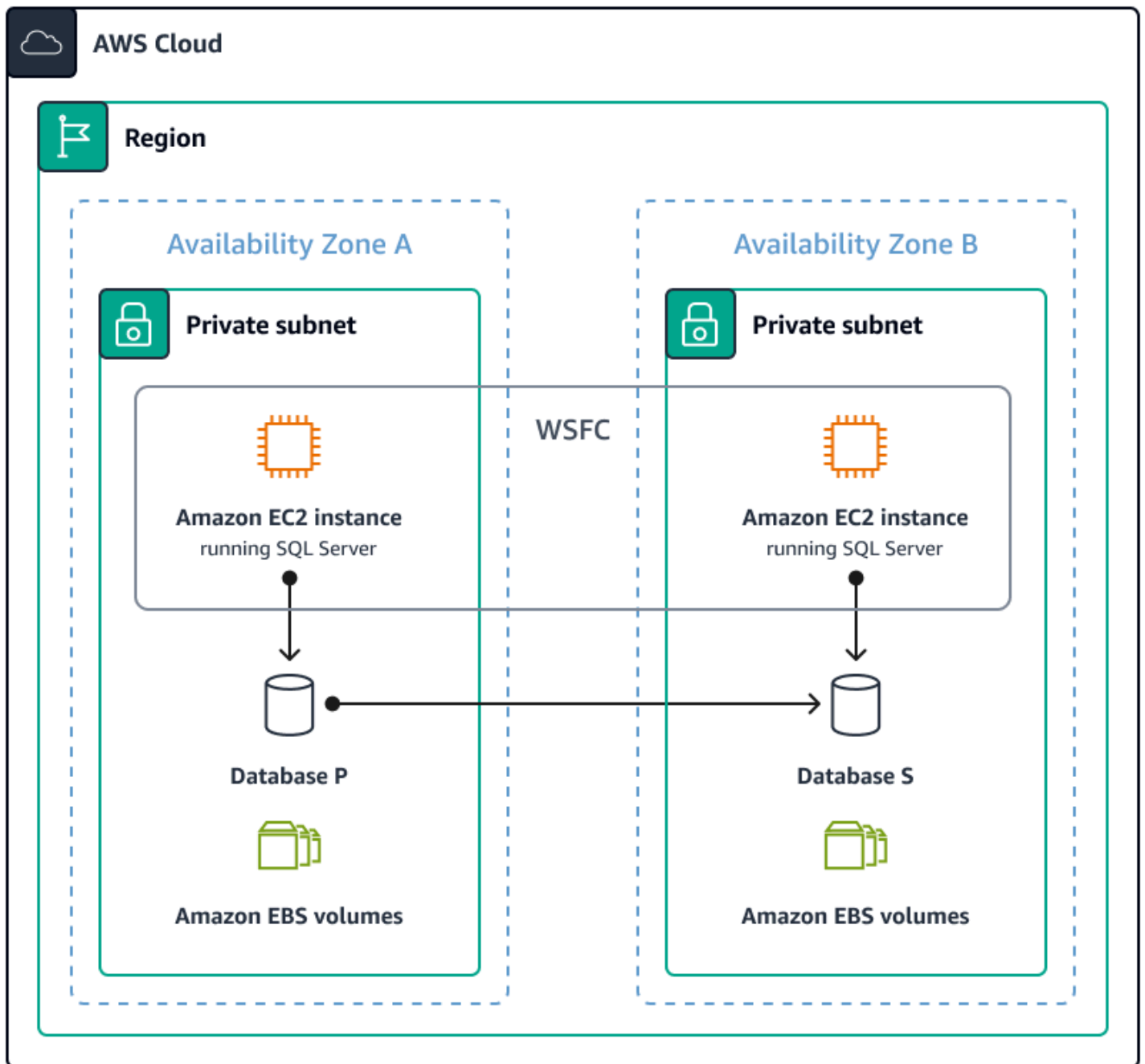
功能

- 在 SQL Server 标准版中可用
- 限制两个副本（主副本和辅助副本）
- 对辅助副本没有读取权限
- 不对辅助副本进行完整性检查

限制

- Support 每个可用性组仅支持一个可用性数据库
- 基本可用性组不能成为分布式可用性组的一部分

下图显示了 Windows 服务器故障转移群集解决方案的示例架构。



SQL Server 始终开启故障转移群集实例

您可以使用故障转移群集实例 (FCI) 来确保数据库的持续运行，同时最大限度地减少停机时间并降低数据丢失的风险。如果您希望在不配置只读副本的情况下实现 SQL Server 数据库的高可用性，FCI 可以提供可靠的解决方案。

与可用性组不同，FCI 无需使用 SQL Server 企业版即可提供可靠的故障转移解决方案。相反，FCI 只需要 SQL Server 标准版许可。您可以使用 FCI 将 SQL Server 的许可成本降低 65-75%。

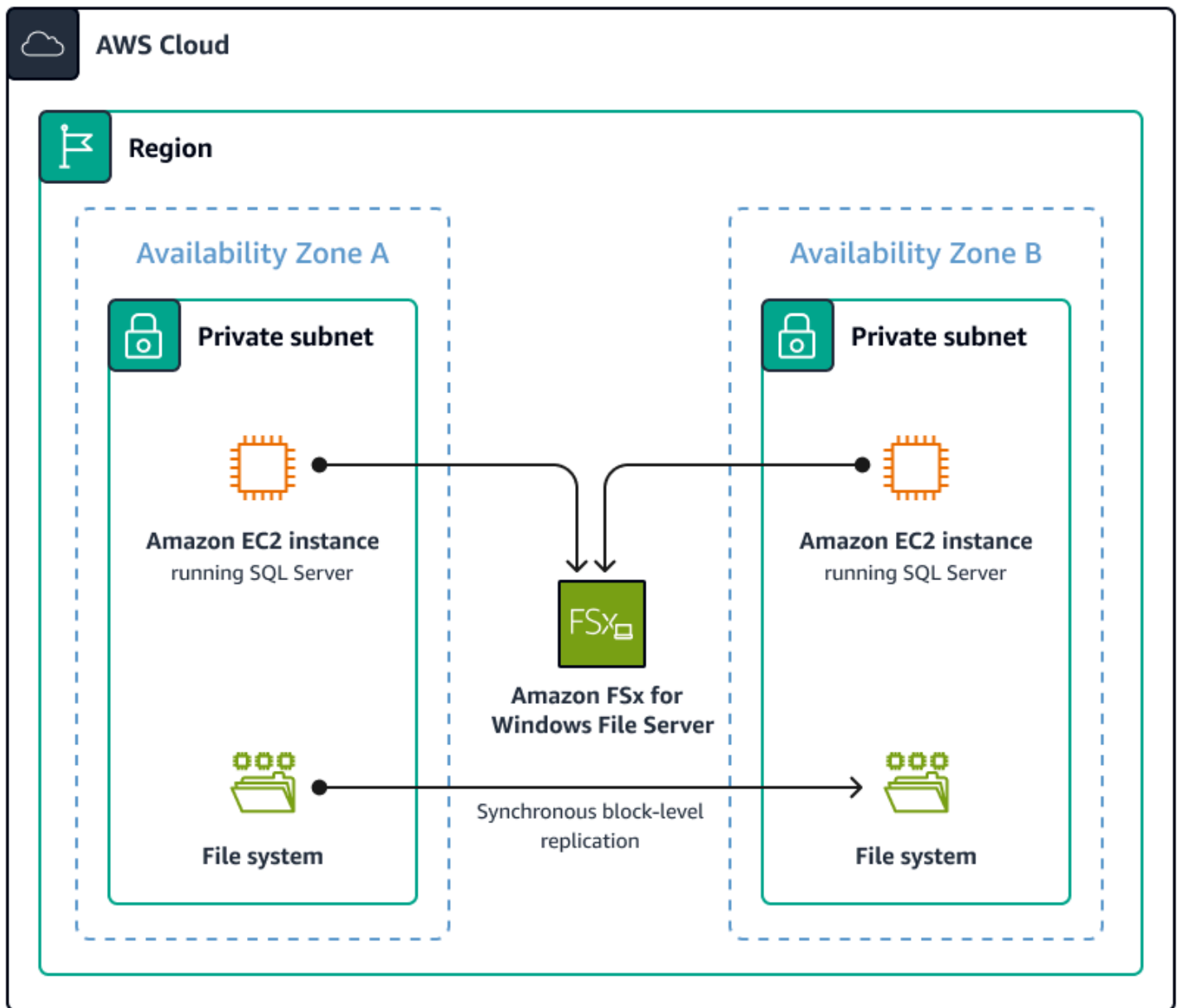
Note

有关 SQL Server 版本之间成本差异的更多信息，请参阅本指南的[比较 SQL Server 版本部分](#)。

请考虑以下事项：

- 适用于 Windows File Server 的 Amazon FSx 提供了一个强大的解决方案，可以满足你的 SQL Server FCI 共享存储需求。你可以使用 FSx for Windows File Server，这样就无需购买存储复制解决方案的许可证，也无需自己管理共享存储。这可以显著节省 30-40% 的成本。有关更多信息，请参阅存储博客上的“[使用适用于 Windows File Server 的 Amazon FSx 简化微软 SQL Server 高可用性部署 AWS](#)”一文。
- 借助[软件保障优势摘要](#)（可下载的 PDF）和自带许可证 (BYOL) 模型，只要辅助服务器处于被动状态，您就可以利用被动故障转移的好处。这可以节省 SQL 许可的成本，因为您不必向群集的被动节点提供许可证。

下图显示了使用适用于 Windows File Server 的 FSx 的 SQL Server FCI 的示例架构。



SIOS DataKeeper

如果您计划在上部署 SQL Server FCI，我们建议您考虑共享存储需求。AWS传统的本地安装通常使用存储区域网络 (SAN) 来满足共享存储需求，但这不是一个可行的选择 AWS。适用于 Windows File Server 的 Amazon FSx 是 SQL Server FCI 的推荐存储解决方案，但它存在一些限制，无法 AWS 在不同的服务器中添加群集服务器。AWS 区域

您可以使用 [SIOS DataKeeper](#) 创建涵盖可用区和区域的 SQL Server FCI，同时将成本降低 58-71%。SIOS DataKeeper 可以帮助您实现 FCI 的高可用性优势。这使得 SIOS DataKeeper 成为组织经济实惠且可靠的解决方案。

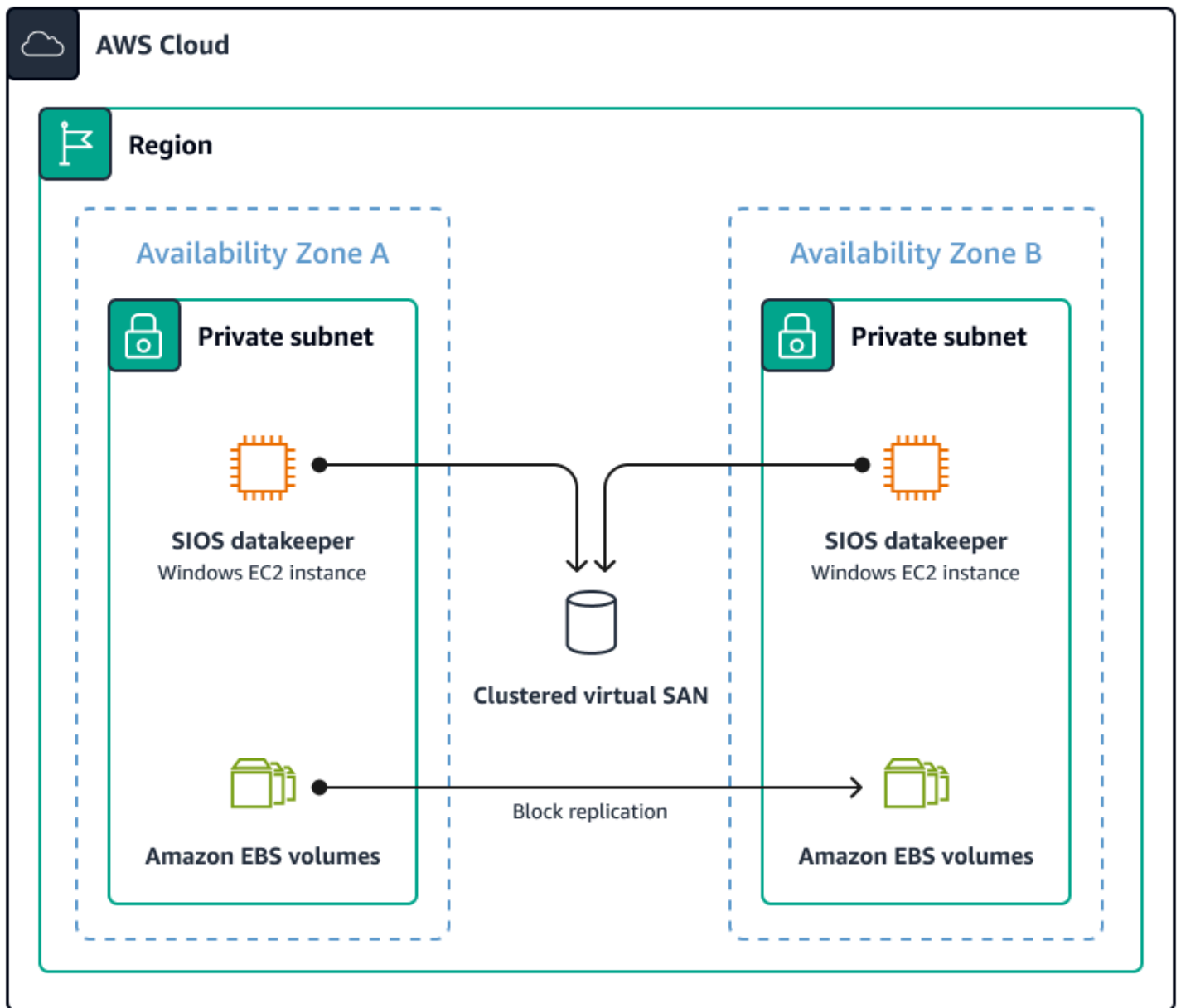
请考虑使用 SIOS DataKeeper 的以下其他好处：

- SIOS 使用本地 EBS 卷 DataKeeper 创建群集虚拟 SAN，并在可用区之间使用同步复制来实现高可用性。为了进行灾难恢复，SIOS DataKeeper 使用区域间的异步复制。
- SIOS 使用 SQL Server 标准版 DataKeeper 提供企业级群集功能。与使用使用 SQL Server 企业版的 SQL Server Always On 可用性组实现高可用性相比，这将 SQL Server 的许可成本降低了 65-75%。借助 SIOS DataKeeper，您可以创建高度可用、灵活且经济实惠的 SQL Server 环境，以满足您的组织需求。

Note

有关 SQL Server 版本之间成本差异的更多信息，请参阅本指南的[比较 SQL Server 版本部分](#)。

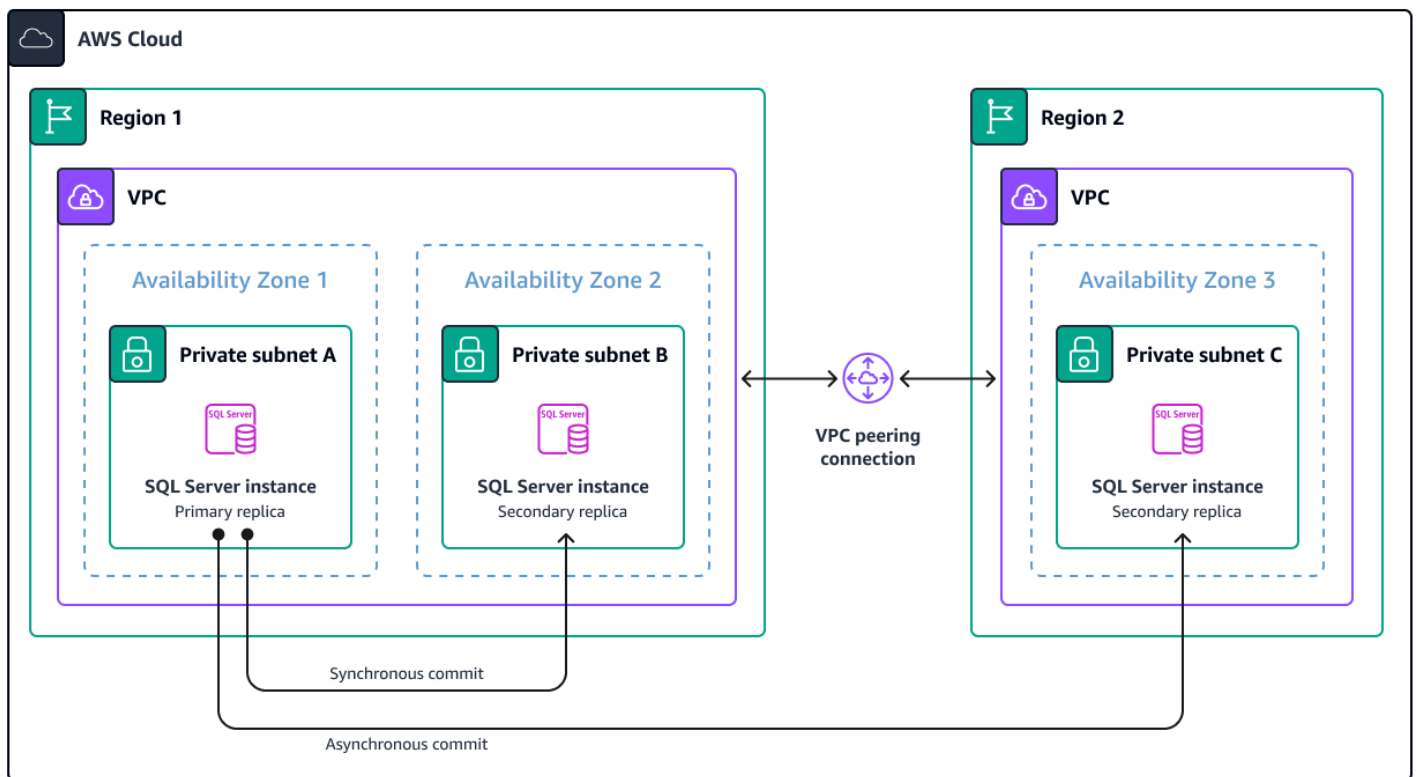
下图显示了使用群集虚拟 SAN 解决方案的 SQL Server FCI 的示例架构。



Always On 可用性组

您可以将 Always On 可用性组用于高可用性和灾难恢复目的。您可以通过跨一个区域的两个可用区部署 SQL Server 来实现高可用性。您可以通过跨区域扩展可用性组来实现灾难恢复。

下图显示了基于 Always On 可用性组的解决方案的架构示例。图中区域 1 中的副本使用同步提交，它提供了可用性组的自动故障转移。区域 2 中的副本正在使用异步提交，这将需要对可用性组进行手动故障转移。



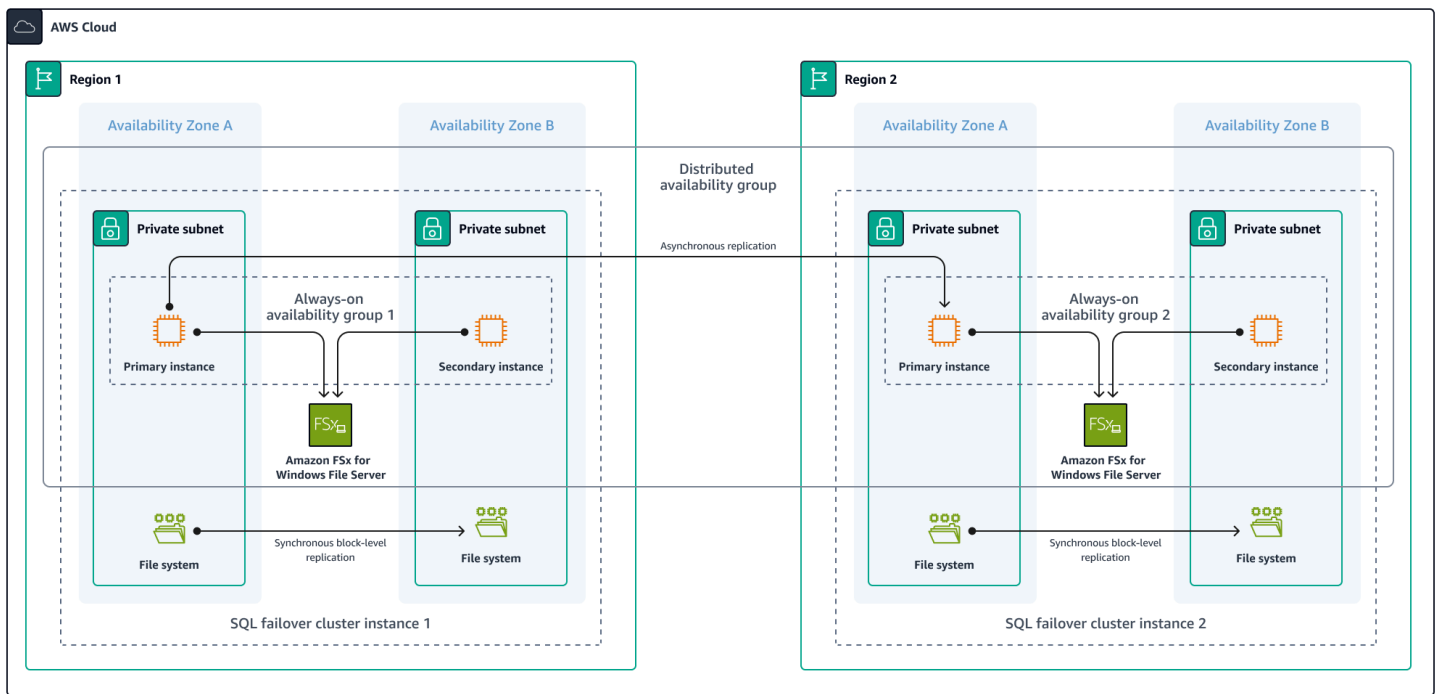
分布式可用性组

对于在可靠性或灾难恢复方面无法妥协的任务关键型 SQL Server 部署，我们建议采用多区域方法。将可用性组分布在多个区域是保持业务连续性和最大限度地减少停机时间的最具弹性的解决方案。

该架构充分利用了适用于 Windows File Server 的 Amazon FSx 的功能，包括共享存储、同步块级复制和 SQL Server FCI。这些功能使您能够创建跨越多个可用区域的高可用性 SQL Server 环境。通过在另一个区域复制此设置，您将获得一个完全冗余的系统，该系统甚至可以处理最严重的中断。该解决方案的与众不同之处在于它提供的灵活性和安全性。分布式可用性组的独立于域的架构使底层 Windows 群集服务器能够加入不同的 Active Directory 域，而基于证书的身份验证可确保最大限度地保护您的 SQL Server 环境，并为多区域灾难恢复策略提供高的 RTO 和 RPO 要求。有关构建多区域架构的信息，请参阅架构博客中的[现场注释：使用 FCI 和分布式可用性组为 SQL Server 构建多区域架构](#)。

AWS

下图显示了使用分布式可用性组的多区域解决方案的示例架构。



日志传送

日志传送是一种久经考验、可靠且经济实惠的方法，可在发生意外中断时保护跨区域的数据库。几十年来，Organizations 一直在使用日志传送来保护其数据。

如果您启用日志传送 AWS，则可以在几分钟内实现 RPO 和 RTO，具体取决于事务和日志传送任务的频率。万一某个区域无法访问，日志传送可确保您的数据安全且可恢复。

请考虑使用日志传送的以下其他好处：

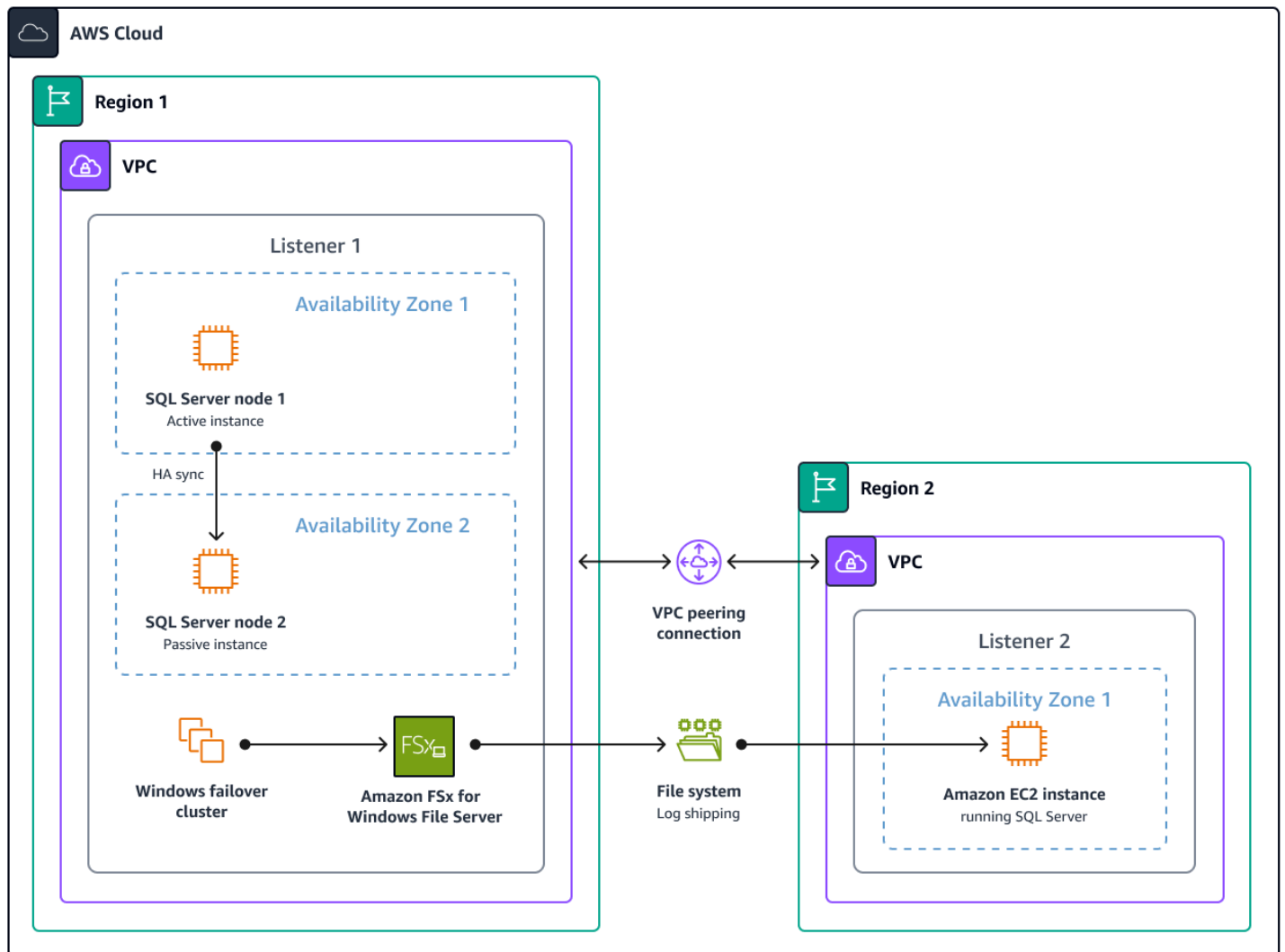
- 使用日志传送实现跨区域的灾难恢复弹性，从而降低成本并满足您的业务需求。日志传送可以降低您的总拥有成本，因为您只需要使用 SQL Server 标准版或 SQL Server Web 版许可证。
- 使用带主动[软件](#)保障的日志传送，降低灾难恢复/被动服务器的许可成本。当您使用带软件保障的日志传送时，只需要对主/主动 SQL Server 进行许可。
- 无需使用 SQL Server 企业版即可在区域之间设置分布式可用性组，从而将 SQL Server 许可成本降低 65-75%。为此，您可以将 SQL Server 标准版和 SQL Server FCI 与日志传送结合使用，以满足您的灾难恢复要求。

Note

有关 SQL Server 版本之间成本差异的更多信息，请参阅本指南的[比较 SQL Server 版本](#)部分。

有关更多信息，请参阅[架构博客中的使用 SQL Server FCI 的日志传送以及适用于 Windows 的 Amazon FSx 配置来扩展 SQL Server 灾难恢复](#)。AWS

下图显示了日志传送解决方案的示例架构。

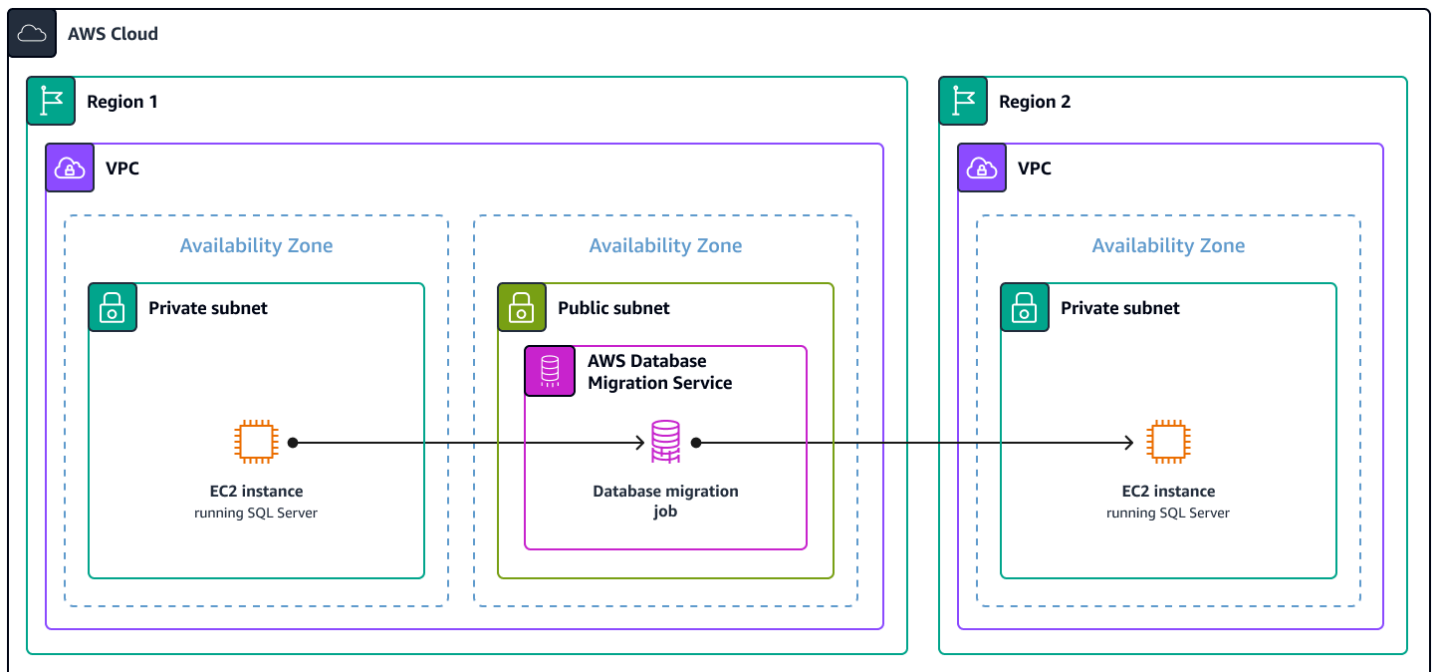


AWS Database Migration Service

您可以使用 AWS Database Migration Service (AWS DMS) 根据您的应用程序需求设计高可用性/灾难恢复解决方案。AWS DMS 使您能够轻松地将数据复制到同一区域 (HA) 或跨区域 (DR) 的辅助 SQL Server 数据库。这种方法在技术上是合理的，可以让您在优化资源利用的同时最大限度地提高 AWS 基础设施投资。

AWS DMS 是一项具有成本效益的服务。您只需为传输过程中使用的 CPU 资源和任何额外的日志存储空间付费。这意味着您可以从该解决方案中受益，而不会产生大量额外成本。您可以使用 AWS DMS 来确保数据可用且可访问，同时最大限度地降低与许可和资源使用相关的成本。

下图显示了基于的解决方案的架构示例 AWS DMS。



AWS Elastic Disaster Recovery

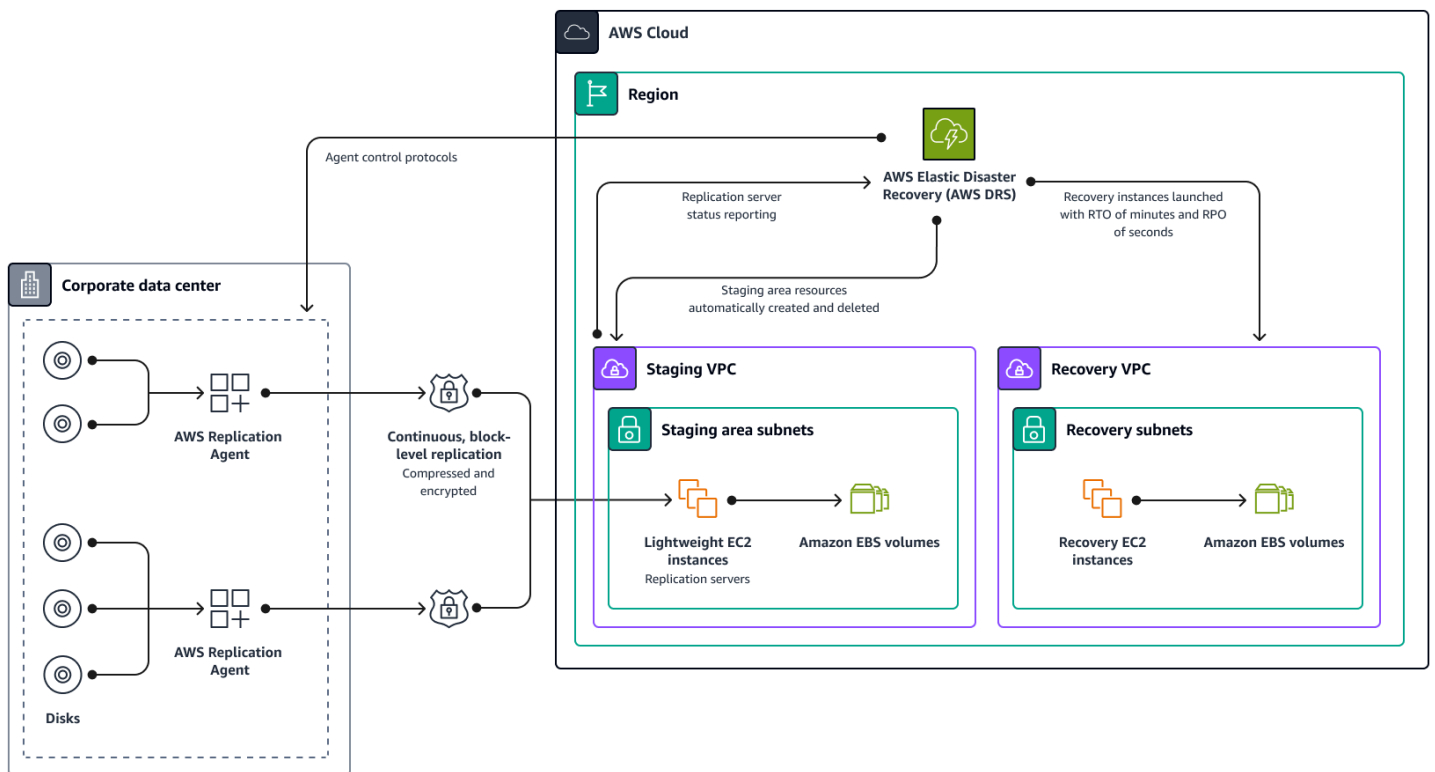
一些组织必须确保所有关键业务应用程序都制定了灾难恢复计划。过去，这些组织中有许多在传统的灾难恢复解决方案上投入了大量资金，这需要您预先构建和维护整个重复的基础架构。这种方法成本高、耗时且难以扩展。

现在，您可以使用 AWS Elastic Disaster Recovery 无需预先构建灾难恢复基础架构。灾难恢复机器只有在必要时才会在 Elastic 灾难恢复中启动，因此您只需在需要时才为实际使用的内容付费。这意味着您可以显著降低软件许可和高性能计算成本。

此外，灾难恢复解决方案的暂存区域包含低成本的亚马逊弹性区块存储 (Amazon EBS) 卷。EBS 卷进一步降低了配置重复资源的成本。这使您能够降低总体灾难恢复成本，同时仍能保持满足业务需求的强大而可靠的灾难恢复解决方案。您可以使用 Elastic 灾难恢复来专注于核心业务活动，AWS 同时为您的灾难恢复解决方案处理底层基础架构。

对于 SQL Server，您可以将弹性灾难恢复用作经济实惠的灾难恢复选项。如果您使用主动软件保障，则涵盖容错、高度可用的 SQL Server 架构中被动节点的许可。但是，要使被动服务器联机，您仍需要支付计算成本。借助 Elastic 灾难恢复，主服务器可以复制到灾难恢复环境，无需维护有效的软件保障，也不必支付灾难恢复计算成本。这种节省的组合可以将您的 SQL Server 灾难恢复成本降低 50% 或更多。

下图显示了基于 Elastic 灾难恢复的解决方案的示例架构。



有关更多信息，请参阅[博文 AWS 客上的 Microsoft 工作负载 AWS Elastic Disaster Recovery 上如何为灾难恢复站点的 SQL Server 设置高可用性](#)。

成本比较

下表比较了本节中介绍的 HA/DR 解决方案的成本。为了进行这种比较，我们做出了以下假设：

- 实例类型 — r5d.xlarge

- 许可证类型 — 包括适用于 Windows 和 SQL Server 的许可证
- 区域 - us-east-1

解决方案	高可用性	灾难恢复	企业版	标准版	费用
日志传送	否	是	是	是	SQL Server 企业版： 32,674.8 美元 (2 个节点) SQL Server 标准版： 14,804.4 美元 (2 个节点)
Always On 可用性组	是	是	是	是，但是基本可用性组 (2 个节点)	SQL Server 企业版： 32,674.8 美元 (2 个节点) SQL Server 标准版： 14,804.4 美元 (2 个节点)
永远在线 FCI	是	否	是	是 (2 个节点)	SQL Server 标准版： 14,804.4 美元
分布式可用性组	是	是	是	不支持	SQL Server 企业版： 65,349.6 美元 (4 个节点)
弹性灾难恢复	否	是	是	是	复制 1 个实例和 1 TB 存储空间的费用约

解决方案	高可用性	灾难恢复	企业版	标准版	费用
					<p>为 107.48 美元/月</p> <p>注意：Elastic 灾难恢复按每台复制服务器按小时计费。无论磁盘数量、存储空间大小、演练或恢复启动次数，或者您要复制的区域如何，成本都是一样的。</p>
SIOS 数据管理器	是	是	是	是	<p>具有软件保障的 Always On 可用性组 (2 个节点, 24 个内核) : 213,480 美元</p> <p>在 SQL Server 标准版上运行的 2 节点 SQL Server 群集, 具有 SIOS DataKeeper 和软件保障 : 61,530 美元 (2 个节点)</p>

解决方案	高可用性	灾难恢复	企业版	标准版	费用
AWS DMS	否	是	是	是	r5.xlarge 实例和 1 TB 存储空间每月 745.38 美元

成本优化建议

我们建议您采取以下后续步骤来选择符合组织要求的高可用性/灾难恢复解决方案：

- 请查看本指南的“[为 SQL Server 工作负载选择合适的 EC2 实例](#)”部分。
- 通过在工作负载峰值期间运行性能计数器来确定工作负载的 IOPS 和吞吐量要求：
 - $IOPS = \text{磁盘读取次数/秒} + \text{磁盘写入次数/秒}$
 - $\text{吞吐量} = \text{磁盘读取字节/秒} + \text{磁盘写入字节/秒}$
- 使用以下存储卷类型可提高性能并节省成本：
 - 用于缓冲池扩展的 NVMe 实例存储 tempdb 和缓冲池扩展
 - 用于存储数据库文件的 io2 卷
- [AWS Trusted Advisor](#) 用于获取有关亚马逊 EC2 上的 SQL Server 成本优化的建议。您无需安装代理即可进行 SQL Server 优化检查。Trusted Advisor 检查您的 Amazon EC2 SQL Server 包含许可证的实例配置，例如虚拟 CPU (vCPU)、版本和版本。然后，根据最佳实践 Trusted Advisor 提出建议。
- 适用 AWS Compute Optimizer 于 Amazon EC2 实例和 Amazon EBS 的正确大小建议。
- 用于设计 [AWS Pricing Calculator](#) 用于成本估算的 HA/DR 策略。
- 要确定是否可以选择从 SQL Server 企业版降级到 SQL Server 标准版，请使用 [sys dm_db_persisted_sku_features](#) 动态管理视图来识别当前数据库中处于活动状态的特定版本的功能。

Note

使用包含许可证的 side-by-side EC2 实例时，需要迁移才能更改 SQL Server 版本。

- 每半年或每年进行一次灾难恢复演练，以更好地设计能够使用定义的 RTO 和 RPO 恢复数据库的设计。这也可以帮助您识别任何架构缺陷。

其他资源

- [使用适用于 Windows File Server 的亚马逊 FSX 简化微软 SQL Server 高可用性部署](#) AWS (存储博客)
- [现场说明：使用 FCI 和分布式可用性组为 SQL Server 构建多区域 AWS 架构](#) (架构博客)
- [为 SQL Server 设计灾难恢复架构 AWS：第 1 部分](#) (AWS 数据库博客)
- [微软 SQL 借助适用于 Windows 的亚马逊 FSX 实现高可用性](#) () YouTube
- [利用@@ 亚马逊 EBS 最大限度地提高微软 SQL Server 性能](#) (AWS 存储博客)
- [将您的本地存储模式与 AWS 存储服务进行比较](#) (AWS 存储博客)
- [计划用 Amazon FSx 文件网关替换数据中心 NAS](#) (AWS 存储博客)
- [优化高可用性 SQL Server 部署的成本 AWS](#) (AWS 存储博客)
- [如何使用 AWS Elastic Disaster Recovery \(开启 Microsoft 工作负载 AWS \) 为 SQL Server Always On 可用性组设置灾难恢复](#)
- [如何在使用恢复的灾难恢复站点上为 SQL Server 设置高可用性 AWS Elastic Disaster Recovery \(开启 Microsoft 工作负载 AWS \)](#)

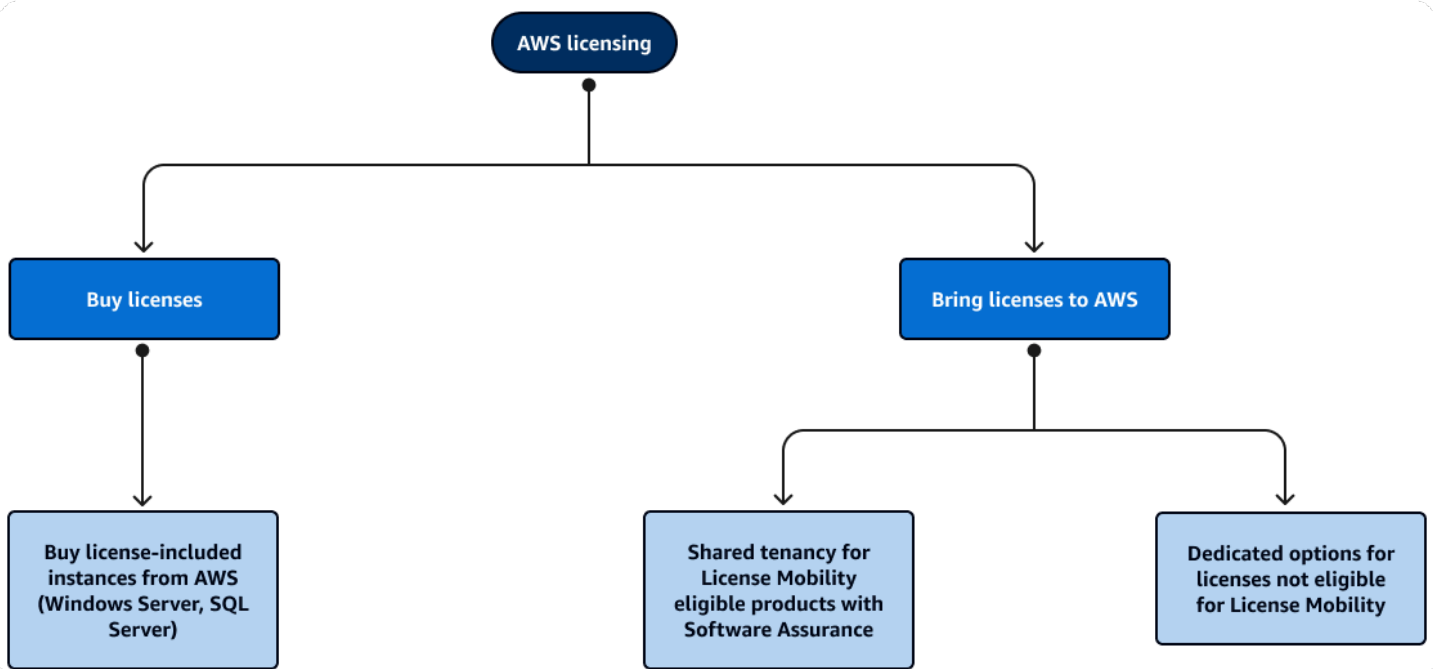
了解 SQL 服务器许可

概述

随着越来越多的企业将其工作负载转移到云端，优化云平台上的成本已成为当务之急。许可是与运行 Microsoft 工作负载相关的最重要成本之一 AWS。本节介绍如何 AWS 通过优化 SQL Server 的 Microsoft 许可来优化成本。

AWS 许可选项

AWS 为许可提供了一系列灵活的成本优化选择。这些许可选项旨在帮助您降低成本、保持合规性并满足您的业务需求。



AWS 将许可证分为三种主要类型：

1. 包含许可证 — 此许可选项允许您按需购买和使用许可证，只需为使用的内容付费。对于需要灵活使用许可并希望避免前期成本的场景，“包含许可证”选项是理想的选择。你可以从一系列 Windows Server、SQL Server 和其他微软产品中进行选择。
2. 自带许可证 (BYOL) 产品，具有许可证移动性 — 此许可选项专为已有许可证并希望在云中使用的场景而设计。AWS 允许客户通过 Microsoft 的许可证[移动性计划将自己的许可证](#)带到云端。您可以将具有许可证移动性的产品（例如带软件保障的 SQL Server (SA)）带到共享或专用租赁，以降低 AWS 实例成本。
3. 没有许可证移动性的 BYOL 产品 — 对于没有许可证移动性的微软产品，例如 Windows Server，AWS 提供了在云中使用这些产品的专用选项。此外，专用主机还提供了在物理核心级别进行许可的机会。这可以为您节省运行工作负载所需的许可证的 50% 或更多。对于大部分时间运行的稳定且可预测的工作负载，专用主机是一个不错的选择。

携带许可证对成本的影响

获得许可证可能会对运行 Microsoft 工作负载的成本产生重大影响 AWS。如果您自带许可证，则无需为在云中运行的实例支付额外的许可费用。这可以节省大量成本。

以下比较显示了全天候运行单个 c5.xlarge 实例的按需每月成本：

- Windows Server + SQL Server 企业版：每月 1353 美元（包括许可证）

- Windows Server + SQL Server 标准版：每月 609 美元 (含许可证)
- 仅限 Windows 服务器：每月 259 美元 (包括许可证)
- 仅限计算 (Linux)：每月 127 美元

归根结底，自带许可证可能会对运行 Microsoft 工作负载的成本产生重大影响 AWS。如果您使用现有的许可证，则可以降低许可成本并节省总 AWS 账单。

许可证优化

AWS 优化和许可评估 (AWS OLA) 可以通过降低计算和许可成本来帮助您优化许可。AWS OLA 旨在评估您对运行的工作负载 AWS 或计划迁移的工作负载的许可要求。AWS OLA 提供了有关优化许可证使用情况的建议。

优化许可使用情况的关键策略之一是[正确调整实例规模](#)。正确调整需要根据工作负载的 CPU、内存和存储要求为其选择正确的实例类型。通过选择适当的实例大小，您可以确保以经济高效的方式使用资源。这可以节省大量成本。

使用 Microsoft 软件许可，软件运行的内核数量是决定许可成本的关键因素。例如，Windows Server 和 SQL Server 许可证通常根据内核数量进行许可。通过正确调整实例规模，您可以减少运行 Microsoft 软件的内核数量，进而降低实例的成本和所需的许可证数量。

成本优化建议

优化许可证是成本优化的关键组成部分 AWS。通过实施正确的策略，您可以降低许可成本，保持合规性，并从许可投资中获得最大价值。本节概述了几种许可证优化的策略。

带上符合条件的 Windows 服务器许可证

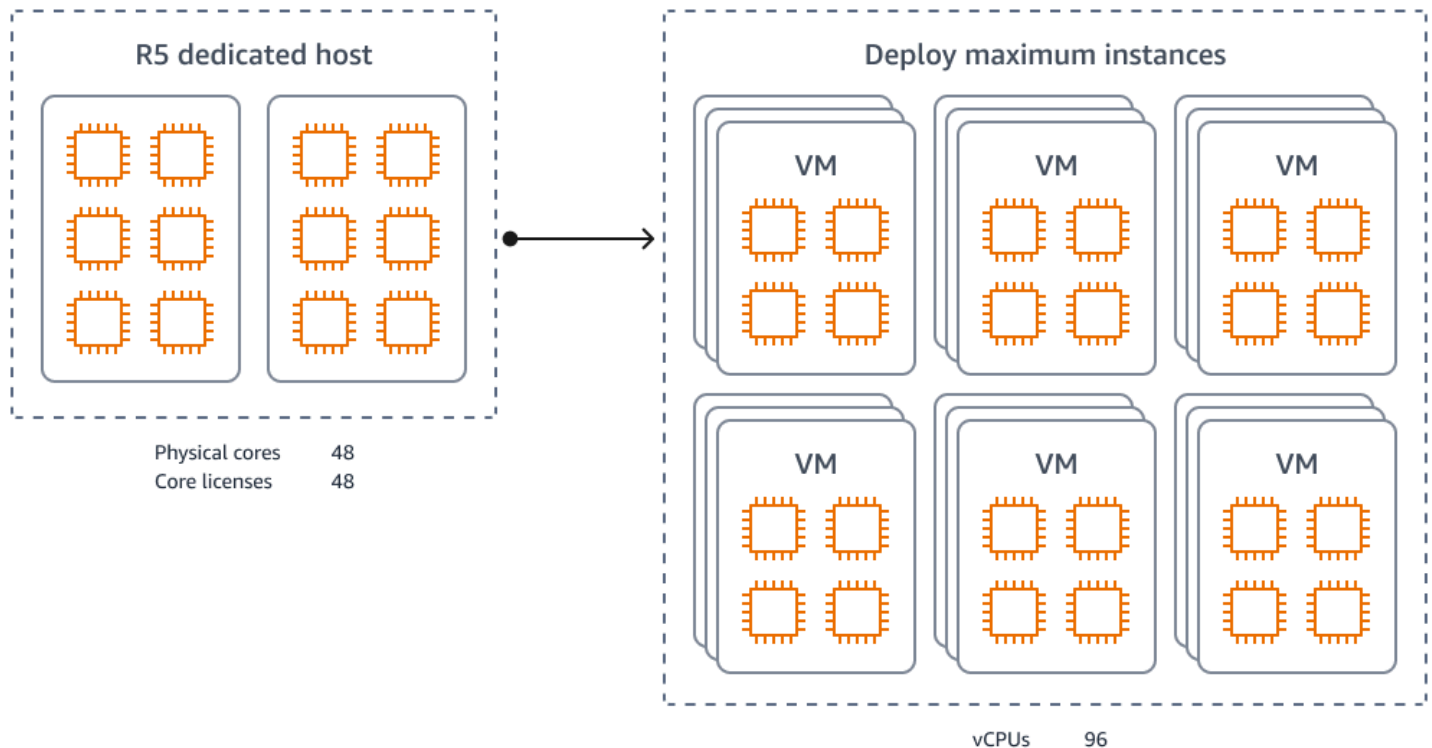
自带 Windows Server 许可证是最有效的许可证优化策略之一。该策略使您能够利用现有投资来减少 AWS 支出。

例如，如果您在 2019 年 10 月 1 日之前购买了许可证，或者根据该日期之前签署的有效企业协议购买了许可证，则可以在[亚马逊 EC2 专用主机](#)上部署 Windows Server 2019 及更早版本。该规则基于微软在 2019 年对没有许可移动性的产品（例如 Windows Server）的许可条款和条件所做的更改，这些产品部署在[上市提供商](#)（例如阿里巴巴或谷歌云）上。AWS 根据新条款，你不能将自己的 Windows Server 许可证带到，AWS 而必须使用包含许可证的实例。但是，如果您在该日期之前购买了永久许可证，那么您仍然可以在 Amazon EC2 专用主机上部署这些 Windows Server 许可证。

物理级别的许可证

物理核心级别的许可使您能够仅许可主机的物理内核，这样您就可以在不影响所需许可证数量的情况下部署最大数量的实例。这通常通过使用 Windows Server 数据中心和 SQL Server 企业版来完成。

举个例子，假设一台具有 48 个内核的 R5 专用主机，相当于 96 个 vCPU。如果您使用 Windows 服务器数据中心版，则只需要 48 个许可证。这使您能够部署多达 96 个 vCPU 的实例组合，如下图所示。



如果您有足够的工作负载来最大限度地提高主机上可以运行的实例数量，则这种方法可能特别具有成本效益。通过在物理核心级别进行许可，您可以避免每个实例的额外许可成本，并实现许可投资的最大价值。

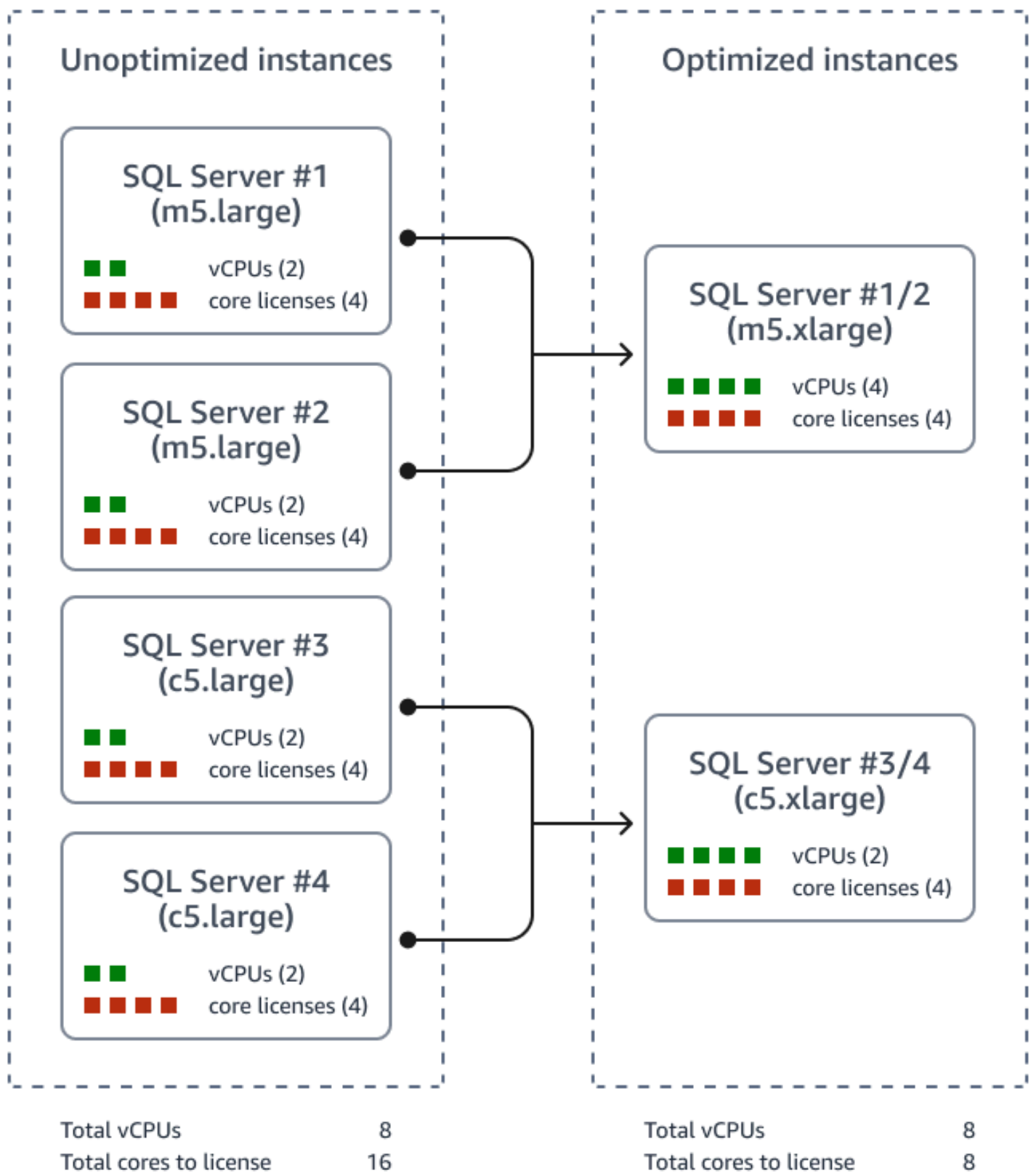
在 SQL Server 的物理核心级别上获得许可

在共享租赁中，SQL Server 许可基于分配给实例的 vCPU 数量。相比之下，对于专用主机，您可以在物理核心级别或 vCPU 级别授权 SQL Server 企业版。

与前面的 R5 专用主机示例一样，如果您在物理核心级别上许可 SQL Server Enterprise 版，则只需要 48 个 SQL Server 企业版许可证即可许可主机。相比之下，在共享租赁中，唯一的选择是通过 vCPU 进行许可，对于相同的工作负载，您必须拥有 96 个 SQL Server 企业版许可。因此，与共享租赁相比，专用主机可以为您节省 50% 的 SQL Server 许可成本。此外，通过携带符合条件的 Windows 许可证，还可以节省实例成本。

整合 SQL 服务器实例

[SQL Server 整合](#)是将多个 SQL Server 实例合并到一台服务器上的过程。SQL Server 要求每个实例至少有四个核心许可证，即使该实例只有两个 vCPU 也是如此。这意味着，在少于四个内核的服务器上运行 SQL Server 可能会导致您过度许可这些实例并使用超出必要数量的许可证。



例如，将两个各有两个 vCPU 的实例整合为一个包含四个 vCPU 的实例，可以将许可要求降低 50%。这是因为只需要四个核心许可证，而不是八个。

有关整合的更多信息，请参阅本指南的 [SQL Server 整合](#) 部分。

降级 SQL Server 版本

[更改 SQL Server 版本](#) 可能是优化许可使用和降低成本的关键策略。SQL Server 的企业版比标准版昂贵得多，因此降级可以节省大量成本。

透明数据加密 (TDE) 和始终开启可用性组是 SQL Server 企业版中的两个热门功能。但是，如果您不需要 SQL Server Enterprise 版的完整功能集，则可以考虑使用具有成本效益的替代功能来替代这些功能。例如，从 SQL Server 2019 开始，您可以在 SQL Server 标准版中获得 TDE。您可以在 FSx for Windows File Server 上使用带有共享存储的故障转移群集来代替 Always On 可用性组，从而在 SQL Server 标准版中实现高可用性。

通过从 SQL Server 企业版降级到 SQL Server 标准版，可以显著降低许可成本。有关更多信息，请参阅 [AWS 存储博客上 AWS 发布的“优化高可用性 SQL Server 部署的成本”](#)。

除了降低许可成本外，降级 SQL Server 版本还可以帮助减少软件保障支出，并帮助您避免将来进行调整。如果您将未使用的许可证退回上架，则可以避免额外的许可成本，并从许可投资中获得尽可能高的价值。

请务必仔细评估您的 SQL Server 工作负载，并确定哪些功能对您的业务需求至关重要。有关更多信息，请参阅 [AWS 规范性指南中的“评估您的环境”](#)，并确定您的 Microsoft SQL Server 数据库是否使用 SQL Server 企业版的特定功能。

如果您选择正确的 SQL Server 版本并使用 SQL Server 企业版功能的替代方案，则可以在保持合规性并满足业务需求的同时节省大量成本。有关降级选项的更多信息，请参阅本指南的 [比较 SQL Server 版本](#) 部分。

在非生产环境中使用 SQL Server 开发人员版

在非生产环境中，您可以通过在本地环境中使用 MSDN 订阅来部署 SQL Server 的许可版本，例如企业版或标准版。但是，MSDN 订阅没有许可证移动性。因此，如果您迁移到 AWS，则无法将这些许可证带过来。您必须改用 SQL Server 开发人员版。

SQL Server 开发者版是 SQL Server 的全功能版本，免费提供。此版本适用于 SQL Server 版本 2016 及更高版本。你可以从微软网站下载。SQL Server 开发者版旨在用于所有非生产环境，例如开发、测试和暂存，前提是它未连接到实时生产数据。

如果您在非生产环境中使用 SQL Server 开发人员版，则可以避免额外的许可成本。有关更多信息，请参阅本指南的“[评估 SQL Server 开发者版](#)”部分。

针对 SQL 服务器工作负载优化 CPU

在某些情况下，由于 RAM 或网络限制等其他因素，您可能需要选择一个 CPU 数量超过工作负载所需的 CPU 数量的实例类型。但是，AWS 提供了一种解决方案来帮助您在这些情况下优化许可成本。

您可以像大多数带有 SQL Server 核心许可证的客户一样，在 EC2 实例上禁用超线程或关闭 CPU，以限制主机可用 CPU 的数量。此选项使您能够利用其他实例功能（例如 RAM），同时还能节省购买额外许可证的成本。此选项仅在使用 BYOL 时可用，如果您使用的是 SQL Server 或 Windows Server 附带的许可证，则此选项不可用。

例如，如果您部署了 r5.4xlarge 实例，因为您的工作负载需要 128 GB 的内存，但您只需要八个 SQL Server 内核，则可以在只有八个活动 CPU 的实例启动时禁用超线程。通过这样做，您可以节省所需的 SQL Server 许可证的 50%，因为您只需要为正在使用的八个内核授予许可即可。

实例类型	vCPU 总数	具有“优化 CPU”功能的主动 vCPU	节省 SQL 服务器许可证
r5.4xlarge	16	8	50%
r5.12xlarge	48	8	83%

如果您在包含许可证的 Windows Amazon 机器映像 (AMI) 上使用优化 CPU 功能，则仍然必须为所有内核上的 Windows 许可付费。这就是为什么正确调整实例大小并定期重新评估实例类型很重要的原因。

如果您调整了实例的大小，则可以确保您的工作负载使用最具成本效益的实例类型。随着新实例类型的 AWS 推出，评估这些新实例能否以更少的内核满足工作负载要求非常重要。

其他资源

- [亚马逊 Web Services 和微软：常见问题解答](#) (AWS 文档)

为 SQL Server 工作负载选择正确的 EC2 实例

⚠ Important

在阅读本节之前，我们建议您先阅读本指南的[了解 SQL Server 许可](#)并为[Windows 工作负载选择正确的实例类型](#)部分。

概述

微软 SQL Server 在亚马逊弹性计算云 (Amazon EC2) 实例上运行已有 15 年多了。AWS 借鉴了这种经验，并利用它来帮助开发 Amazon EC2 实例，以适应从最低规格一直运行到高性能、多区域集群的 SQL Server 工作负载。

为 SQL Server 选择正确的 EC2 实例在很大程度上取决于您的工作负载。了解 SQL Server 的许可方式、内存使用方式以及 SQL Server 功能如何与 Amazon EC2 产品保持一致，可以帮助您找到最适合您的应用程序的 EC2 实例。

本节介绍各种 SQL Server 工作负载，以及如何将它们与某些 EC2 实例配对以将许可和计算成本降至最低。

成本比较

Amazon EC2 允许您自带许可证 (BYOL) 或使用 Windows Server 和 SQL Server 许可按需付费。对于 pay-as-you-go 许可，Windows Server 和 SQL Server 许可证的许可成本包含在 EC2 实例的每小时成本中。例如，您可以拥有不同价格的不同 AMI。AMI 的价格取决于运行 AMI 的 SQL Server 版本。

Windows Server 和 SQL Server 的定价没有逐项列出。您找不到诸如此类的工具的逐项定价。[AWS Pricing Calculator](#)如果您选择不同的许可证内含产品组合，则可以推算许可成本，如下表所示。

EC2 实例	AMI	计算价格	Windows 许可证价格	SQL 许可证价格	总价格
r5.xlarge	Linux (计算定价)	183.96 美元	-	-	183.96 美元
r5.xlarge	Linux + SQL 开发	183.96 美元	\$0	\$0	183.96 美元

EC2 实例	AMI	计算价格	Windows 许可证价格	SQL 许可证价格	总价格
r5.xlarge	Windows 服务器 (LI)	183.96 美元	134.32 美元	-	318.28 美元
r5.xlarge	Windows + SQL 开发人员	183.96 美元	134.32 美元	\$0	318.28 美元
r5.xlarge	Windows + SQL Web (LI)	183.96 美元	134.32 美元	49.64 美元	367.92 美元
r5.xlarge	Windows + SQL Standard (LI)	183.96 美元	134.32 美元	350.4 美元	668.68 美元
r5.xlarge	Windows + SQL 企业版 (LI)	183.96 美元	134.32 美元	1095 美元	1413.28 美元

Note

上表中的定价基于该us-east-1地区的按需定价。

运行 SQL Server 的最具成本效益的方法是保持较低级别的版本，直到需要更高级别版本的功能。有关更多信息，请参阅本指南的“[比较 SQL Server 版本](#)”部分。从 SQL Server Web 版升级到 SQL Server 标准版的成本是 SQL Server 许可成本的七倍多，也是从标准版升级到企业版的成本的三倍多。许可成本的差异是需要考虑的主要因素，本节的其余部分将对此进行探讨。

成本优化场景

以一个示例场景为例，一家跟踪送货车辆的分析公司正在寻求提高其 SQL Server 性能。在 MACO 专家审查了公司的性能瓶颈后，该公司从 x1e.2xlarge 实例过渡到 x2iedn.xlarge 实例。尽管实例大小较小，但对 x2 实例的增强通过使用缓冲池扩展提高了 SQL Server 的性能和优化。这使该公司能够从 SQL Server Enterprise 版降级到 SQL Server 标准版，并将其 SQL Server 许可从 8 个 vCPU 减少到 4 个 vCPU。


优化前：

Server	EC2 实例	SQL Server 版本	月度成本
proddb1	x1e.2xlarge	企业	3,918.64 美元
proddb2	x1e.2xlarge	企业	3,918.64 美元
总计			7,837.28 美元

优化后：

Server	EC2 实例	SQL Server 版本	月度成本
proddb1	x2iedn.xlarge	Standard	1,215.00 美元
proddb2	x2iedn.xlarge	Standard	1,215.00 美元
总计			2,430.00 美元

从 x1e.2xlarge 实例到 x2iedn.xlarge 实例的合并更改使示例客户能够每月在生产数据库服务器上节省 5,407 美元。这使工作负载的总成本降低了 69%。

 Note

上表中的定价基于该us-east-1地区的按需定价。

成本优化建议

内存优化型实例

SQL Server 最重要的方面之一是了解其对内存的依赖。SQL Server 尝试使用操作系统未使用的所有可用内存（默认安装最多为 2 TB）。它这样做是出于性能方面的考虑。处理内存中的数据比经常从磁盘中提取数据、进行更改然后将其写回磁盘要高得多。相反，SQL Server 会尝试从连接的数据库中加载尽可能多的数据，并将这些数据保留在 RAM 中。对数据所做的更改发生在内存中，稍后会硬化到磁盘。

Note

有关 SQL Server 如何写入更改的详细说明，请参阅 Microsoft 文档中的[编写页面](#)。

由于 SQL Server 在内存量较大时性能更好，因此我们通常建议从 [Amazon EC2 内存优化的实例类型](#) 开始。内存优化型实例用途广泛，并提供各种不同的选项。R 系列的 vCPU 与 RAM 的比率为 1 比 8，可选英特尔处理器、AMD 处理器、增强的网络、增强的 EBS 性能、实例存储和增强的处理器速度。对于内存密集型工作负载，还有 X 系列结合了许多相同的选项，将 vCPU 与 RAM 的比率扩展到 1 比 32。由于内存优化实例的多功能性，您可以将其应用于各种形式和大小的 SQL Server 工作负载。

低于最低资源的工作负载 (少于 4 个 vCPU)

尽管某些用例适用于突发性 (T3) 实例，但我们建议您通常避免对 SQL Server 工作负载使用突发性实例。SQL Server 的许可基于分配给实例的 vCPU 数量。如果 SQL Server 一天中的大部分时间都处于闲置状态并且正在获得突发积分，则需要为未充分利用的 SQL 许可证付费。此外，SQL Server 的最低许可证要求为每台服务器 4 个内核。这意味着，如果您的 SQL Server 工作负载不需要 4 个 vCPU 的计算能力，则需要支付未使用的 SQL Server 许可。在这些情况下，最好将[多个 SQL Server 实例整合](#)到一个更大的服务器上。

使用最少资源 (内存小于 64 GB) 的工作负载

许多 64 GB RAM 以下的 SQL Server 工作负载并不优先考虑高性能或高可用性。对于这些类型的工作负载，如果应用程序受微软的许可限制，则该应用程序可能非常适合 SQL Server Web 版。

Important

根据微软的许可条款，SQL Server Web 版的使用案例受到限制。SQL Server Web 版的许可仅支持公用且可通过互联网访问的网页、网站、Web 应用程序和 Web 服务。它不能用于支持 line-of-business 应用程序 (例如，客户关系管理、企业资源管理和其他类似的应用程序)。

SQL Server 网络版最多可扩展到 32 个 vCPU 和 64 GB 内存，比 SQL Server 标准版便宜 86%。对于资源不足的工作负载，使用像 r6a 这样的 AMD 内存优化实例也是将计算和 SQL 许可成本降至最低的好方法，它的计算价格比英特尔同类实例低 10%。

资源平均值 (内存小于 128 GB) 的工作负载

SQL Server 标准版用于大多数 RAM 不超过 128 GB 的 SQL Server 工作负载。SQL Server 标准版比 SQL Server 企业版便宜 65-75%，并且可以扩展到 48 个 vCPU 和 128 GB 内存。由于 128 GB 的

RAM 限制通常在 48 个 vCPU 限制之前达到，因此大多数希望避免升级到 SQL Server 企业版的客户都关注这个问题。

SQL Server 有一个叫做[缓冲池扩展](#)的功能。此功能使 SQL Server 能够使用磁盘的一部分来充当 RAM 的扩展。缓冲池扩展与超快速存储（例如 [Amazon EC2](#) 实例存储中使用的 NVMe 固态硬盘）结合使用时效果很好。包含实例存储的 Amazon EC2 实例在实例名称中用“d”表示（例如 r5d、r6id 和 x2iedn）。

缓冲池扩展不能取代普通 RAM。但是，如果您需要超过 128 GB 的 RAM，则可以对 EC2 实例（例如 r6id.4xlarge 和 x2iedn.xlarge）使用缓冲池扩展来延迟企业版许可的升级。

高性能工作负载（减少 128 GB 内存）

由于需要高性能的 SQL Server 工作负载依赖大量资源，因此成本优化具有挑战性。但是，了解 EC2 实例的差异可以防止您做出错误的选择。

下表显示了各种内存优化的 EC2 实例及其性能限制。

	r5b	r6idn	r7iz	x2iedn	x2iezn
处理器	3.1 GHz 第二代英特尔至强处理器	3.5 GHz 第三代英特尔至强处理器	3.9 GHz 第四代英特尔至强可扩展处理器	3.5 GHz 第三代英特尔至强处理器	4.5 GHz 第二代英特尔至强处理器
CPU: RAM 比率	1:8	1:8	1:8	1:32	1:32
最大 vCPU	96	128	128	128	48
最大内存	768 GB	1,024 GB	1,024 GB	4,096 GB	1,536 GB
实例存储	–	NVMe SSD (4x 1900 GB)	–	NVMe SSD (2x 1900 GB)	–
io2 区块快车	支持	支持	支持	支持	–
EBS 最大 IOPS	260000	350,000	160000	260000	80,000

	r5b	r6idn	r7iz	x2iedn	x2iezn
EBS 最大吞吐量	60 Gbps	80 Gbps	40Gbps	80 Gbps	19 Gbps
最大网络带宽	25 Gbps	200 Gbps	50 Gbps	100 Gbps	100 Gbps

每个实例都有不同的用途。了解您的 SQL Server 工作负载可以帮助您选择最适合自己的实例类型。

有关属性的详细信息：

- r5b — r5b 中的“b”属性表示此实例类型侧重于高 EBS 性能。在第五代内存优化实例中，r5b 是首选。它是第一个使用 io2 Block Express 卷并达到最大存储 IOPS 为 260,000 的实例类型。对于高 EBS 性能需求，r5b 实例类型仍然是一种经济实惠的替代方案。
- r6idn — 与上一代相比，第六代内存优化型实例提供了相当大的改进。r5b 的 EBS 性能增强在 r6idn 上更进一步，将最大 IOPS 提高到 350,000。r6idn 还有一个用于存放 tempdb 的实例存储卷和缓冲池扩展，以进一步提高 SQL Server 的性能。
- x2iedn — x2iedn 与 r6idn 类似。它提供类似级别的增强型 EBS、增强型网络和 NVMe 固态硬盘实例存储，但对于高内存工作负载和低 CPU 数量（更低 SQL Server 许可成本），它的 vCPU 与 RAM 的比率为 1:32。
- x2iezn — x2iezn 中的“z”属性表示此实例类型侧重于高处理器性能。Cascade Lake 处理器的全核涡轮频率高达 4.5 GHz。我们建议您在希望将 vCPU 数量保持在较低水平的场景中使用此 EC2 实例，并采用 1:32 的 vCPU 与 RAM 的比率。这反过来又可以降低 SQL Server 许可成本。
- r7iz — r7iz 中的“z”属性表示此实例类型侧重于高处理器性能。Sapphire rapids 处理器的全核涡轮频率高达 3.9 GHz。与 x2iezn 实例一样，r7iz 优先考虑高频处理器性能，但 vCPU 与 RAM 的比率为 1:8。

其他资源

- [通用型 Amazon EC2 实例](#) (AWS 文档)
- [比较工具](#) (Vantage)
- [许可 — SQL Server](#) (AWS 文档)

整合实例

本节重点介绍将多个 SQL Server 实例合并到同一台服务器上的成本优化技术，以最大限度地降低许可成本并最大限度地提高资源利用率。

概述

创建实例是安装 SQL Server 数据库引擎过程的一部分。SQL Server 实例已完成安装，包含其自己的服务器文件、安全登录和系统数据库（主数据库、模型数据库、msdb 和 tempdb）。由于一个实例拥有自己的所有文件和服务，因此您可以在同一个操作系统上安装多个 SQL Server 实例，而这些实例不会相互干扰。但是，由于这些实例都安装在同一台服务器上，因此它们共享相同的硬件资源，例如计算、内存和网络。

在生产环境中，通常每台服务器只使用一个 SQL Server 实例，这样“繁忙”的实例就不会过度使用共享的硬件资源。与依赖资源治理相比，为每个 SQL Server 实例提供自己的操作系统和自己的资源是一个更好的界限。对于需要大量 RAM 和 CPU 资源的高性能 SQL Server 工作负载尤其如此。

但是，并非所有 SQL Server 工作负载都使用大量资源。例如，出于合规性或安全目的，一些组织为每位客户分配自己的专用 SQL Server 实例。对于较小的客户端或通常不处于活动状态的客户端，这意味着使用最少的资源运行 SQL Server 实例。

正如[微软 SQL Server 2019：许可指南](#)中所述，每台运行 SQL Server 的服务器必须至少占用四个 CPU 许可证。这意味着，即使你运行的服务器只有两个 vCPU，你仍然必须为四个 vCPU 的 SQL Server 授予许可。根据[微软的公开 SQL Server 定价](#)，如果你使用 SQL Server 标准版，则相差 3,945 美元。对于使用最少资源运行多台服务器且使用单个 SQL Server 实例的组织来说，必须许可未使用资源的综合成本可能很高。

成本优化场景

本节探讨了一个示例场景，该场景比较了运行四台 Windows Server 服务器（每台服务器只有一个 SQL Server 实例）与同时运行多个 SQL Server 实例的大型 Windows Server 服务器之间的区别。

如果每个 SQL Server 实例只需要两个 vCPU 和 8 GB 内存，那么 SQL Server 许可证每台服务器的总成本为 7,890 美元，此外每小时的计算成本为 0.096 美元。

EC2 实例	vCPU	RAM	Price	需要许可的 vCPU	SQL Server 许可总成本
m6i.large	2	8	0.096	4	7,890 美元

将其扩展到四台服务器，SQL Server 许可证的总成本为 31,560 美元，每小时的计算成本为 0.384 美元。

EC2 实例	vCPU	RAM	Price	需要许可的 vCPU	SQL Server 许可总成本
4x m6i.large	2	32	0.384	16	31,560 美元

如果您将所有四个 SQL Server 实例合并到一个 EC2 实例中，则计算资源和计算的总量将保持不变。但是，通过消除不必要的 SQL Server 许可成本，您可以将运行工作负载的总成本降低 15,780 美元。

EC2 实例	vCPU	RAM	Price	需要许可的 vCPU	SQL Server 许可总成本
m6i.2xlarge	8	32	0.384	8	15,780 美元

Note

在前面的表格中，计算成本显示了 us-east-1 在该地区运行 Windows Server 的 Amazon EC2 服务器的按小时按需定价。SQL Server 标准版许可成本指的是 [微软的 SQL Server 公开定价](#)。

成本优化建议

如果您正在考虑整合 SQL Server 实例，那么最担心的是要整合的每个实例的资源消耗。为了更好地了解每台服务器上的工作负载模式，获取长期的性能指标非常重要。一些用于监控资源消耗的常用工具包括 [亚马逊 CloudWatch](#)、[Windows 性能监控器 \(perfmon\)](#) 和 SQL Server 的 [本机监控工具](#)。

在分析您的 SQL Server 工作负载是否可以组合使用相同的服务器资源而不相互干扰时，我们建议您考虑以下问题：

- 稳定状态期间消耗了哪些资源（CPU、内存和网络带宽）？
- 峰值期间消耗了哪些资源（CPU、内存和网络带宽）？
- 尖峰多久发生一次？尖峰是否一致？
- 一台服务器的资源峰值是否与另一台服务器的资源峰值重合？
- SQL Server 使用的存储 [IOPS](#) 和 [吞吐量](#) 是多少？

如果您想继续制定合并 SQL Server 实例的计划，请参阅 AWS 云运营与迁移博客上的“[在一个 Amazon EC2 实例上运行多个 SQL Server 实例](#)”一文。这篇文章提供了有关如何在 SQL Server 中进行配置更改以添加其他实例的说明。在开始之前，请考虑在同一台服务器上安装多个实例时的细微差别：

- 默认 SQL Server 数据库实例已命名 MSSQLSERVER 并使用端口 1433。
- 安装在同一台服务器上的每个其他实例都是一个“命名”数据库实例。
- 每个命名的实例都有一个唯一的实例名称和一个唯一的端口。
- 必须运行 [SQL Server 浏览器](#) 来协调指定实例的流量。
- 每个实例都可以使用不同的位置来存放数据库数据文件和单独的登录。
- SQL Server 的 [最大服务器内存设置](#) 必须根据每个实例的性能需求进行配置，其组合总内存还会为底层操作系统留出足够的内存。
- 您可以使用 SQL Server [本机备份和恢复](#) 功能，也可以 [AWS DMS](#) 用于迁移或整合。

其他资源

- [SQL Server 许可数据表](#) (AWS 云运营和迁移博客)
- [SQL Server 多实例设置博客文章](#) (AWS 云操作和迁移博客)
- [SQL Server 最佳实践指南](#) (AWS 规范性指导文档)

比较 SQL 服务器版本

概述

微软 SQL Server 许可是 Windows 工作负载环境中最大的支出之一。SQL Server 的许可成本很容易超出运行工作负载的计算成本。如果你选择了错误的版本，你可以为你没有使用或甚至不需要的功能付费。本节比较了以下 SQL Server 版本，包括其功能和相关成本：

- 企业 — SQL Server 企业版提供数据中心功能，包括高性能、无限虚拟化和多种商业智能 (BI) 工具。
- 标准 — SQL Server 标准版为小型组织和部门提供基本的数据管理和商业智能。
- Web — SQL Server 网络版适用于网络托管商或网络增值提供商 (VAP) 的公司。此版本提供了较低的总拥有成本，并且为小型到大型的网络资产提供了可扩展性和可管理性功能。

⚠ Important

您可以使用 SQL Server Web 版仅支持公共网页、可通过互联网访问的网页、网站、Web 应用程序和 Web 服务。您不能使用 SQL Server Web 版来支持 line-of-business 应用程序（例如客户关系管理或企业资源管理应用程序）。

- 开发人员-SQL Server 开发者版包含企业版的所有功能，但仅用于开发目的。
- Express — SQL Server Express 版是一个免费的数据库，可用于学习或构建桌面应用程序。您可以将 Express 版更新到其他版本。

📘 Note

SQL Server 评估版提供 180 天的试用期。

成本影响

你可以从微软经销商处购买 SQL Server 许可证，然后 AWS 使用软件保障将其带到。或者，您可以将 SQL Server 许可证与包含许可证的 Amazon EC2 AMI 的 pay-as-you-go 型号一起使用。

如果您从 Microsoft 经销商处购买 SQL Server 许可证，则核心许可证以两个一包的形式出售，并且每台服务器必须至少获得四个内核的许可。下表显示了企业版和标准版之间的成本比较。

版本	SQL Server 企业版 (2 个内核套装)	SQL Server 标准版 (2 个内核套装)	节省成本
2022	15,123 美元	3,945 美元	74%
2019	13,748 美元	3,586 美元	74%

📘 Note

上表中的定价基于微软对 [SQL Server 2022](#)和[SQL Server 2019](#) 的公开定价。

以下成本比较显示了使用包含许可证的 Amazon EC2 AMI 托管不同版本的 SQL Server。在此比较中，SQL Server 托管在该区域的 r6i.xlarge (4 个 vCPU) 上。us-east-1

实例	计算成本	Windows 许可费用	SQL Server 许可证成本	Total
r6i.xLarge (Linux)	183.96 美元	–	–	183.96 美元
r6i.xLarge + Windows	183.96 美元	134.32 美元	–	318.28 美元
r6i.xLarge + SQL Server 网络版	183.96 美元	134.32 美元	49.35 美元	367.63 美元
r6i.xLarge + SQL Server 标准版	183.96 美元	134.32 美元	350.4 美元	668.68 美元
r6i.xLarge + SQL Server 企业版	183.96 美元	134.32 美元	1,095 美元	1,413.28 美元

通过为您的工作负载选择正确的 SQL Server 版本，您最多可以节省 95% 的 SQL Server 许可成本。下表比较了 r6i.xlarge 实例上的 SQL Server 许可证成本。

版本	节省百分比
标准版与企业版对比	68%
网络版与标准版对比	86%
Web 与企业版的比较	95%

在大多数情况下，组织会从企业版切换到标准版，但在某些情况下，可以从标准版或企业版切换到网络版。

成本优化建议

您可以根据扩展限制、高可用性、性能和安全性为工作负载选择最佳版本。下表显示了各个 SQL Server 版本所支持的功能。这可以帮助您决定使用哪个版本。这种比较适用于 [SQL Server 2016 SP1 及更高版本](#)。

扩展限制

下表比较了不同 SQL Server 版本的扩展限制。

功能	企业版	标准版	网络版	极速版
SQL Server 数据库引擎、SQL Server 分析服务 (SSAS) 或 SQL Server 报告服务 (SSRS) 的单个实例使用的最大计算容量	操作系统最大值	仅限于 4 个插槽或 24 个内核中较小的插槽	仅限于 4 个插槽或 16 个内核中较小的插槽	限于 4 个插槽或 4 个内核中较小的插槽
每个 SQL Server 数据库引擎实例的缓冲池最大内存	操作系统最大值	128 GB	64 GB	1410 MB
每个 SQL Server 数据库引擎实例的缓冲池扩展的最大容量	配置的最大内存为 32 倍	配置的最大内存的 4 倍	不适用	不适用
最大关系数据库大小	524 PB	524 PB	524 PB	10 GB
Columnstore 缓存或内存优化数据的最大内存	操作系统最大值	32 GB	16 GB	352 MB

如果您的应用程序需要少于 16 个内核 (32 个 vCPU) 和 64 GB 的内存，则可以从 SQL Server 网页版开始评估。如果您的工作负载需要超过 64 GB 的内存或其他高可用性选项，则必须升级到 SQL Server 标准版。

您可以使用 SQL Server Web 版来支持公共和可访问互联网的网页、网站、Web 应用程序和 Web 服务，但不能使用 SQL Server Web 版来支持业务线应用程序。有关 SQL Server 网络版用例的更多信息，请联系[微软许可支持部门](#)或你的微软经销商。

您可以将 SQL Server 标准版用于处理最多 24 个内核 (48 个 vCPU) 和 128 GB 内存的工作负载。但是，您可以使用[缓冲池扩展](#)来启用 SQL Server 标准版以利用[本地实例存储](#)，例如 r6id EC2 实例中存在的存储空间。这样可以将内存扩展到最大内存配置的四倍。当内存需求开始增加时，这种功能组合可能会延迟服务器升级到企业版的时间。

您可以通过在缓冲池和页面[预期寿命计数器中查找数据库页](#)来识别内存利用率。页面预期寿命告诉你该页在刷回磁盘之前在内存中还有多长时间。此计数器的默认值为 300。如果页面在内存中驻留数小时或数天，则有可能减少分配的内存。

高可用性

下表比较了不同 SQL Server 版本的高可用性功能。

功能	企业版	标准版	网络版	极速版
服务器核心支持 1	是	是	是	是
日志传送	是	是	是	不支持
数据库镜像	是	全安全模式	只能作为证人	只能作为证人
Backup 压缩	是	是	否	否
始终开启故障转移群集实例	16 个节点	2 个节点	否	否
Always On 可用性组	最多 8 个辅助副本，包括 2 个同步辅助副本	否	否	否
基本可用性组	否	2 个节点	否	否

功能	企业版	标准版	网络版	极速版
在线页面和文件恢复	是	否	否	否
在线索引	是	否	否	否
在线架构更改	是	否	否	否
快速恢复	是	否	否	否
镜像备份	是	否	否	否
热添加内存和 CPU	是	否	否	否
加密备份	是	是	否	否
混合备份到微软 Azure (备份到 URL)	是	是	否	否
用于灾难恢复的故障转移服务器	是	是	否	否
用于实现高可用性的故障转移服务器	是	是	否	否

其他常见功能

下表比较了不同 SQL Server 版本的最常见功能。有关功能的详尽列表，请参阅 Microsoft 文档中的 [SQL Server 2019 版本和支持的功能](#)。

功能	企业版	标准版	网络版	极速版
(性能) 资源调控器	是	否	否	否

功能	企业版	标准版	网络版	极速版
(安全) 透明数据库加密 (TDE)	是	是	是	不支持
(安全) 可扩展密钥管理 (EKM)	是	否	否	否
(复制) 甲骨文出版物	是	否	否	否
(复制) 点对点事务复制	是	否	否	否
更改数据捕获	是	是	否	否

SQL 服务器开发者版

所有非生产工作负载，例如开发、QA、测试、暂存和 UAT 环境，都可以使用 SQL Server 开发人员版来节省 100% 的 SQL Server 许可成本。[下载 SQL Server](#) 后，您可以使用共享租赁在 EC2 实例上安装 SQL Server 开发人员版。SQL Server 开发者版不需要专用基础架构。有关更多信息，请参阅本指南对 [SQL Server 开发人员版](#) 的建议。

切换版本

对于现有工作负载，从一个版本切换到另一个版本需要进行大量测试。最佳做法是检查在企业版或标准版上运行的工作负载，以查看是否使用了特定版本的功能，以及这些功能是否有其他解决方案。例如，如果您想查看您的数据库是否正在使用任何企业级功能，则可以在所有数据库上运行 [动态管理视图 \(DMV\)](#)，如以下示例命令所示。

```
SELECT feature_name FROM sys.dm_db_persisted_sku_features; GO
```

有些企业版功能无法在 T-SQL 中捕获，例如作为 SQL 维护作业一部分的在线重新索引。这些必须手动验证。

迁移注意事项

如何许可 SQL Server 将决定您切换版本的选项。包括 SQL Server AMI 在内的 AMI 将许可成本包含在 EC2 实例的价格中，许可成本与 AMI 绑定。您可以使用 [AWS 账单代码](#) 来验证您的 AMI 中包含的 SQL

Server 版本。对于 AWS 包含许可证的实例，在操作系统内更改 SQL Server 版本不会更改与 AMI 关联的账单。您必须使用运行新版 SQL Server 的 AMI 将数据库迁移到新 EC2 实例。

如果您自带许可证，则可以获得更大的灵活性。通常仍建议迁移到另一个运行新版本的 EC2 实例。如果某些事情没有按计划进行，则可以轻松进行故障恢复。但是，如果必须使用现有服务器，则仍然可以 side-by-side 安装 SQL Server 并在实例之间迁移数据库。有关 side-by-side 版本降级的更多详细步骤，请参阅 MssqlTips 网站上的 [SQL Server 版本升级和降级](#)。

其他资源

- [SQL Server 2022 的版本和支持的功能](#) (微软学习)
- [sys.dm_db_persisted_sku_features \(Transact-SQL\)](#) (微软学习)
- [你应该使用哪个版本的 SQL Server ?](#) (布伦特·奥扎尔无限)
- [AWS Pricing Calculator](#) (AWS)

评估 SQL Server 开发者版

概述

[SQL Server 开发者版](#)是 SQL Server 的免费版本，包含企业版的所有功能，可以在任何非生产环境中使用。在无法使用 Microsoft 开发者网络 (MSDN) 许可的云中，SQL Server 开发者版是节省成本的好方法，无需为开发和测试工作负载提供许可证。对于运行大型开发和测试环境并寻求减少不必要成本的团队来说尤其如此。

生产环境被定义为可由应用程序（例如互联网网站）的最终用户访问的环境，其用途不仅仅是收集该应用程序的反馈或验收测试。构成生产环境的其他场景包括：

- 连接到生产数据库的环境
- 支持灾难恢复或生产环境备份的环境
- 至少有一段用于生产的环境，例如在活动高峰期轮换到生产环境的服务器

有关更多许可信息，请参阅 AWS 文档中的 [亚马逊 Web Services 和 Microsoft：常见问题](#)。

成本影响

如果您将 SQL Server 开发者版用于非生产工作负载，则可以百分之百地节省当前 SQL Server 开发和测试环境的许可成本。

SQL Server 版本	SQL Server 企业版 (2 个内核套装)	SQL Server 标准版 (2 个内核套装)	SQL 服务器开发者版
2022	15,123 美元	3,945 美元	免费
2019	13,748 美元	3,586 美元	免费

Note

上表中的定价基于微软对 [SQL Server 2022](#)和[SQL Server 2019](#) 的公开定价。

下表比较了在该地区使用 4 个 vCPU 运行并使用按需定价的不同 SQL Server 版本的成本。us-east-2 这适用于依赖中包含许可证的实例的场景。AWS

EC2 实例	AMI	计算价格	Windows 许可证价格	SQL Server 许可证价格	总价格
r5.xlarge	Linux (计算定价)	183.96 美元	–	–	183.96 美元
r5.xlarge	Linux + SQL 服务器开发者版	183.96 美元	\$0	\$0	183.96 美元
r5.xlarge	Windows 服务器 (LI)	183.96 美元	134.32 美元	–	318.28 美元
r5.xlarge	Windows + SQL Server 开发者版	183.96 美元	134.32 美元	\$0	318.28 美元
r5.xlarge	Windows + SQL Server 网络版 (LI)	183.96 美元	134.32 美元	49.64 美元	367.92 美元

EC2 实例	AMI	计算价格	Windows 许可证价格	SQL Server 许可证价格	总价格
r5.xlarge	Windows + SQL Server 标准版 (LI)	183.96 美元	134.32 美元	350.4 美元	668.68 美元
r5.xlarge	Windows + SQL Server 企业版 (LI)	183.96 美元	134.32 美元	1095 美元	1413.28 美元

成本优化场景

在一家数据完整性公司进行新的收购之后，它希望将新收购的工作负载从其当前位置迁移到托管主机提供商上，以便与托管主机提供商中的其他工作负载进行整合 AWS Cloud。初步定价显示，该公司的 SQL Server 工作负载运行成本将 AWS 比当前托管服务提供商高 60%。一家 MACO SME 评估了估算结果，发现客户实际上是在为其开发和测试环境的托管主机提供商处的 SQL Server 许可证付费。通过在迁移期间将非生产工作负载切换到 SQL Server 开发人员版，该公司将其 SQL Server 许可减少了 40%。

亚马逊 EC2 中包含 SQL Server 许可证

如果您的 EC2 实例上有 SQL Server，且这些实例使用的是[包含许可证的 AMI](#)，则无法直接从企业版转换为开发者版。包含许可证的实例的许可费用与 AMI 挂钩。即使从操作系统中卸载 SQL Server，仍会向 EC2 实例收取许可费用。

要转换为开发者版，必须[下载 SQL Server 开发者版](#)，将其安装在新的 EC2 实例上，然后迁移数据库。您可以使用多种方法在 EC2 实例之间迁移 SQL Server 数据库。有关更多信息，请参阅《将微软 SQL Server 数据库迁移到 AWS Cloud 指南》中的 SQL Server 数据库[迁移方法](#)。您还可以使用[自动 SQL Server 开发人员解决方案](#)来准备计划迁移到的新实例。

亚马逊 EC2 上的 SQL Server BYOL

如果您有使用 BYOL 的 SQL Server 实例，则可以从以下就地转换或 side-by-side 降级选项中进行选择：

- 从微软网站下载[SQL Server 开发者版](#)。有关手动或自动安装的说明，请参阅 AWS 博客上的“[自动化 SQL Server 开发人员部署](#)”一文。

- 使用 [SQL Server 本机备份和还原](#) 将数据库从一个 SQL 实例迁移或分离/附加到另一个 SQL 实例。
- 使用 [自动化工具](#) 进行批量部署。

Note

SQL Server 开发者版仅适用于非生产环境。

其他资源

- [自动部署 SQL Server 开发人员以在 EC2 上部署 SQL Server 开发人员版](#) (AWS 博客)
- [SQL 2022 定价](#) (微软)
- [SQL 2019 定价](#) (微软)
- [许可选项](#) (亚马逊 EC2 上的 SQL Server)
- [AWS Pricing Calculator](#) (亚马逊 EC2 上的 SQL Server 文档)
- [微软 SQL Server 2019 许可指南](#) (从微软下载)
- [SQL Server 2022 开发者版](#) (从微软下载)

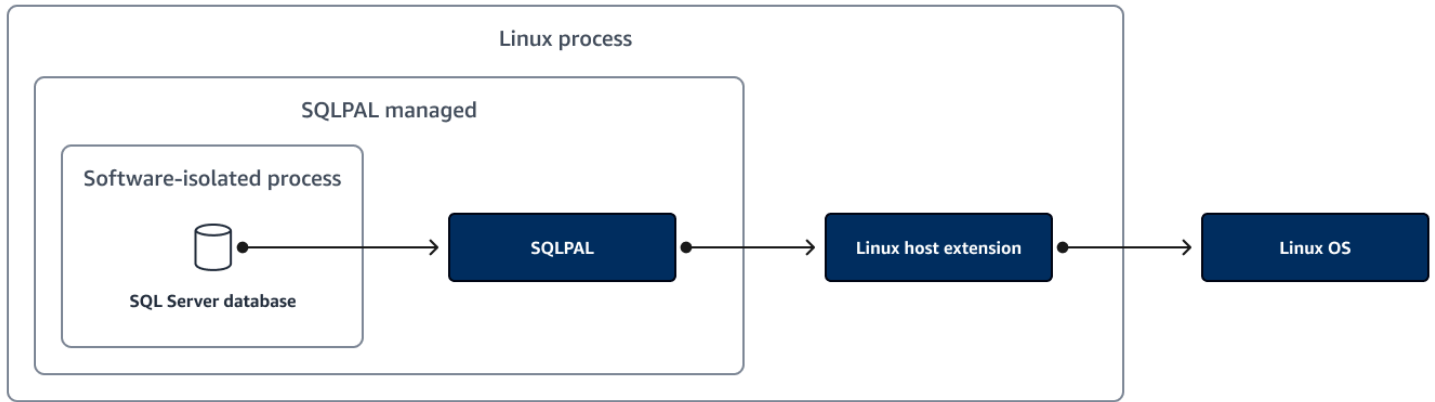
在 Linux 上评估 SQL 服务器

概述

自 SQL Server 2017 以来，可以在 Linux 操作系统上安装 SQL Server。Linux 上的 SQL Server 已为企业做好准备，具有灵活性、高性能、安全功能、降低的总体拥有成本、HA/DR 功能和出色的用户体验。你可以从 Windows 服务器上的 SQL Server 切换到 Linux 上的 SQL Server，以节省 Windows 服务器的许可成本。

对于 Linux，SQL Server 可以部署在红帽企业 Linux (RHEL)、SUSE Linux 企业服务器 (SLES)、Ubuntu 和亚马逊 Linux 2 上。SQL Server 数据库引擎在 Windows Server 和 Linux 上的运行方式相同，但是在使用 Linux 时，某些任务有一些根本性的变化。在 Linux 和 Windows 上运行 SQL Server Always On 应用程序之间的一个关键区别与故障转移群集有关。如果您在 Windows 服务器主机上部署 Always On 可用性组，则可以利用 [Windows 服务器故障转移群集 \(WSFC\)](#) 和 Active Directory 作为支持故障转移群集的内置功能。但是，WSFC 和 Active Directory 都无法支持 Linux 上的故障转移群集。如果要在 Linux 上启动 SQL Server 的故障转移群集，则可以使用 [ClusterLabs Pacemaker](#) 和 [AWS Launch Wizard](#) 来简化集群设置和 Linux 实例上的 SQL 安装。

Windows 和 Linux 上的 SQL Server 共享一个共同的代码库。也就是说，SQL Server 核心引擎根本没有更改为在 Linux 上运行。SQL Server 引入了平台抽象层 (SQLPAL)，如下图所示。



SQLPAL 负责抽象 SQL Server 和底层操作系统之间的调用和通信。主机扩展只是一个原生 Linux 应用程序。低级操作系统函数是用于优化 I/O、内存和 CPU 使用率的本机调用。当主机扩展启动时，它会加载并初始化 SQLPAL，然后启动 SQL Server。SQLPAL 启动独立的软件进程，为其余代码提供所需的翻译。在 SQL Server 架构中添加这一新层意味着，无论使用哪种操作系统，都可以使用使 SQL Server 在 Windows 上如此强大的企业级核心功能和优势。

成本影响

对于 r5.2xlarge 实例，在每种情况下，Windows Server 的许可成本降低约为 268 美元。与使用更便宜的 SQL Server 版本相比，降低的比例在服务器总成本中所占的比例更高。下表显示了节省的成本。

实例	版本	Windows 上的 SQL Server 的月度费用	Linux 上的 SQL Server 每月费用	节省成本
r5.2xlarge	Web	735 美元	466 美元	37%
r5.2xlarge	Standard	1,337 美元	1,068 美元	20%
r5.2xlarge	企业	2,826 美元	2,558 美元	10%

Note

上表中的价格估算基于该 us-east-1 地区的按需定价，可以直接在中查看 [AWS Pricing Calculator](#)。

举一个示例场景，即中小型企业领域的独立软件供应商客户希望在开发环境中节省成本。他们已经在了一组 Windows 服务器上使用了 SQL Server 开发者版。通过从带有 SQL Server 的 Windows 开发者版切换到带有 SQL Server 开发者版的 Linux，ISV 客户可以节省 33% 的开发工作量。下表显示了该场景的以下估计成本。

估计	月度成本
Windows + SQL 服务器	9,307.72 美元
Linux + SQL 服务器	6,218.36 美元
预计节省的成本	3,089.36 美元 (33%)

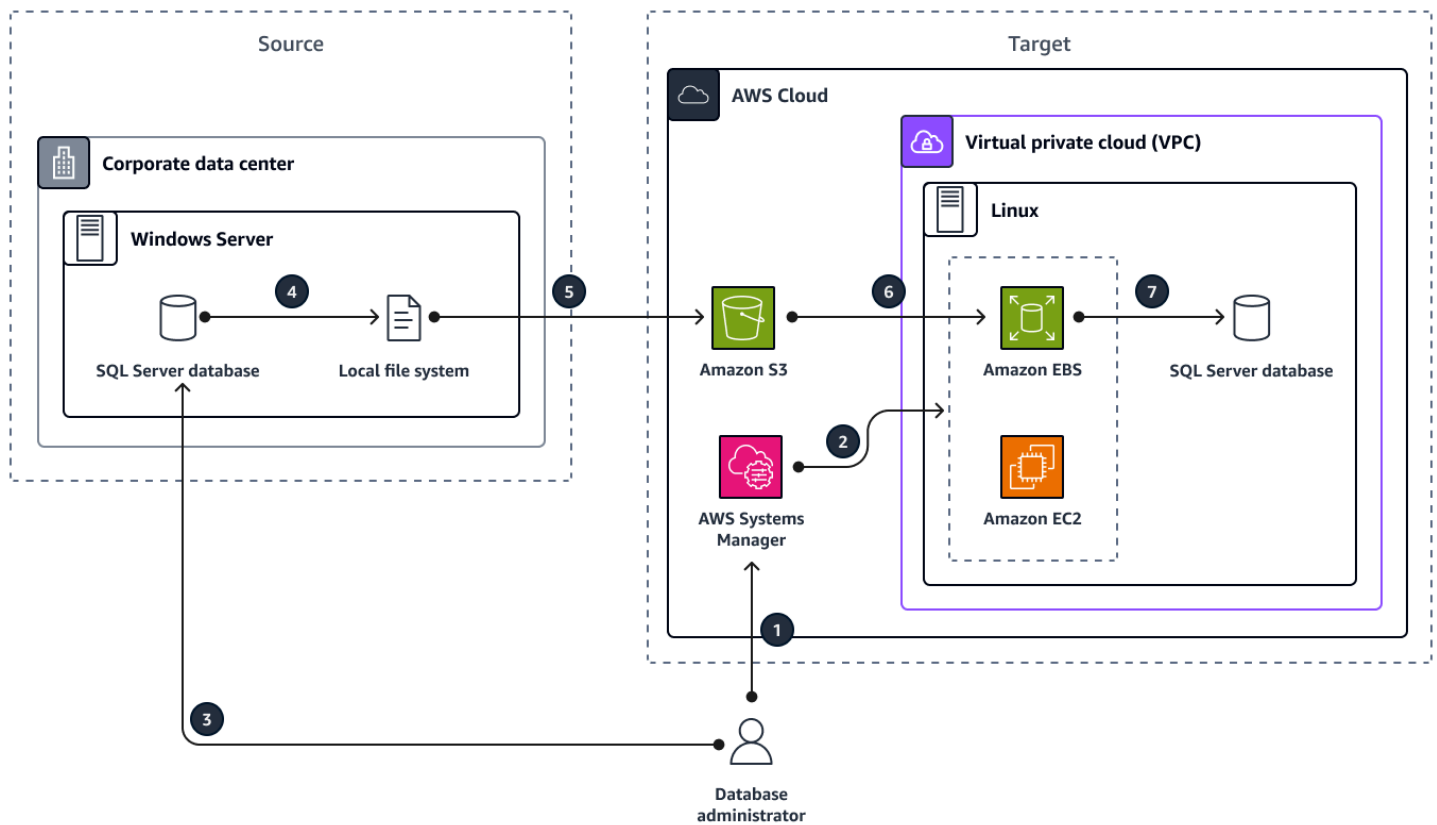
在另一个示例场景中，一家公司将包含许可证的 SQL Server EC2 实例从 Windows 迁移到 Linux。该公司每年在 Windows Server 许可成本上总共节省 30 万美元，约占其总账单的 20%。AWS

成本优化建议

我们建议您考虑以下几点：

- 从 SQL Server 2017 开始，支持 Linux 上的 SQL 服务器。
- 为了帮助切换，你可以使用[微软 SQL Server 数据库的 Windows 到 Linux 平台重构助手](#)。平台重塑助手是一款脚本工具，通过检查常见的不兼容性、从 Windows 主机导出数据库，然后将数据库导入在 Ubuntu 16.04 上运行 Microsoft SQL Server 2017 的 EC2 实例，可以帮助你将有 SQL Server 工作负载从 Windows 迁移到 Linux 操作系统。
- 你也可以使用 SQL Server 中的[备份和恢复](#)功能从 Windows 上的 SQL Server 切换到 Linux。
- 你可以使用，轻松快速地部署到 Linux 或 Ubuntu 上的 SQL Server。[AWS Launch Wizard](#) Launch Wizard 可以根据您的应用程序需求，在独立和高可用性场景中 Linux 或 Ubuntu 上部署 SQL Server。有关更多信息，请参阅 Microsoft Work [loads AWS 博客上的“始终在 Linux 上部署到 SQL Server”](#) 的文章。AWS Launch Wizard

下图显示了使用微软 SQL Server 数据库的 Windows 到 Linux 平台重构助手的解决方案架构。



其他资源

- [Linux 上的 SQL Server 概述](#) (微软学习)
- [Linux 上的 SQL Server 安装指南](#) (微软学习)
- 使用 AWS Launch Wizard (AWS 博客@@ [上的微软工作负载](#)) 在 Linux 上部署到 SQL Server
- [Linux 上的高可用性 SQL Server](#) (AWS 开源博客)

优化 SQL 服务器备份策略

概述

大多数组织都在寻找合适的解决方案来保护其在 [Amazon EC2](#) 上的 SQL Server 上的数据，以满足其当前的恢复点目标 (RPO) (自上次备份以来的最大可接受时间) 和恢复时间目标 (RTO) (服务中断和服务恢复之间的最大可接受延迟) 的要求。如果您在 EC2 实例上运行 SQL Server，则有多个选项可用于创建数据备份和恢复数据。保护 Amazon EC2 上的 SQL Server 数据的备份策略包括以下内容：

- 使用启用了 Windows 卷影复制服务 (VSS) 的 [Amazon Elastic Block Store \(Amazon EBS \)](#) 快照或 [AWS Backup](#) 进行服务器级备份

- 在 SQL Server 中使用[本机备份和恢复](#)进行数据库级备份

[数据库级本机备份](#)有以下存储选项：

- 带有 [Amazon EBS](#) 卷的本地备份
- 使用[适用于 Windows File Server 的亚马逊 FSx 或适用于 ONTAP 的亚马逊 FSx 进行网络文件系统备份 NetApp](#)
- 使用网络备份到亚马逊简单存储服务 (Amazon S3) Simple Service [AWS Storage Gateway](#)
- 直接备份到适用于 SQL Server 2022 的亚马逊 S3

本节执行以下操作：

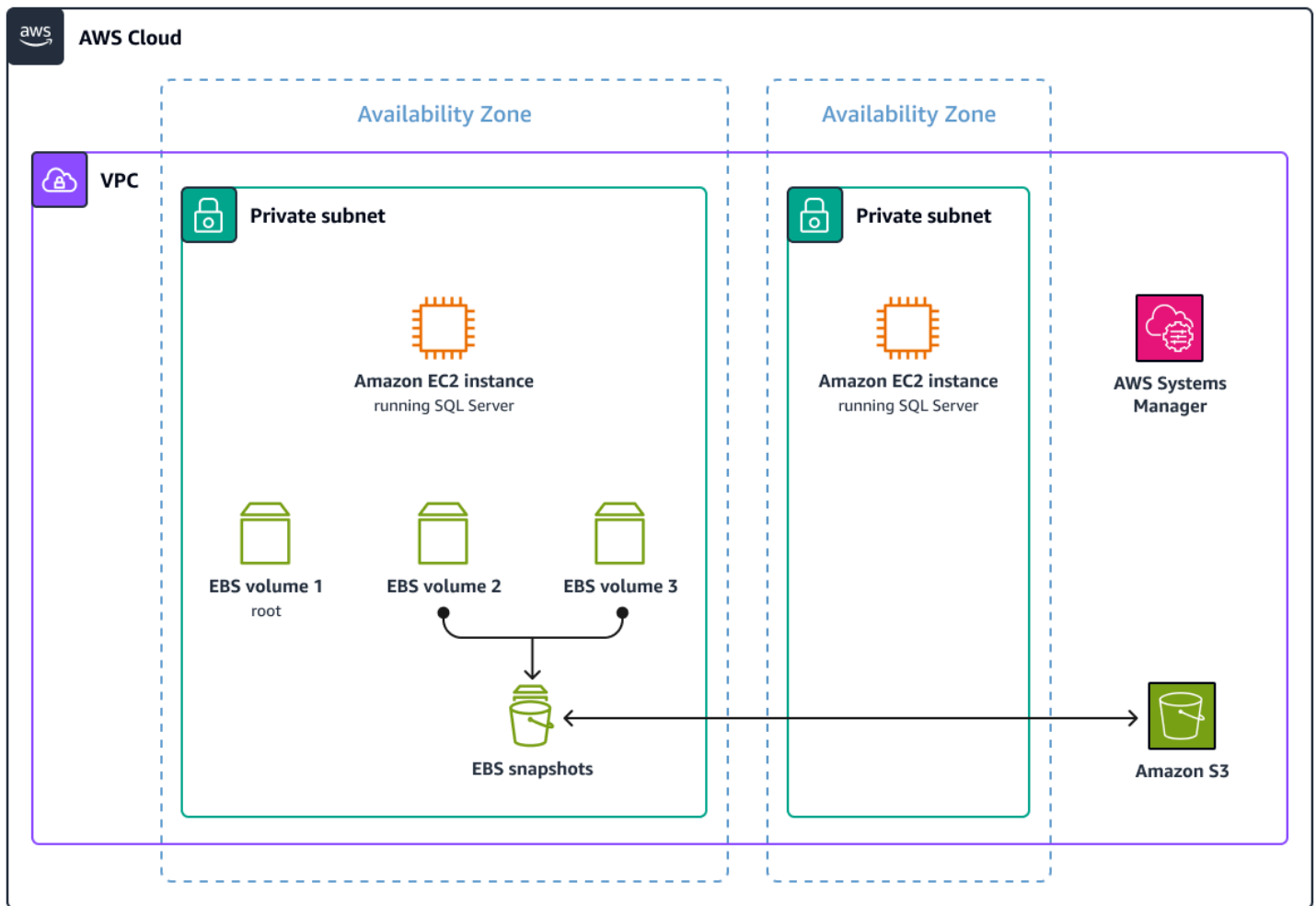
- 重点介绍可帮助您节省存储空间的功能
- 比较不同后端存储选项之间的成本
- 提供深入文档的链接，以帮助实施这些建议

使用支持 VSS 的快照进行服务器级备份

支持 VSS 的快照架构使用 AWS Systems Manager [运行命令在](#) SQL Server 实例上安装 VSS 代理。您还可以使用 Run Command 调用整个工作流程，包括将操作系统和应用程序缓冲区刷新到磁盘、暂停 I/O 操作、拍摄 EBS 卷的 point-in-time 快照，然后恢复 I/O。

这个 Run Command 会为所有连接到目标实例的 EBS 卷创建自动化快照。您还可以选择排除根卷，因为用户数据库文件通常存储在其他卷上。如果您对多个 EBS 卷进行条带化来为 SQL Server 文件创建单个文件系统，Amazon EBS 还支持使用单个 API 命令的崩溃一致性多卷快照。有关[支持应用程序一致性 VSS 的 EBS 快照](#)的更多信息，请参阅 Amazon EC2 文档中的[创建 VSS 应用程序一致性快照](#)。

下图显示了使用支持 VSS 的快照进行服务器级备份的架构。



请考虑使用支持 VSS 的快照的以下好处：

- 数据库实例的第一个快照包含完整数据库实例的数据。相同数据库实例的后续快照为 [增量](#) 快照，这意味着仅会保存在最新快照后更改的数据。
- EBS 快照提供 point-in-time 恢复功能。
- 您可以[从快照还原到新的 SQL Server EC2 实例](#)。
- 如果实例使用 Amazon EBS 加密，或者实例中的数据库使用 TDE 加密，则该实例或数据库将使用相同的加密方式自动恢复。
- 您可以复制您的[自动跨区域备份](#)。
- 当您从快照恢复 EBS 卷时，应用程序可以立即访问该卷。这意味着，在从快照中还原 SQL Server 的一个或多个底层 EBS 卷后，您可以立即将 SQL Server 置于在线状态。
- 默认情况下，还原后的卷会在应用程序首次尝试读取卷时从 Amazon S3 获取底层块。这意味着从快照还原 EBS 卷后，性能可能会出现延迟。卷最终捕获到标称性能。但是，您可以使用[快速快照还原 \(FSR\)](#) 快照避免出现该延迟。

- 您可以使用 [EBS 快照的生命周期管理](#)。

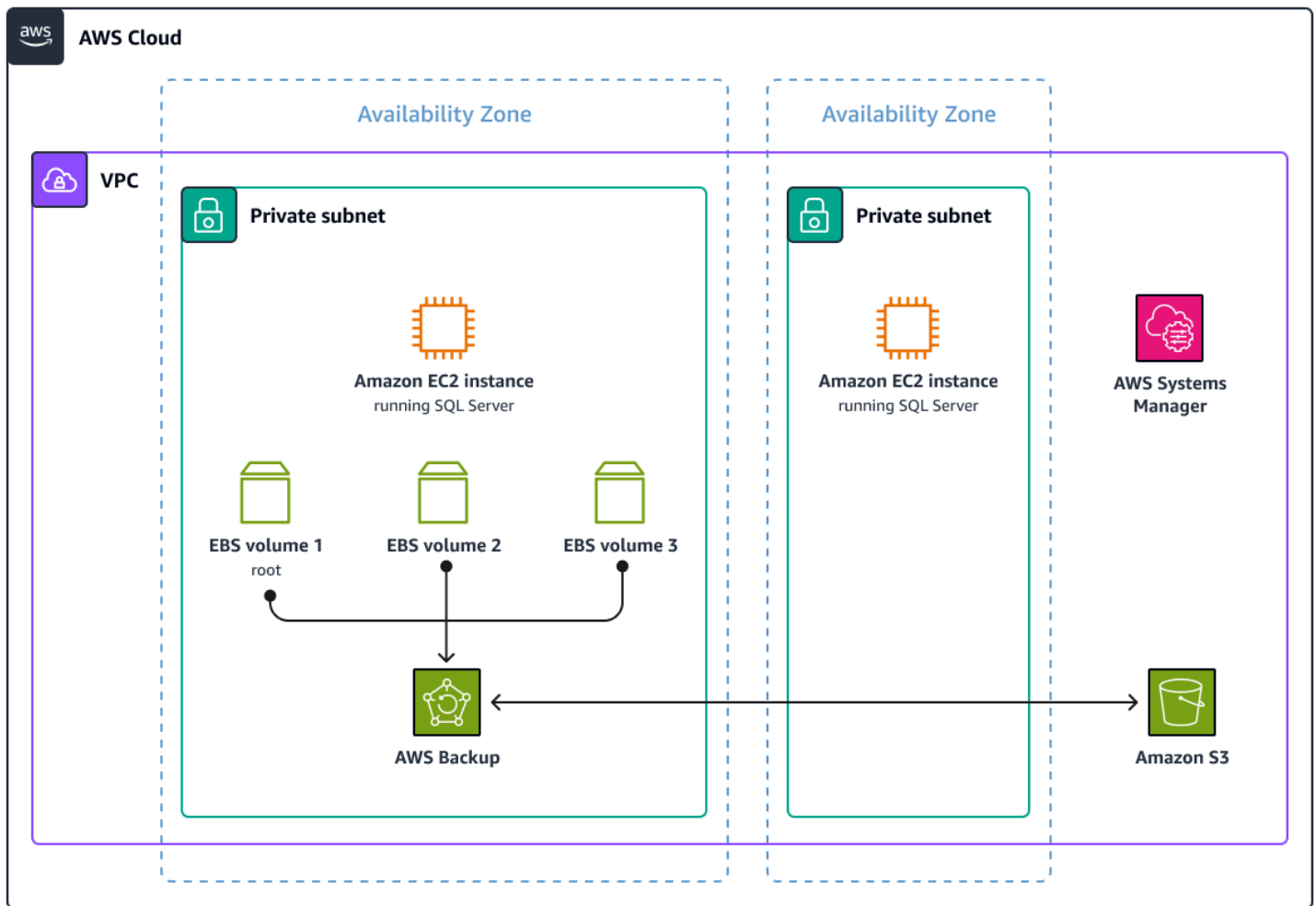
在使用启用 VSS 的快照时，请考虑以下限制：

- 您无法使用 SQL Server 实例的加密快照执行跨区域 point-in-time 恢复。
- 您无法创建未加密实例的加密快照。
- 您无法恢复单个数据库，因为快照是在 EBS 卷级别拍摄的。
- 您无法将实例还原为自身。
- 数据库实例的快照必须使用与数据库实例相同的 AWS Key Management Service (AWS KMS) 密钥进行加密。
- 在快照备份过程中，存储 I/O 会暂停几分之一秒（大约 10 毫秒）。

使用 SQL 服务器备份 AWS Backup

您可以使用[AWS Backup](#)来集中和自动执行数据 AWS 服务保护。AWS Backup 提供了一种经济实惠、完全托管、基于策略的解决方案，可大规模简化数据保护。AWS Backup 还可以帮助您履行监管合规义务并实现业务连续性目标。结合使用 AWS Organizations，AWS Backup 使您能够集中部署数据保护（备份）策略，以配置、管理和治理组织内 AWS 账户 和资源的备份活动。

下图显示了使用在 EC2 上的 SQL Server 的备份和还原解决方案的架构 AWS Backup。



请考虑使用备份 SQL Server 的以下好处 AWS Backup：

- 您可以自动备份计划、保留管理和生命周期管理。
- 您可以将备份策略集中到整个组织中，跨多个账户和。AWS 区域
- 您可以集中监控备份活动并发出警报。AWS 服务
- 您可以为灾难恢复计划实施跨区域备份。
- 该解决方案支持跨账户备份。
- 您可以使用二级备份加密来执行安全备份。
- 所有备份都支持使用加密密钥进行 AWS KMS 加密。
- 该解决方案适用于 TDE。
- 您可以从 AWS Backup 控制台中还原到特定的恢复点。
- 您可以备份整个 SQL Server 实例，其中包括所有 SQL Server 数据库。

数据库级备份

这些方法使用 Microsoft SQL Server 的本机备份功能。您可以对 SQL Server 实例上的单个数据库进行备份，也可以恢复单个数据库。

这些用于本机 SQL Server 备份和恢复的选项还支持以下内容：

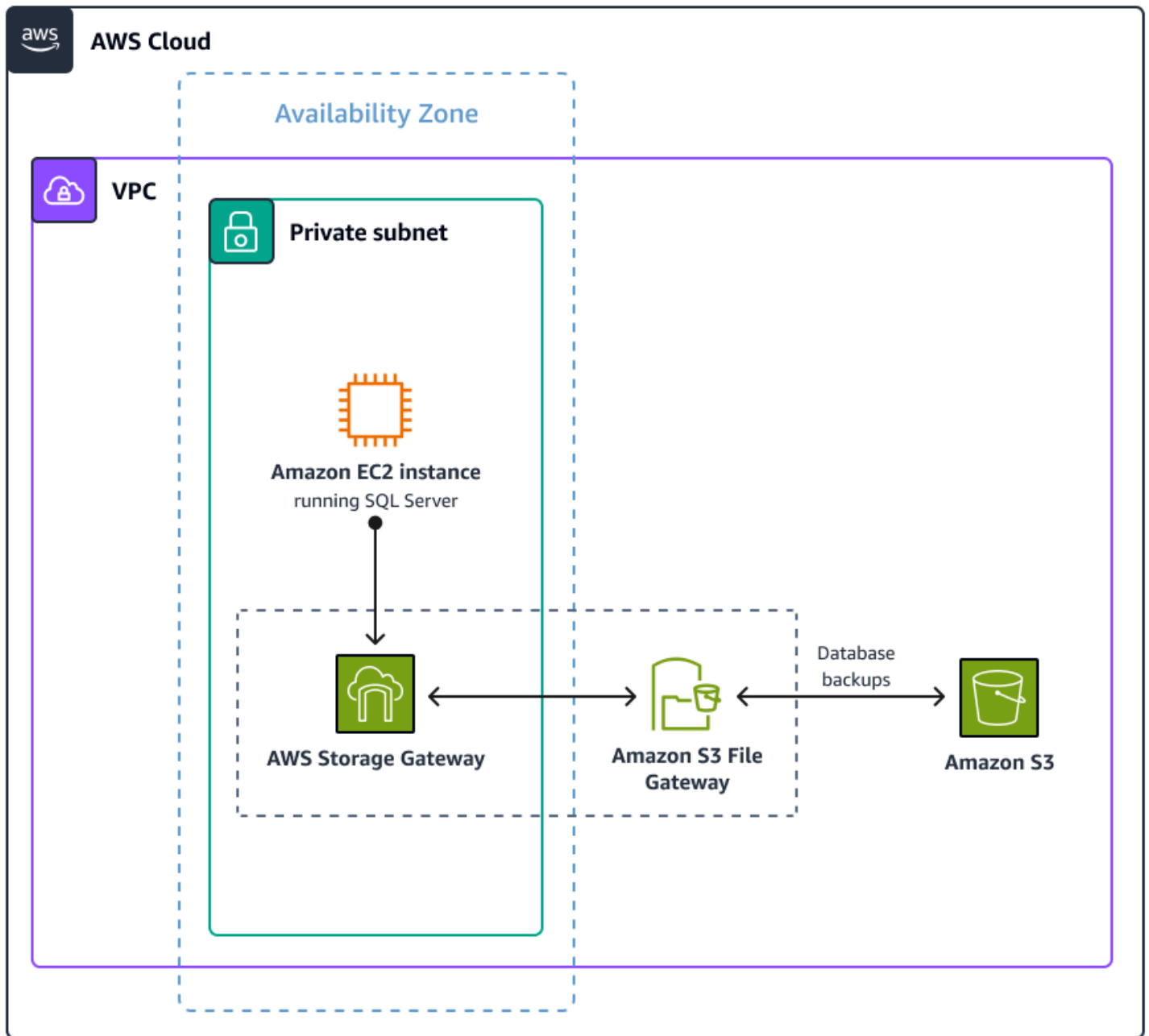
- 压缩备份和多文件备份
- 完整备份、差异备份和 T 日志备份
- TDE 加密的数据库

SQL Server 原生备份和还原到亚马逊 S3

亚马逊 EC2 上的 SQL Server 支持 SQL Server 数据库的本机备份和恢复。您可以对 SQL Server 数据库进行备份，然后将备份文件还原到现有数据库、新的 SQL Server EC2 实例、Amazon RDS for SQL Server 或本地服务器。

Storage Gateway 是一项混合云存储服务，它为本地应用程序提供对几乎无限的云存储的访问权限。您可以使用 Storage Gateway 将微软 SQL Server 数据库直接备份到 Amazon S3，从而减少本地存储占用空间，并使用 Amazon S3 获得持久、可扩展且经济实惠的存储。

下图显示了使用 Storage Gateway 和 Amazon S3 的本机备份和还原解决方案的架构。



考虑在 Storage Gateway 中使用本机 SQL Server 备份有以下好处：

- 您可以将存储网关映射为 EC2 实例上的服务器消息块 (SMB) 文件共享，并将备份发送到 Amazon S3。
- 备份直接存入 S3 存储桶或通过 Storage Gateway 文件缓存。
- 支持多文件备份。

请考虑使用 Storage Gateway 进行本机备份的以下限制：

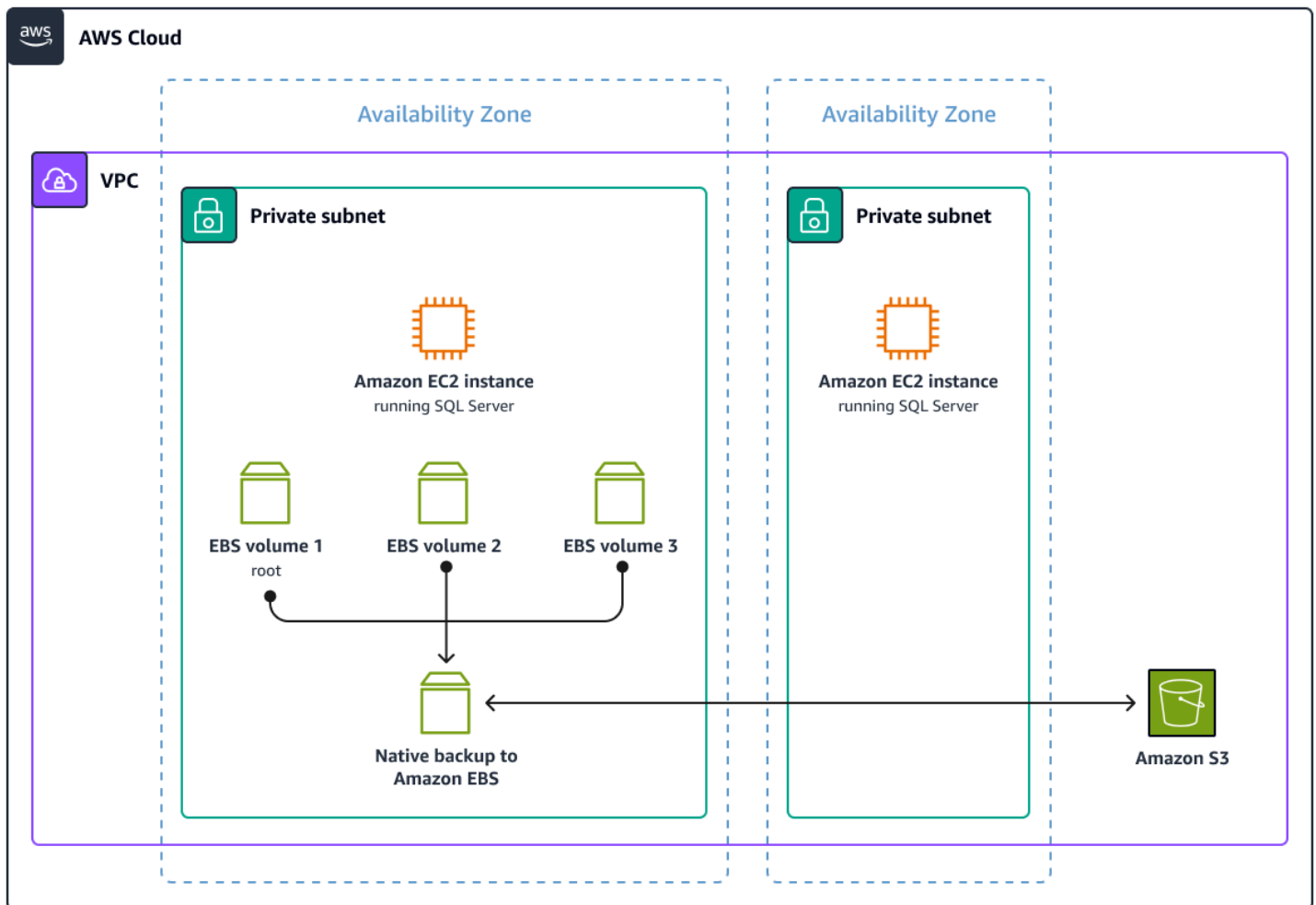
- 必须为每个单独的数据库设置备份和恢复。
- 您必须为备份文件管理 [Amazon S3 生命周期政策](#)。

有关如何设置 [Storage Gateway](#) 的更多信息，请参阅使用 [AWS 博客上的 AWS Storage Gateway](#) 文章将 SQL Server 备份存储在 Amazon S3 中。

SQL Server 本机备份到 EBS 卷

您可以对 SQL Server 数据库进行本机备份，并将该文件存储在 Amazon EBS 卷中。Amazon EBS 是一项高性能的数据块存储服务。EBS 卷具有弹性，支持加密。它们可以分离并连接到 EC2 实例。您可以在 EC2 实例上使用相同的 EBS 卷类型或不同的 EBS 卷类型备份 SQL Server。备份到不同的 EBS 卷的一个优点是可以节省成本。

下图显示了 EBS 卷的本机备份的架构。



请考虑使用 SQL Server 本机备份到 EBS 卷有以下好处：

- 您可以对 SQL Server EC2 实例上的单个数据库进行备份，也可以恢复单个数据库，无需恢复整个实例。
- 支持多文件备份。
- 您可以使用 SQL Server Agent 和 SQL Server 作业引擎安排备份作业。
- 您可以通过选择硬件来实现更高的性能。例如，您可以使用 st1 存储卷来实现更高的吞吐量。

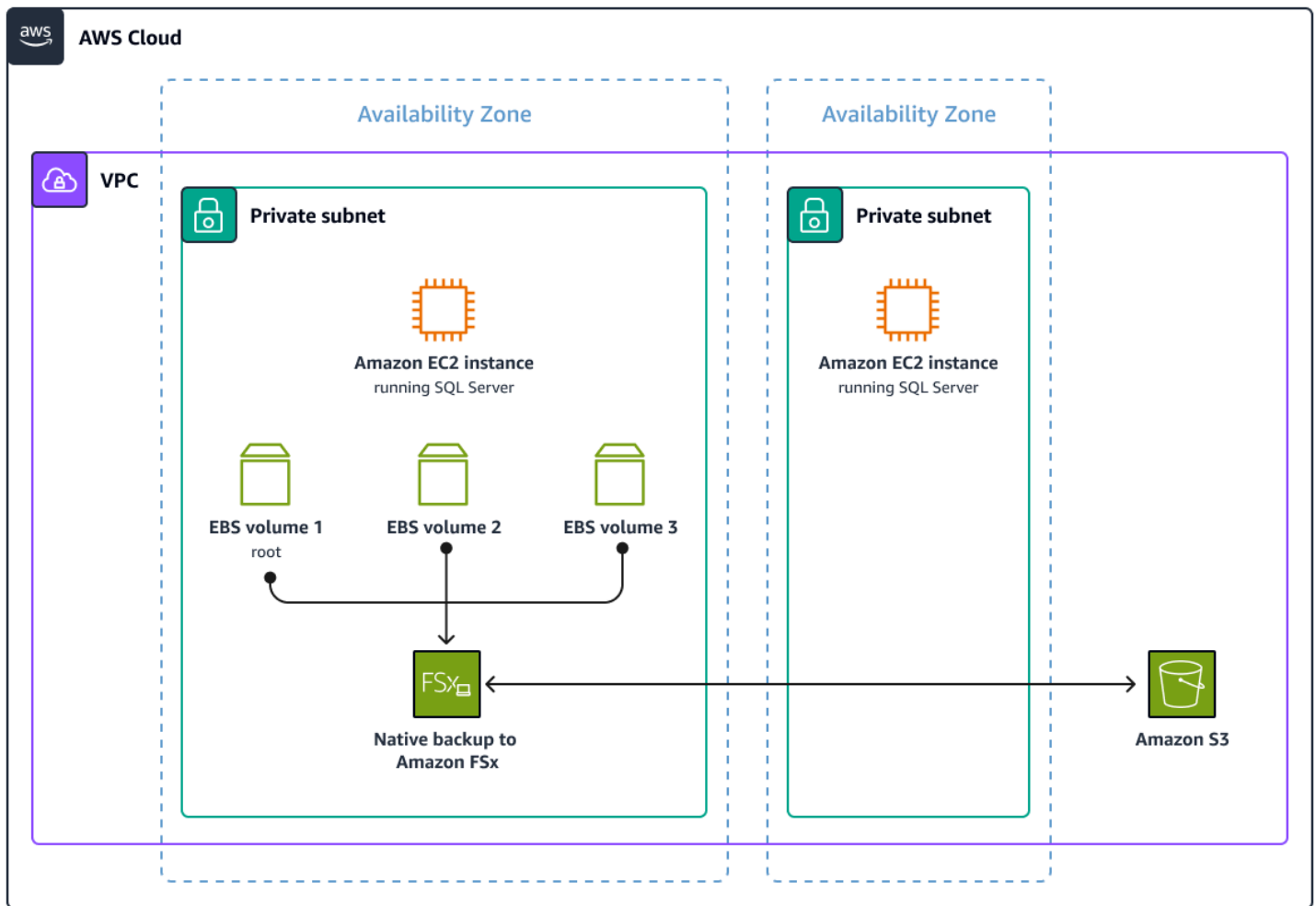
请考虑使用本机备份到 EBS 卷的以下限制：

- 您必须手动将备份从 EBS 卷移至 Amazon S3。
- 对于大型备份，您必须管理 Amazon EC2 上的磁盘空间。
- 在 EC2 实例上，Amazon EBS 吞吐量可能是一个瓶颈。
- 在 Amazon EBS 上存储备份需要额外的存储空间。

SQL Server 本机备份到适用于 Windows File Server 的 Amazon FSx

[适用于 Windows File Server 的 Amazon FSx](#) 是一个完全托管的本机 Windows 文件系统，可提供高达 64 TB 的存储空间，旨在提供快速、可预测和稳定的性能。AWS 在 [FSx for Windows File Server 上引入了对多可用区文件系统部署的本机支持](#)。本机支持使部署 Windows 文件存储变得更加容易，并且 AWS 具有跨多个可用区域的高可用性和冗余性。AWS 还引入了对 [SMB 持续可用 \(CA\) 文件共享](#) 的支持。您可以使用适用于 Windows File Server 的 FSx 作为 SQL Server 数据库的备份存储。

下图显示了适用于 Windows File Server 的 FSx 的本机 SQL Server 备份的架构。



考虑使用适用于 Windows File Server 的本机 SQL Server 备份到 FSx 有以下好处：

- 您可以将 SQL Server 数据库备份到 Amazon FSx 文件共享中。
- 您可以对 SQL Server 实例上的单个数据库进行备份，也可以恢复单个数据库，无需恢复整个实例。
- 支持多部分备份。
- 您可以使用 SQL Server Agent 和作业引擎安排备份作业。
- 与 Amazon EBS 相比，这些实例的网络带宽更高。

使用适用于 Windows File Server 的本机 SQL Server 备份到 FSx 时，请考虑以下限制：

- 您必须使用或 AWS Backup 手动将备份从 Amazon FSx 移至 Amazon S3。AWS DataSync
- 在 Amazon FSx 上，大型备份可能需要额外的磁盘空间管理开销。
- EC2 实例网络吞吐量可能是一个瓶颈。
- 在适用于 Windows File Server 的 FSx 上存储备份需要额外的存储。

适用于 ONTAP 的 SQL Server 备份到亚马逊 FSx NetApp

使用 FSx for ONTAP 的快照始终保持崩溃一致性，但它们需要您暂停（或暂停）数据库的 I/O 才能创建应用程序一致性的快照。您可以将（带有特定应用程序 NetApp SnapCenter（包括 SQL Server）插件的编排工具）与 FSx for ONTAP 一起使用，以创建应用程序一致性快照并保护、复制和克隆数据库，无需支付额外费用。

NetApp SnapCenter

NetApp SnapCenter 是实现应用程序一致性数据保护的统一平台。SnapCenter 将快照称为备份。本指南采用相同的命名惯例。SnapCenter 提供了用于管理应用程序一致性备份、恢复和克隆的单一管理面板。您可以为特定的数据库应用程序添加 SnapCenter 插件来创建应用程序一致性备份。SQL Server SnapCenter 插件提供以下功能，可简化您的数据保护工作流程。

- 完整备份和日志备份的粒度备份和还原选项
- 就地还原并恢复到其他位置

有关更多信息 SnapCenter，请参阅存储博客上的 [“使用 NetApp SnapCenter 适用于 NetApp ONTAP 的 Amazon FSx 保护您的 SQL Server 工作负载”](#) 一文。AWS

备份成本优化

以下选项可以帮助您降低在上存储 SQL Server 备份的成本 AWS。

- 在创建备份文件期间启用 [SQL Server 压缩](#)，并将尽可能小的文件发送到存储器。例如，3:1 的压缩比表示您节省了大约 66% 的磁盘空间。要对这些列进行查询，您可以使用以下 Transact-SQL 语句：`SELECT backup_size/compressed_backup_size FROM msdb..backupset;`
- 对于传送到 S3 存储桶的备份，启用 [Amazon S3 智能分层](#) 存储类可将存储成本降低 30%。
- 对于向 FSx for Windows File Server 或 FSx for ONTAP 的备份，使用单个可用区可节省 50% 的成本（与使用多个可用区相比）。有关定价信息，请参阅适用于 [Windows 文件服务器的亚马逊 FSx](#) 定价和适用于 ONTAP 的 [亚马逊 FSx](#) 定价。NetApp
- SQL Server 2022 最有效的选择是直接备份到亚马逊 S3。您可以避开 Storage Gateway，从而节省额外成本。

备份的基准测试结果

本节根据对本指南中涵盖的备份解决方案进行性能基准测试的结果，从成本和性能的角度比较了示例 1 TB 数据库的以下选项。

- EC2 实例规格 — r5d.8xlarge 搭载 Windows Server 2019 和 SQL Server 2019 开发者版
- 数据库规格-禁用 TDE 时大小为 1 TB

测试是以 r5d.8xlarge 实例和 1 TB 的 SQL Server 数据库作为源进行的。源系统是根据最佳实践配置的，源数据库包含四个数据文件（每个文件为 250 GB）和一个日志文件（50 GB），分布在不同的 gp3 卷上。SQL Server 本机BACKUP命令包括写入 10 个备份文件，使用压缩来优化备份性能，减少通过网络发送和写入目标的数据量。在所有测试案例中，存储性能都是瓶颈。

这些类型的测试几乎有无穷无尽的可能配置。该测试的重点是针对性能、成本、可扩展性和实际用例进行优化。下表显示了为备份目标选项捕获的性能指标。

备份选项	级别	运行时长（大约）	备份速率	每月费用美元*
本机备份到本地 EBS st1 硬盘，2 TB	数据库	00:30:46 分钟	554.7 Mbps	92.16 美元
本机备份到本地 EBS 固态硬盘 gp3，2 TB	数据库	00:22:00 分钟	512 Mbps	193.84 美元
适用于 Windows File Server 的 fsX 本机备份 File Server 硬盘，2 TB @512 Mbps 吞吐量	数据库	00:20:58 分钟	814.0Mbps	1,146 美元
适用于 Windows File Server 固态硬盘的 fsX 本机备份，2 TB @512 Mbps 吞吐量	数据库	00:20:00 分钟	814.0Mbps	1,326 美元
本机备份到带有 2 TB gp3 的	数据库	00:23:20 分钟	731.5Mbps	470.42 美元

备份选项	级别	运行时长 (大约)	备份速率	每月费用美元*
S3 文件网关 m6i.4xlarge (16 个 vCPU , 64 GB)				
EBS VSS 快照	EBS 卷	00:00:02 秒 00:00:53 秒	N/A 快照	51 美元
AWS Backup (AMI 备份)	AMI	00:00:04 秒 00:08:00 分钟	N/A 快照	75 美元
将原生 SQL Server 直接备 份到亚马逊 S3 (SQL Server 2022)	数据库	00:12:00 分钟	731.5Mbps	前 50 TB / 月 , 每 GB 0.023 美元/ 月 23.55 美元
适用于 ONTAP 的 FSx 本机 备份 (使用) SnapCenter	数据库	–	–	440.20 美元

上表假设以下内容：

- 数据传输和 Amazon S3 费用不包括在内。
- 存储价格包含在实例定价中。
- 费用以该us-east-1地区为准。
- 吞吐量和 IOPS 增长了 10%，其中多个备份在一个月内的总体变化率为 10%。

测试结果显示，最快的选择是将本机 SQL Server 数据库备份到 FSx for Windows File Server 的 fsX。备份到 Storage Gateway 和本地连接的 EBS 卷是更具成本效益的选择，但性能较慢。对于服务器级备份 (AMI)，我们建议使用 AWS Backup 以获得最佳性能、成本和可管理性。

成本优化建议

了解在 Amazon EC2 上备份 SQL Server 的可能解决方案是保护数据、确保满足备份需求以及制定从关键事件中恢复的计划的计划的关键。本节探讨的备份和恢复 SQL Server 实例和数据库的不同方法可以帮助您设计保护数据并满足组织要求的备份和还原策略。

本节介绍以下备份选项：

- 压缩
- Amazon S3 Intelligent-Tiering
- 单可用区
- Backup 到 URL

为这些选项中的每一个提供的指导都是高级的。如果您希望在组织中实施这些建议中的任何一项，我们建议您联系您的客户团队。然后，该团队可以与 Microsoft Speciality SA 接触，领导对话。你也可以通过发送电子邮件至 optimize-microsoft@amazon.com 联系我们。

总之，我们提出以下建议：

- 如果你使用的是 SQL Server 2022，那么备份到亚马逊 S3 是最具成本效益的选择。
- 如果你使用的是 SQL Server 2019 和更早的 SQL Server 版本，可以考虑将备份到由 Amazon S3 提供支持的 Storage Gateway 作为最具成本效益的选择。

压缩

压缩的目标是减少每次备份消耗的存储空间，这对各种存储选项都有好处。必须在 SQL Server [实例级别为 SQL Server](#) 备份启用压缩。以下示例说明如何在备份数据库中添加压缩关键字：

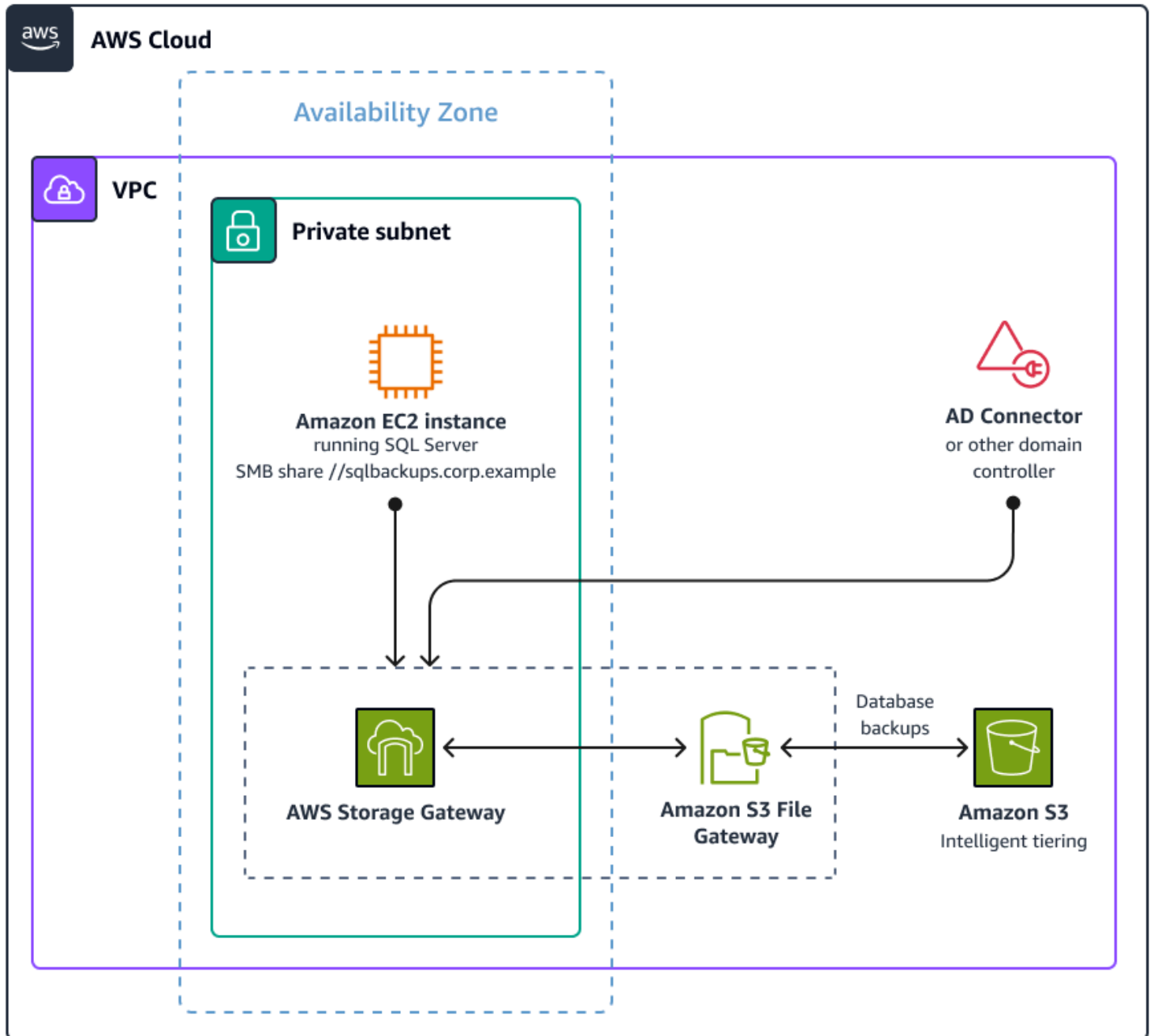
```
BACKUP DATABASE <database_name> TO DISK WITH COMPRESSION (ALGORITHM =
QAT_DEFLATE)
```

Amazon S3 Intelligent-Tiering

[对于传送到 Amazon S3 存储桶的备份，您可以启用 Amazon S3 智能分层作为您的 Amazon S3 文件网关存储类别。](#)这可以将存储成本降低多达 30%。然后，您可以使用可与 [Active Directory 域](#) 集成的 SMB 文件共享将 S3 文件网关挂载到 SQL 服务器。这为您提供了共享的访问控制、利用现有服务账户的能力，以及使用常见的 Microsoft 文件协议访问 Amazon S3。对于可能无法直接连接到域控制器的

帐户，您可以使用 [Active Directory 连接器](#) 来促进与本地或云端的 Active Directory 的通信。要在网关上配置 Active Directory 设置，必须为域控制器指定 Active Directory 连接器 IP，以代理对活动目录的请求。

下图显示了基于 S3 智能分层的解决方案的架构。



默认情况下，写入 S3 存储桶的备份文件使用标准层。要将备份文件从标准层转换为 S3 智能分层，必须 [创建](#) 生命周期规则。您也可以使用启用 S3 智能分层。 [AWS Management Console](#) 有关更多信息，请参阅文档中的 [Amazon S3 智能分层使用入门](#)。AWS

单可用区

要创建单可用区文件系统，请在为 [Windows File Server 文件系统创建 FSX](#) 时选择单可用区选项。Amazon FSx 每天还使用 Windows 卷影复制服务对您的文件系统进行高度持久的备份（存储在 Amazon S3 中），并允许您随时进行其他备份。请记住使用单可用区的一些问题。例如，如果配置文件系统的受影响可用区一次停机数小时，则无法访问 SMB 文件共享。如果您需要访问数据，则必须从源区域内可用区域的备份中恢复数据。有关更多信息，请参阅本指南的“[使用单一可用区](#)”部分。

Backup 到 URL

对于 SQL Server 2022，[网址备份](#)功能允许直接备份到亚马逊 S3。这是运行 SQL Server 2022 的理想备份方法，AWS 因为您可以在存储层获得 Amazon S3 的完整功能集，并消除了先前版本中实现此功能所需的 AWS Storage Gateway 设备成本。实现此功能时需要考虑两个主要成本：数据传输成本和所选的 S3 存储类别。如果您想获得 Amazon S3 的原生灾难恢复功能，则必须考虑到[跨区域复制会产生跨区域数据](#)流出成本。要了解有关如何配置此选项的更多信息，请参阅微软工作负载博客上的“[将 SQL Server 数据库备份到 Amazon S3](#)”的 AWS 文章。

其他资源

- [亚马逊 EC2 上的 SQL Server 的备份和还原选项](#) (AWS 规范性指南)
- [使用 AWS Backup \(AWS 存储博客\) 实现 Amazon RDS 的 Point-in-time 恢复和持续备份](#)
- [使用适用于 NetApp ONTAP 的 Amazon FSx \(NetApp SnapCenter AWS 存储博客\) 来保护你的 SQL Server 工作负载](#)
- [开始使用 Amazon S3 智能分层](#) (AWS 入门资源中心)
- [适用于 SQL Server 的 Amazon RDS 的备份和还原策略](#) (AWS 数据库博客)
- [将@@本地微软 SQL Server 数据库迁移到亚马逊 EC2](#) (AWS 规范性指南)
- [在亚马逊 EC2 上部署微软 SQL Server 的最佳实践](#) (AWS 白皮书)

实现 SQL 服务器数据库的现代化

概述

如果您正在开始对传统数据库进行现代化改造，以实现可扩展性、性能和成本优化，那么在使用像 SQL Server 这样的商用数据库时，您可能会面临诸如 SQL Server 之类的挑战。商业数据库价格昂贵，会锁定客户，并提供惩罚性的许可条款。本节概述了从 SQL Server 向开源数据库迁移和实现现代化的选项，以及有关为工作负载选择最佳选项的信息。

您可以将 SQL Server 数据库重构为开源数据库，例如亚马逊 Aurora PostgreSQL，以节省 Windows 和 SQL Server 的许可成本。像 Aurora 这样的云原生现代数据库将开源数据库的灵活性和低成本与商业数据库的强大企业级功能融为一体。如果您有可变的工作负载或多租户工作负载，也可以迁移到 [Aurora 无服务器 V2](#)。这可以将成本降低多达 90%，具体取决于工作负载特征。[此外，还 AWS 提供诸如适用于 Aurora PostgreSQL 的 Babelfish、\(\) 之类的工具以及诸如 AWS Schema Conversion Tool \(AWS SCT\) 之类的服务，用于简化 SQL Server 数据库的迁移和现代化。](#) [AWS Database Migration Service](#) [AWS DMS](#) [AWS](#)

数据库产品

从 Windows 上的 SQL Server 迁移到开源数据库，例如亚马逊 Aurora、适用于 MySQL 的 Amazon RDS 或适用于 PostgreSQL 的 Amazon RDS，可以在不影响性能或功能的情况下节省大量成本。请考虑以下事项：

- 从亚马逊 EC2 上的 SQL Server Enterprise 版切换到适用于 PostgreSQL 的亚马逊 RDS 或适用于 MySQL 的亚马逊 RDS 可以节省高达 80% 的成本。
- 从亚马逊 EC2 上的 SQL Server 企业版切换到兼容亚马逊 Aurora PostgreSQL 的版本或兼容亚马逊 Aurora MySQL 的版本可以节省高达 70% 的成本。

对于传统的数据库工作负载，适用于 PostgreSQL 的 Amazon RDS 和适用于 MySQL 的 Amazon RDS 可以满足需求并为关系数据库提供经济实惠的解决方案。Aurora 增加了许多以前仅限于昂贵的商业供应商的可用性和性能功能。Aurora 中的弹性功能需要额外付费。但是，与其他商业供应商的类似功能相比，Aurora 的弹性成本仍然比商业软件为同类功能收取的费用便宜。与标准的 MySQL 和 PostgreSQL 部署相比，Aurora 架构经过优化，可显著提高性能。

由于 Aurora 与开源 PostgreSQL 和 MySQL 数据库兼容，因此还有一个额外的好处，那就是便携性。无论最佳选择是适用于 PostgreSQL 的亚马逊 RDS、适用于 MySQL 的 Amazon RDS 还是 Aurora，都取决于了解业务需求并将必要的功能映射到最佳选择。

亚马逊 RDS 和 Aurora 的比较

下表汇总了亚马逊 RDS 和 Amazon Aurora 之间的主要区别。

类别	适用于 PostgreSQL 的亚马逊 RDS 或适用于 MySQL 的亚马逊 RDS	Aurora PostgreSQL 或 Aurora MySQL
Performance	表现不错	3 倍或更高的性能

类别	适用于 PostgreSQL 的亚马逊 RDS 或适用于 MySQL 的亚马逊 RDS	Aurora PostgreSQL 或 Aurora MySQL
故障转移	通常为 60—120 秒*	通常为 30 秒
可扩展性	最多 5 个只读副本 以秒为单位延迟	最多 15 个只读副本 延迟 (以毫秒为单位)
存储	高达 64 TB	高达 128 TB
存储 HA	多可用区，一个或两个备用区域，每个备用区域都有数据库副本	默认情况下，在 3 个可用区中有 6 个数据副本
备份	每日快照和日志备份	持续、异步备份到 Amazon S3
Aurora 的创新	NA	100 GB 快速克隆数据库
	自动缩放只读副本	
	查询计划管理	
	Aurora Serverless	
	带有全球数据库的跨区域副本	
	集群缓存管理**	
	并行查询	
	数据库活动流	

*大型事务可能会增加故障转移时间

**在 Aurora PostgreSQL 中可用

下表显示了本节所涵盖的不同数据库服务的每月估计成本。

数据库服务	每月费用美元*	AWS Pricing Calculator (需要 AWS 账户)
适用于 SQL Server 的亚马逊 RDS 企业版	3,750 美元	估计
适用于 SQL Server 的亚马逊 RDS 标准版	2,318 美元	估计
亚马逊 EC2 上的 SQL Server 企业版	2,835 美元	估计
亚马逊 EC2 上的 SQL Server 标准版	1,345 美元	估计
Amazon RDS for PostgreSQL	742 美元	估计
Amazon RDS for MySQL	712 美元	估计
Aurora PostgreSQL	1,032 美元	估计
Aurora MySQL	1,031 美元	估计

* 存储价格已包含在实例定价中。费用根据 us-east-1 地区而定。吞吐量和 IOPS 是假设值。计算是针对 r6i.2xlarge 和 r6g.2xlarge 实例的。

成本优化建议

异构数据库迁移通常需要将数据库架构从源数据库引擎转换为目标数据库引擎，并将数据从源数据库迁移到目标数据库。迁移的第一步是评估 SQL Server 架构和代码对象并将其转换为目标数据库引擎。

你可以使用 [AWS Schema Conversion Tool \(AWS SCT\)](#) 来评估和评估数据库是否与各种目标开源数据库选项兼容，例如适用于 MySQL 的 Amazon RDS 或适用于 PostgreSQL 的 Amazon RDS、Aurora MySQL 和 PostgreSQL。你也可以使用 Babelfish Compass 工具来评估与 Babelfish for Aurora PostgreSQL 的兼容性。这使得 AWS SCT 和 Compass 成为了在决定迁移策略之前了解所涉及的前期工作的强大工具。如果您决定继续，则 AWS SCT 会自动对架构进行所需的更改。Babelfish Compass 背后的核心理念是允许 SQL 数据库在不进行任何修改或很少修改的情况下迁移到 Aurora。Compass 将评估现有的 SQL 数据库，以确定能否实现这一目标。这样，在将数据从 SQL Server 迁移到 Aurora 之前，就可以知道结果。

AWS SCT 自动将数据库架构和代码转换和迁移到目标数据库引擎。您可以使用 Babelfish for Aurora PostgreSQL 将您的数据库和应用程序从 SQL Server 迁移到 Aurora PostgreSQL，无需进行任何架构更改或进行最少的架构更改。这可以加快您的迁移速度。

架构迁移完成后，您可以使用迁 AWS DMS 移数据。AWS DMS 可以执行完整数据加载和复制更改，以便在最短的停机时间内执行迁移。

本节更详细地探讨了以下工具：

- AWS Schema Conversion Tool
- 适用于 Aurora PostgreSQL 的 Babelfish
- Babelfish 指南针
- AWS Database Migration Service

AWS Schema Conversion Tool

您可以使用 AWS SCT 评估现有的 SQL Server 数据库并评估与 Amazon RDS 或 Aurora 的兼容性。为了简化迁移过程，您还可以在异构数据库迁 AWS SCT 移中使用将架构从一个数据库引擎转换为另一个数据库引擎。您可以使用评估您的应用程序 AWS SCT，并转换编写 C#、C++、Java 和其他语言的应用程序的嵌入式应用程序代码。有关更多信息，请参阅 AWS SCT 文档 AWS SCT 中的 [使用转换应用程序 SQL](#)。

AWS SCT 是一款支持许多数据库源的免费 AWS 工具。要使用 AWS SCT，请将其指向源数据库，然后进行评估。然后，[AWS SCT](#) 评估架构并生成评估报告。评估报告包括执行摘要、复杂性和迁移工作、合适的目标数据库引擎以及转换建议。要下载 AWS SCT，请参阅 AWS SCT 文档 AWS SCT 中的 [安装、验证和更新](#)。

下表显示了由生成的执行摘要示例 AWS SCT，用于显示将数据库更改为不同目标平台所涉及的复杂性。

目标平台	自动更改或最小更改			复杂的动作			
	存储对象	代码对象	转化操作	存储对象		代码对象	
Amazon RDS for MySQL	60 (98%)	8 (35%)	42	1 (2%)	1	15 (65%)	56

Amazon Aurora MySQL 兼容版	60 (98%)	8 (35%)	42	1 (2%)	1	15 (65%)	56
Amazon RDS for PostgreSQL	60 (98%)	12 (52%)	54	1 (2%)	1	11 (48%)	26
Amazon Aurora PostgreSQL 兼容版	60 (98%)	12 (52%)	54	1 (2%)	1	11 (48%)	26
Amazon RDS for MariaDB	60 (98%)	7 (30%)	42	1 (2%)	1	16 (70%)	58
Amazon Redshift	61 (100%)	9 (39%)	124	0 (0%)	0	14 (61%)	25
AWS Glue	0 (0%)	17 (100%)	0	0 (0%)	0	0 (0%)	0
Babelfish	59 (97%)	10 (45%)	20	2 (3%)	2	12 (55%)	30

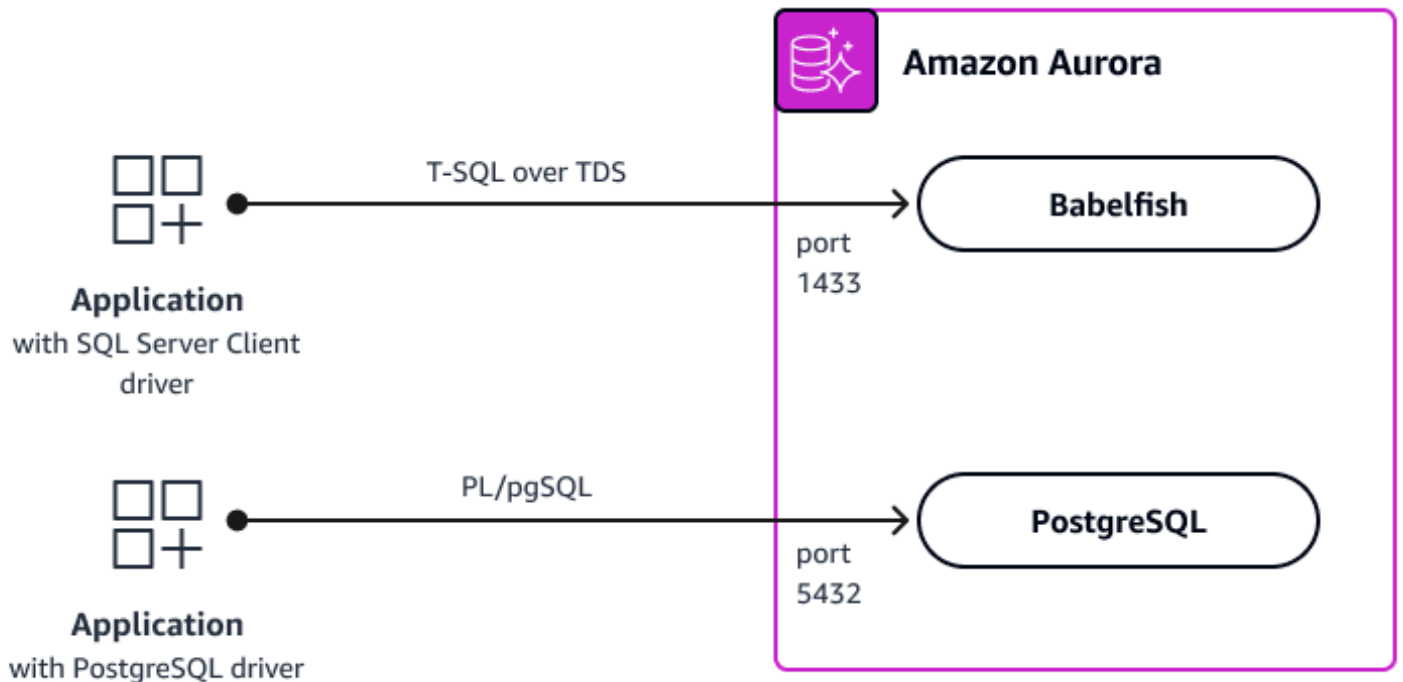
AWS SCT 报告还提供有关无法自动转换的架构元素的详细信息。您可以参考[AWS 迁移行动手册](#)来缩小 AWS SCT 转化差距并优化目标架构。有许多数据库迁移行动手册可以帮助进行异构迁移。

适用于 Aurora PostgreSQL 的 Babelfish

适用于 Aurora 的 Babelfish PostgreSQL 扩展了 Aurora PostgreSQL，使其能够接受来自 SQL Server 客户端的数据库连接。Babelfish 使最初为 SQL Server 构建的应用程序可以直接与 Aurora PostgreSQL 配合使用，只需很少的代码更改，也无需更改数据库驱动程序。Babelfish 将 Aurora PostgreSQL 变成了双语，这样 Aurora PostgreSQL 就可以同时使用 T-SQL 和 PL/pgSQL 语言。Babelfish 最大限度地减少了从 SQL Server 迁移到 Aurora PostgreSQL 的工作量。这加快了迁

移速度，最大限度地降低了风险，并显著降低了迁移成本。您可以在迁移后继续使用 T-SQL，但也可以选择使用 [PostgreSQL 原生工具进行开发](#)。

下图说明了使用 T-SQL 的应用程序如何连接到 SQL Server 中的默认端口 1433 并使用 Babelfish 转换器与 Aurora PostgreSQL 数据库通信，而使用 PL/pgSQL 的应用程序如何使用 Aurora PostgreSQL 中的默认端口 5432 直接同时连接到 Aurora PostgreSQL 数据库。



Babelfish 不支持某些 SQL Server T-SQL 功能。因此，Amazon 提供了评估工具，用于 line-by-line 分析您的 SQL 语句，并确定 Babelfish 是否不支持其中任何语句。

Babelfish 评估有两种选择。AWS SCT 可以评估你的 SQL Server 数据库与 Babelfish 的兼容性。另一种选择是 Babelfish Compass 工具，这是推荐的解决方案，因为指南针工具已根据新版本的 Aurora PostgreSQL 版 Babelfish 进行了更新。

Babelfish 指南针

[Babelfish Compass](#) 是一款可免费下载的工具，与最新发布的 Aurora PostgreSQL 版 Babelfish 保持一致。相比之下，AWS SCT 将在一段时间后支持更新的 Babelfish 版本。[Babelfish Compass](#) 是针对 SQL Server 数据库架构运行的。您还可以使用 SQL Server 管理工作室 (SSMS) 等工具提取源 SQL Server 数据库架构。然后，您可以通过 Babelfish Compass 运行架构。这将生成报告，详细说明 SQL Server 架构与 Babelfish 的兼容性，以及在迁移之前是否需要进行任何更改。Babelfish Compass 工具还可以自动执行其中的许多更改，最终加快迁移速度。

评估和更改完成后，您可以使用 SSMS 或 sqlcmd 等 SQL Server 原生工具将架构迁移到 Aurora PostgreSQL。有关说明，请参阅 AWS 数据库博客上的[“使用 Babelfish 从 SQL Server 迁移到 Amazon Aurora”](#)一文。

AWS Database Migration Service

迁移架构后，您可以使用 AWS Database Migration Service (AWS DMS) 在停机时间最短的情况下将数据迁移到。AWS DMS 不仅可以加载完整数据，还可以在源系统启动并运行时将更改从源系统复制到目标。源数据库和目标数据库同步后，可以在应用程序指向完成迁移的目标数据库时进行直接转换活动。AWS DMS 目前仅使用 Babelfish 对 Aurora PostgreSQL 目标执行完整数据加载，不会复制更改。有关更多信息，请参阅文档 AWS Database Migration Service 中的[“使用 Babelfish 作为目标”](#) AWS DMS。

AWS DMS 可以进行同构（跨同一个数据库引擎）和异构（跨不同的数据库引擎）迁移。AWS DMS 支持许多源数据库和目标数据库引擎。有关更多信息，请参阅数据库博客中的[使用 AWS DMS 文章将您的 SQL Server 数据库迁移到 Amazon RDS for SQL Server](#)。AWS

其他资源

- [再见微软 SQL Server，你好 Babelfish](#) (AWS 新闻博客)
- [使用 AWS Schema Conversion Tool CLI](#) (数据库博客) 转换 AWS 数据库架构和应用程序 SQL
- [使用最佳实践和现场 AWS 经验教训将 SQL Server 迁移到 Amazon Aurora PostgreSQL](#) (数据库博客)
- [在@@ 从微软 SQL Server 迁移到亚马逊 RDS for PostgreSQL 和亚马逊 Aurora PostgreSQL](#) (数据库博客) AWS 后验证数据库对象

优化 SQL 服务器的存储

概述

本节重点介绍适用于 EC2 工作负载的 SQL Server 的亚马逊弹性块存储 (Amazon EBS) Elastic Block Store 固态硬盘存储的成本优化。

您可以使用多种存储选项来部署和运行 SQL Server 工作负载 AWS。应根据用途、架构、耐久性、性能、容量和成本来选择合适的存储。AWS 运行 SQL Server 工作负载的客户通常会组合使用亚马逊 EBS、NVMe、亚马逊 FSX 和亚马逊简单存储服务 (Amazon S3) Simple Service 存储。

Amazon EBS 是连接到 EC2 计算实例的网络连接存储，用于存储和处理一般操作系统、应用程序、数据库和备份文件。亚马逊 EBS 固态硬盘 (SSD) 存储包括通用固态硬盘 (gp2 和 gp3) 和预配置 IOPS 固态硬盘 (io1、io2 和 io2bx)。请考虑以下事项：

- 某些 EC2 实例 (例如 r5d) 将本地 NVMe 固态硬盘物理连接到主机实例。这些卷提供块级存储，通常用于 SQL Server tempdb 或缓冲池扩展。
- 适用于 Windows File Server 的 Amazon FSx 是一项完全托管的文件存储服务，而 NetApp 适用于 ONTAP 的 Amazon FSx 是 NetApp 基于广受欢迎的 ONTAP 文件系统构建的完全托管的共享存储。Amazon FSx 经常用于在高可用性 SQL Server 故障转移集群实例 (FCI) 配置中运行 SQL Server 工作负载。该解决方案托管 SQL Server 数据和日志文件，从而降低了 EC2 实例上的 EBS 性能要求。
- Amazon S3 是一项对象存储服务，可提供业界领先的可扩展性、数据可用性、安全性和性能。您可以在 Amazon S3 上存储 SQL Server 原生备份文件、AMI、EBS 快照、应用程序日志等。

Amazon EBS 的固态硬盘存储类型、性能和成本

随着耐用性和性能的提高，Amazon EBS 的固态硬盘存储成本通常会增加。存储目前有五种卷类型，每种都有[自己独特的性能指标](#)。有关支持 SSD 的卷的用例和特征的摘要，请参阅 Amazon EBS 文档[固态硬盘 \(SSD\) 卷](#)部分的表格。

您可以使用 Amazon CloudWatch 监控固态硬盘性能，捕获趋势数据，并在达到特定阈值时设置警报。如果您正在运行 SQL Server 工作负载 AWS，请考虑启用[详细监控](#)并部署[CloudWatch 自定义指标](#)以捕获详细的卷性能指标，例如磁盘延迟、IOPS、吞吐量、磁盘队列长度、已用容量与可用容量等。您可以使用这些 CloudWatch 性能指标来识别配置不足和过度配置的存储，并提供历史数据点以准确定义存储需求。

Amazon EBS 的固态硬盘存储成本也因分配的容量而异。下表显示了不同卷类型的比较。所有卷类型都有 1 TB 的容量和类似的性能配置。

卷类型	最大 IOPS (16 KiB I/O)	最大吞吐量 (128 KiB 输入/输出)	每 1TB 的价格	成本节省百分比
gp2	3000	250	102.40 美元	
gp3	3000	250	86.92 美元	15%
io1	16000	500	1,168 美元	

卷类型	最大 IOPS (16 KiB I/O)	最大吞吐量 (128 KiB 输入/输出)	每 1TB 的价格	成本节省百分比
io2	16000	500	1,168 美元	
gp3	16000	500	146.92 美元	87%
io2bx	16000	4,000	1,168 美元	
gp3	16000	1000	181.92 美元	84%

Note

上表中的性能和成本指标是按卷计算的，基于的[估计](#) AWS Pricing Calculator。需要 AWS 账户使用才能访问中的估算值 AWS Pricing Calculator。

Amazon EBS 固态硬盘 gp3 卷以低成本提供卓越的性能。对于需要低于 16,000 IOPS 和 500 吞吐量的工作负载，如果您选择 gp3 卷而不是 io1 或 io2 卷，则最多可以节省 87%。MiBps

与普通 io2 卷相比，io2 Block Express (io2bx) 卷的性能更高。在 16,000 IOPS 时，io1 或 io2 卷只能达到 500 个 MiBps 吞吐量，而 io2bx 卷最多可以配置为 4,000 个吞吐量。MiBps 与 io1 和 io2 卷相比，io2bx 卷以完全相同的价格，在 16,000 到 64,000 IOPS 之间提供的吞吐量是其四倍多。通过将常规 io2 卷附加到 io2bx 支持的 EC2 实例，可以将其转换为 io2bx 卷。有关支持 IO2BX 的 EC2 实例的列表，请参阅 Amazon EBS 文档中的[预配置 IOPS 固态硬盘卷](#)。在部署新存储之前，您可以使用[AWS Pricing Calculator](#)来估算每月成本，并根据耐久性、性能和容量之间的权衡来了解对成本的影响。

亚马逊 EBS 的通用固态硬盘成本优化

我们建议您评估要存储的内容，并确保使用正确的存储类型和类别。例如，Amazon S3 提供了不错的价格、内置的生命周期策略和非常适合 SQL Server 备份的复制选项。SQL Server 2022 能够直接备份到亚马逊 S3，而以前版本的 SQL Server 则依赖本地本地本地备份。如果你运行的是旧版本的 SQL Server，可以考虑备份到亚马逊 EBS 硬盘卷，然后将备份复制到亚马逊 S3。与使用 gp3 卷进行备份相比，此解决方案可以节省 53% 的费用。

下表显示了亚马逊 EBS gp3、亚马逊 EBS HDD st1 和亚马逊 S3 上 1 TB 存储空间的价格差异。

存储类型	容量	下午的价格
EBS gp3 500 MiBps	1TB	96.92 美元
EBS st1 burst 500 MiBps		46.08 美元
S3 标准		23.55 美元
S3 标准 (不经常访问)		12.80 美元
S3 Glacier Deep Archive		1.03 美元

Note

上表中的成本指标基于中的[估计值 AWS Pricing Calculator](#)。需要 AWS 账户 使用才能访问中的估算值 [AWS Pricing Calculator](#)。

我们建议您考虑以下几点：

- 启用详细监控并部署 CloudWatch 自定义指标，以准确捕获其存储性能需求。
- 将 Amazon EBS 存储从 gp2 升级到 gp3，以降低成本、提高灵活性并提高性能。
- 将 Amazon EBS 存储从 io1 升级到 io2，以提高耐用性和性能灵活性。
- 尽可能使用 io2bx 代替 io1 或 io2，以提高耐用性和性能。
- 在选择存储器时，请考虑一种有助于降低容量需求和高性能卷成本 mix-and-match 的方法。例如，您可以将低成本的 gp3 卷用于根卷（操作系统）、SQL Server 安装、系统数据库（不包括 tempdb）和性能较低的用户数据库。这可能有助于降低 io2 卷的容量和成本，该卷可以专用于高性能用户数据库。
- 如果您在上托管 SQL Server 数据库 AWS，我们建议您在每个数据库中使用多个 SQL Server 数据文件。这允许将读/写工作负载分配到多个卷，从而降低每个卷的性能和容量需求，从而降低成本。
- 即使生产工作负载需要更高性能的存储（例如 io1 或 io2/io2bx），也可考虑将非生产工作负载的 gp3 卷用于帮助降低成本。
- 跟踪和预测一段时间内的存储利用率，以轻松识别使用量高峰和意外成本。
- [AWS Compute Optimizer](#) 用于获取有关根据实际利用率向上或向下扩展 EBS 卷的建议。

- 使用的弹性 AWS 来调整 Amazon EBS 固态硬盘卷的性能和容量需求。与本地环境不同，您无需为未来的工作负载过度配置存储性能和容量。您可以将现有的 SQL Server 工作负载迁移到上 AWS 并根据需要调整性能或容量，同时保持数据库在线。

其他资源

- [亚马逊 EBS 卷类型](#) ([亚马逊 EBS 文档](#))
- [亚马逊 Elastic Block Store \(亚马逊 EBS\) Block Store](#) ([亚马逊 EBS 文档](#))
- [预配置 IOPS 固态硬盘卷](#) ([亚马逊 EBS 文档](#))
- [固态硬盘实例存储卷](#) ([亚马逊 EC2 文档](#))
- [亚马逊 EBS 的亚马逊 CloudWatch 指标](#) ([亚马逊 EBS 文档](#))
- [亚马逊 EC2 存储优化型实例的规格](#) ([亚马逊 EC2 文档](#))
- [使用适用于 NetApp ONTAP 的 Amazon FSx \(NetApp SnapCenter AWS 存储博客 \) 来保护你的 SQL Server 工作负载](#)
- [亚马逊 EC2 常见问题解答](#) ([AWS 产品页面](#))

使用 Compute Optimizer 优化 SQL Server 许可

有关如何使用优化 SQL Server 许可证的指南 [AWS Compute Optimizer](#)。

概述

[AWS Compute Optimizer](#) 可以推荐亚马逊弹性计算云 (Amazon EC2) 上微软 SQL Server 工作负载的许可优化机会。Compute Optimizer 可以提供自动推荐以降低许可成本。Compute Optimizer 的建议列在每个拥有微软 SQL Server 许可证的 EC2 实例旁边。提供的信息包括建议节省机会、EC2 实例按需价格和每小时自带许可 (BYOL) 价格。这些信息可以帮助您决定是否应该降级许可证版本。

Compute Optimizer 会根据推断的工作负载类型自动发现您在 Amazon EC2 上的 SQL Server 实例。要查看许可建议，您可以在 Compute Optimizer 中选择 SQL Server 实例，然后使用您的只读数据库凭证通过 [Amazon App CloudWatch lication Insights](#) 进行身份验证。Compute Optimizer 会分析您是否在使用任何 SQL Server 企业版功能。如果未使用企业版功能，Compute Optimizer 建议您降级到标准版以降低许可成本。

您还可以使用 Compute Optimizer 为运行 SQL Server 工作负载的 Amazon EC2 实例提出大小建议。有关更多信息，请参阅本指南中的 [使用 Compute Optimizer 优化 SQL Server 的大小](#)。

成本优化建议

Compute Optimizer 中的许可建议可以帮助你评估你在 Microsoft SQL Server 中使用的功能，并为你的工作负载选择最具成本效益的版本。SQL Server 企业版比标准版贵得多。有关更多信息，请参阅本指南中的[比较 SQL Server 版本](#)和微软网站上的[SQL Server 2022 定价](#)。花时间配置 Compute Optimizer 来评估你的 SQL Server 队列并提供建议可以大大降低许可成本。

许可证详细信息页面提供以下信息：

- 使用下表将您当前的许可设置（例如版本、型号和实例核心数）与 Compute Optimizer 建议进行比较。
- 使用利用率图表查看分析期间使用的企业版功能数量。

有关更多信息，请参阅 Compute Optimizer [文档中的查看商业软件许可建议的详细信息](#)。

配置 Compute Optimizer

Compute Optimizer 使用该 `mssql_enterprise_features_used` 指标分析商业软件许可证。有关该指标的更多信息，请参阅[商用软件许可证的指标](#)。

1. 请确保您拥有相应的权限来选择加入 Compute Optimizer。有关更多信息，请参阅下列内容：
 - [选择加入 Compute Optimizer 的政策](#)
 - [授予独立版 Compute Optimizer 访问权限的策略 AWS 账户](#)
 - [授予组织管理账户访问 Compute Optimizer 访问权限的策略](#)
2. 附加 App CloudWatch Location Insights 所需的实例角色和策略。有关说明，请参阅[启用商用软件许可证推荐的策略](#)。
3. 使用你的 Microsoft SQL Server 数据库凭据启用 CloudWatch 应用程序见解。有关说明，请参阅[Amazon CloudWatch 应用程序见解入门](#)。

Note

要生成商用软件许可证推荐，至少需要连续 30 小时的 CloudWatch 指标数据。有关更多信息，请参阅[CloudWatch 指标要求](#)。

4. 使用以下 SQL 查询配置 App CloudWatch Location Insights 的最低权限访问权限。

```
GRANT VIEW SERVER STATE TO [LOGIN];
```

```
GRANT VIEW ANY DEFINITION TO [LOGIN];
```

这将启用一项新服务 PrometheusSqlExporter SQL。

5. 从目标 AWS 账户 或组织管理账户中，选择加入 Compute Optimizer。有关说明，请参阅[选择加入您的账户](#)。

Note

选择加入后，最长可能需要 24 小时才能生成结果和优化建议。

6. 在 [Compute Optimizer 控制台](#) 中，选择导航窗格中的许可证。
7. 在“调查结果”列中，搜索所有发现的指标不足的实例。如果 Compute Optimizer 检测到 CloudWatch 应用程序见解未启用或权限不足，则会返回此结果。有关更多信息，请参阅[查找原因](#)。要解决这些问题，请执行以下操作：
 - a. 选择实例。
 - b. 添加一个秘密。
 - c. 确认已附加实例角色和策略。
 - d. 选择“启用许可证推荐”。
8. 在“查找结果”列中，搜索查找结果为“未优化”的所有实例。如果 Compute Optimizer 检测到你的亚马逊 EC2 基础设施没有使用你付费购买的任何微软 SQL Server 许可功能，它就会返回这一发现。有关更多信息，请参阅[查找原因](#)。要解决这些问题，请执行以下操作：
 - a. 选择实例。
 - b. 将当前许可证版本与推荐版本进行比较。
 - c. 查看当前的许可证使用率图。
 - d. 如果要降级许可证，请选择实施建议。
 - e. 查看要求并按照说明降级许可证。如果要自动化该过程，请参阅[使用 AWS Systems Manager 文档降级 SQL Server 企业版以降低成本](#)（AWS 博客）。

其他资源

- 通过 AWS Compute Optimizer (AWS 博客) [降低微软 SQL Server 许可成本](#)
- [什么是 AWS Compute Optimizer ?](#) (AWS 文档)
- [查看商用软件许可建议](#) (AWS 文档)
- [降级你的微软 SQL Server 版本](#) (AWS 文档)

- [微软 SQL Server 开启 AWS \(AWS\)](#)
- [微软在 AWS\(AWS\) 上许可](#)
- [微软 SQL Server 2019 定价 \(微软\)](#)
- [微软 SQL Server 2022 定价 \(微软\)](#)

使用 Compute Optimizer 优化 SQL Server 的大小

概述

[AWS Compute Optimizer](#) 帮助数据库管理员 (DBA) 发现亚马逊弹性计算云 (Amazon EC2) 上的微软 SQL Server 工作负载和适当大小的 EC2 实例，从而将许可成本降低多达 25%。Compute Optimizer 中的 [推断工作负载类型](#) 功能使用机器学习 (ML)，并自动检测可能在您的资源上 AWS 运行的应用程序。Compute Optimizer 支持将 SQL Server 作为推断的工作负载类型。通过使用推断的工作负载类型功能，您可以根据在 Amazon EC2 实例上运行的特定工作负载查明节省成本的机会。

使用此功能，您可以按支持的推断工作负载类型（例如 SQL Server）对节省成本的机会进行分类。Compute Optimizer 可以自动发现过度配置的 SQL Server EC2 实例。您可以切换到 EC2 控制台来缩小实例规模，这有助于降低许可和基础设施成本。

您也可以使用 Compute Optimizer 提出 SQL Server 许可建议。有关更多信息，请参阅本指南中的 [使用 Compute Optimizer 优化 SQL Server 许可](#)。

配置 Compute Optimizer

有关将 Compute Optimizer 用于 SQL Server 推断的工作负载的说明，请参阅 [优化性能和降低许可成本：利用 Amazon AWS Compute Optimizer EC2 SQL Server 实例](#) (AWS 博客)。您可以选择加入独立账户、组织成员账户和组织管理账户。对于独立账户和成员账户，选择加入仅为该账户启用 Compute Optimizer。对于组织管理账户，你可以选择是仅在该账户中启用 Compute Optimizer，还是为该组织的所有成员账户启用 Compute Optimizer。

Compute Optimizer 选择加入流程会自动创建一个 AWS Identity and Access Management (IAM) 服务相关角色。有关更多信息，请参阅 [AWS Compute Optimizer 使用服务相关角色](#)。

Compute Optimizer 根据亚马逊 CloudWatch 指标（例如 CPU、I/O、网络和亚马逊弹性区块存储 (Amazon EBS) 的使用情况）分析您的资源。要生成推荐，需要在过去 14 天内至少连续 30 小时的 CloudWatch 指标数据。如果您启用增强型基础设施指标功能，则利用率指标将延长至 93 天。有关更多信息，请参阅 Compute Optimizer [文档中的 CloudWatch 指标要求和增强型基础架构指标](#)。

Compute Optimizer 根据 vCPU、内存、存储、网络、风险和迁移工作量提供选项以及与每个选项相关的节省。您可以使用 CloudWatch 指标仪表板来分析用于提出建议的数据。利用这些数据，您可以调整运行 SQL Server 工作负载的 EC2 实例的大小。有关如何更改实例类型的更多信息，请参阅 Amazon EC2 文档中的[更改实例类型](#)。

其他资源

- [AWS Compute Optimizer 识别和筛选微软 SQL Server 工作负载 \(AWS\)](#)
- [优化性能和降低许可成本：利用 AWS Compute Optimizer Amazon EC2 SQL Server 实例 \(AWS 博客\)](#)
- [什么是 AWS Compute Optimizer? \(AWS 文档\)](#)
- [查看 EC2 实例建议 \(AWS 文档\)](#)

查看 SQL Server 工作负载 Trusted Advisor 的建议

概述

[AWS Trusted Advisor](#) 提供了一些建议，可帮助您遵循 AWS 最佳实践。通过分析您的使用情况、配置和支出，Trusted Advisor 提供切实可行的建议，以降低成本、提高系统可用性和性能，或帮助填补安全漏洞。本节重点介绍可帮助您降低在中操作 SQL Server 工作负载的成本的 Trusted Advisor 检查 AWS Cloud。

成本优化建议

Trusted Advisor 提供了一些建议，可帮助您在亚马逊弹性计算云 (Amazon EC2) 上优化 SQL Server 工作负载。这些检查会检查您的 SQL Server 工作负载并自动列出需要优化的实例。实施 Trusted Advisor 建议可以降低成本并改善组织的安全状况。

以下是以微软 SQL Server 为重点的 Trusted Advisor 检查：

- 为@@ [微软 SQL Server 超额配置的亚马逊 EC2 实例](#) — 此检查会分析运行 SQL Server 的 Amazon EC2 实例，并在实例超过 SQL Server 软件 vCPU 限制时提醒您。例如，采用 SQL Server 标准版的实例最多可以使用 48 个 vCPU。使用 SQL Server Web 版的实例最多可以使用 32 个 vCPU。

版本	最小 vCPU	vCPU 最大值
Web	4	32

版本	最小 vCPU	vCPU 最大值
Standard	4	48
企业	4	操作系统限制

- [适用于微软 SQL Server 的 Amazon EC2 实例整合](#) — 此检查会分析您的亚马逊 EC2 实例，并在您的实例的 SQL Server 许可证数量少于最小数量时提醒您。您可以整合较小的 SQL Server 实例以帮助降低成本。如果您有许多附带许可证的小型 SQL Server 实例，请考虑整合。根据[微软 SQL Server 2019 许可指南](#)，SQL Server 每个实例至少需要 4 个 vCPU 许可证。如果整合这些数据库，则可以节省许可成本。您可以根据实例上的数据库数量、最大数据库大小和数据库的总大小来做出决定。SQL Server 的 Web 版、标准版和企业版支持整合。有关更多信息，请参阅[整合 SQL Server 数据库](#)（微软博客文章）。

AWS 不建议将大型生产数据库只放在一台服务器上。但是，您可以整合用于非生产环境（例如开发、测试和暂存）的小型环境。这取决于您当前的 SQL Server 使用情况；如果您的数据库使用率低，则可以在一台服务器上进行整合。

配置 Trusted Advisor

执行以下操作来评估以 SQL Server 为重点的签入 Trusted Advisor。

1. 登录到 AWS Management Console。
2. 打开[AWS Trusted Advisor 控制台](#)。
3. 在导航窗格的建议下，选择成本优化。
4. 在成本优化检查清单中，查看微软 SQL Server 的亚马逊 EC2 实例整合状态以及为微软 SQL Server 检查而超额配置的亚马逊 EC2 实例。
 - 绿色复选符号表示您的 Amazon EC2 实例配置最佳。
 - 橙色警报符号表示有改进的机会。
5. 选择一张支票以查看其详细信息和建议。
6. 按照支票提供的说明来优化运行 SQL Server 工作负载的 Amazon EC2 实例。
7. 定期监控您的实例，并定期刷新检查结果。

其他资源

- [Trusted Advisor 查看参考资料](#)（AWS 文档）

- [微软 SQL Server 开启 AWS \(AWS\)](#)
- [微软在 AWS\(AWS\) 上许可](#)
- [SQL Server 2019 定价 \(微软 \)](#)
- [AWS Launch Wizard 适用于 SQL 服务器 \(AWS 文档 \)](#)

容器

现代化是一段变革之旅，它提供了许多选项，包括将整体分解为微服务，使用无服务器函数 () 重新架构应用程序以实现事件驱动，AWS Lambda以及将数据库从 SQL Server 重新利用到 Amazon Aurora 或专门构建的托管数据库。与其他现代化选项相比，将 .NET Framework 应用程序重新平台化到 Linux 和 Windows 容器的现代化路径所需的精力更少。容器具有以下优点：

- 加快创新 — 迁移到容器可以更轻松地实现开发生命周期各个阶段的自动化，包括构建、测试和部署应用程序。通过实现这些流程的自动化，开发和运营团队可以有更多时间专注于创新。
- 降低总拥有成本 (TCO) — 迁移到容器还可以减少对许可证管理和端点保护工具的依赖。由于容器是临时计算单元，因此您可以自动化和简化诸如修补、扩展以及备份和恢复之类的管理任务。这降低了管理和操作基于容器的工作负载的总拥有成本。最后，与虚拟机相比，容器效率更高，因为您可以使用容器通过提供更好的隔离来最大限度地放置应用程序。这可以提高应用程序基础架构资源的利用率。
- 提高资源利用率-与虚拟机相比，容器效率更高，因为您可以使用容器来最大限度地放置应用程序。通过提供更好的隔离，这可以提高应用程序基础架构资源的利用率。
- 缩小技能差距 — AWS 提供沉浸式学习日，提高开发团队在容器技术和 DevOps 实践方面的技能。

本节将介绍以下主题：

- [将 Windows 应用程序移至容器](#)
- [优化 Amazon ECS 上 AWS Fargate 任务的成本](#)
- [了解您的 Amazon EKS 成本](#)
- [使用 App2Container 对 Windows 应用程序进行平台化](#)

有关许可信息，请参阅[亚马逊 Web Services](#) 和 [Microsoft：常见问题解答](#) 的“许可”部分，或者通过电子邮件将问题[发送至 microsoft@amazon.com](mailto:microsoft@amazon.com)。

将 Windows 应用程序移至容器

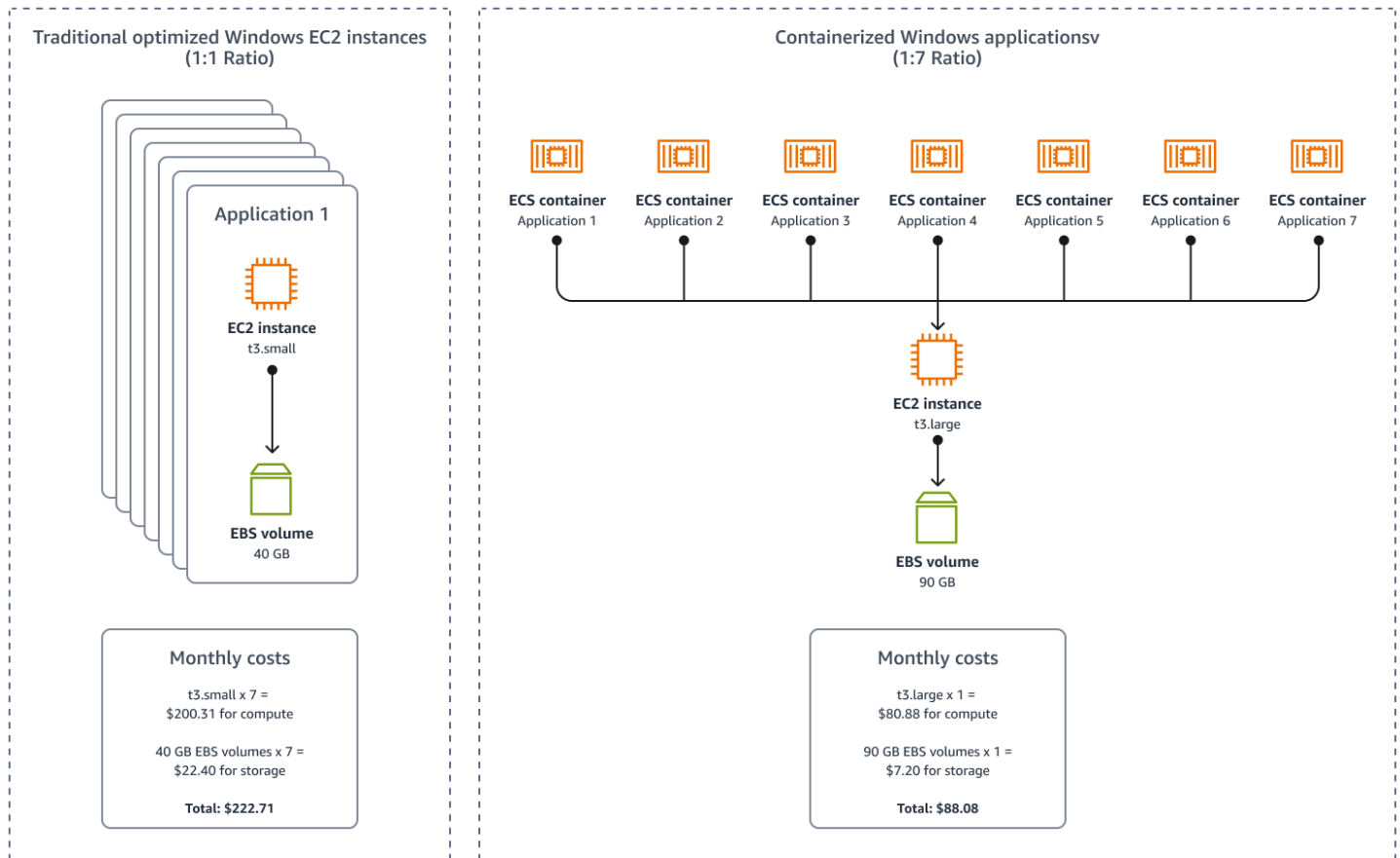
概述

根据 [2021年CNCF年度调查](#)，96%的组织正在使用或评估容器来实现其基础设施的现代化。这是因为容器可以帮助您的组织降低风险，提高运营效率和速度，并实现敏捷性。您还可以使用容器来降低应

用程序的运行成本。本节为跨 AWS 容器服务 (包括[亚马逊弹性容器服务 \(Amazon ECS\)](#)、[Amazon Elastic Kubernetes Service \(Amazon EK S \)](#)) 和 [AWS Fargate](#)

成本收益

以下信息图显示了企业根据[优化和许可评估AWS \(OLA\)](#) 建议，将其 [ASP.NET Framework 应用程序整合到亚马逊弹性计算云 \(Amazon EC2\)](#)[AWS 实例](#)上可以节省的成本。以下信息图显示了通过将应用程序移至 Windows 容器可以节省多少额外费用。



AWS OLA 建议企业进行提升，转向单个 t3.small 实例。如以下性能利用率分析所示，该企业可以通过在本地服务器上运行七个 ASP.NET 应用程序来实现这些节省。

Server name	Storage	Operating system	On-premises CPU AVG utilization	On-premises CPU peak utilization	On-premises RAM (GB)	On-premises RAM AVG utilization (GB)	On-premises RAM peak utilization (GB)	Instance size	vCPU	RAM (GB)
1 AppServer01	60	Windows Server 2012	7.00%	17.00%	8	13.50%	17.10%	t3.small	2	2
2 AppServer02	39	Windows Server 2012	20.07%	22.00%	16	7.50%	12.40%	t3.small	2	2
3 AppServer03	39	Windows Server 2012	24.00%	25.50%	16	8.80%	11.90%	t3.small	2	2
4 AppServer04	4	Windows Server 2012	21.40%	24.00%	16	7.80%	10.70%	t3.small	2	2
5 AppServer05	40	Windows Server 2012	21.30%	23.00%	16	8.20%	12.00%	t3.small	2	2
6 AppServer06	39	Windows Server 2012	21.50%	23.50%	16	7.90%	10.90%	t3.small	2	2
7 AppServer07	39	Windows Server 2012	21.60%	22.90%	16	8.40%	11.50%	t3.small	2	2

进一步的分析表明，通过在容器上运行工作负载，该企业可以节省更多成本。容器可减少操作系统对系统资源（如 CPU、RAM 和磁盘使用量）的开销（将在下一节中介绍）。在这种情况下，企业可以将所有七个应用程序整合到一个 t3.large 实例中，但仍有 3 GB 的备用 RAM。通过使用容器代替 Amazon EC2，迁移到容器可以帮助企业在计算和存储方面平均节省 64% 的成本。

成本优化建议

下一节提供了通过整合应用程序和使用容器来优化成本的建议。

减少你的 Windows 在亚马逊 EC2 上的占用空间

Windows 容器允许你将更多的应用程序整合到更少的 EC2 实例上，从而减少你的 Windows 在 Amazon EC2 上的占用空间。例如，假设您有 500 个 ASP.NET 应用程序。如果你在亚马逊 EC2 上为每个 Windows 应用程序运行一个内核，则等于 500 个 Windows 实例 (t3.small)。如果您假设使用 Windows 容器（使用 t3.large）的比例为 1:7（根据 EC2 实例类型/大小，该比例可能会显著增加），那么您只需要大约 71 个 Windows 实例。这意味着你的 Windows 在亚马逊 EC2 上的占用空间减少了 85.8%。

降低 Windows 许可成本

如果您许可 Windows 实例，则无需许可在该实例上运行的容器。因此，使用 Windows 容器整合 ASP.NET 应用程序可以显著降低 Windows 许可成本。

减少存储占用空间

每次启动新的 EC2 实例时，都需要创建新的亚马逊弹性区块存储 (Amazon EBS) Block Store 卷并支付费用来容纳操作系统。随着规模的扩展，成本也随之扩展。如果您使用容器，则可以降低存储成本，因为所有容器共享相同的基本操作系统。此外，容器使用层的概念，将容器镜像的不可变部分重复用于所有基于该镜像的正在运行的容器。在前面的示例场景中，所有容器都运行 .NET Framework，因此所有容器都共享中间且不可变的 ASP.NET 框架层。

将 end-of-support 服务器迁移到容器

对 Windows Server 2012 和 Windows Server 2012 R2 的支持已于 2023 年 10 月 10 日结束。您可以迁移在 Windows Server 2012 或之前版本上运行的应用程序，方法是将它们容器化为在新操作系统上运行。这样，您就可以避免在不合规的操作系统上运行应用程序，同时利用容器提供的成本效益、更低的风险、运营效率、速度和敏捷性。

使用这种方法需要考虑的一个警告是，您的应用程序是否需要与当前使用的操作系统版本相关的特定 API（例如 COM Interop）。在这种情况下，您必须测试如何将应用程序迁移到较新的 Windows 版本。Windows 容器将其基本容器映像（例如 Windows Server 2019）与容器主机的操作系统（例如，Windows Server 2019）保持一致。通过更改 Dockerfile 中的基础映像并部署到一组运行最新版本 Windows 的新主机，测试和迁移到容器可以让将来更轻松地上升级操作系统。

移除第三方管理工具和许可证

管理服务器群需要使用多种第三方系统操作工具进行修补和配置管理。这会使基础架构管理变得复杂，并且您经常会产生第三方许可成本。如果您在上使用容器 AWS，则无需在操作系统方面管理任何内容。容器运行时管理容器。这意味着底层主机是临时性的，可以很容易地被替换。无需直接管理容器主机即可运行容器。此外，您还可以使用免费工具 AWS Systems Manager Session Manager，例如轻松访问主机和解决问题。

改善控制和便携性

与对 EC2 实例相比，容器可让您更精细地控制服务器资源（如 CPU 和 RAM）。对于 EC2 实例，您可以通过选择实例系列、实例类型和 [CPU 选项来控制 CPU 和 RAM](#)。但是，对于容器，您可以精确地定义要在 ECS 任务定义中为容器或 [Amazon EKS 中的容器分配多少 CPU 或 RAM](#)。实际上，我们建议为 [Windows 容器指定容器级别 CPU 和内存](#)。这种粒度级别带来了成本效益。考虑以下示例代码：

```
json
{
  "taskDefinitionArn": "arn:aws:ecs:us-east-1:123456789012:task-definition/demo-
service:1",
  "containerDefinitions": [
    {
      "name": "demo-service",
      "image": "mcr.microsoft.com/dotnet/framework/samples:aspnetapp-
windowsservercore-ltsc2019",
      "cpu": 512,
      "memory": 512,
```

```
"links": [],
"portMappings": [
  {
    "containerPort": 80,
    "hostPort": 0,
    "protocol": "tcp"
  }
],
```

加速创新

迁移到容器可以更轻松地实现开发生命周期各个阶段的自动化，包括构建、测试和部署应用程序。如果您将这些流程自动化，则可以让开发和运营团队有更多时间专注于创新。

降低总拥有成本

迁移到容器通常可以减少对许可证管理和端点保护工具的依赖。由于容器是临时计算单元，因此您可以自动化和简化诸如修补、扩展以及备份和恢复之类的管理任务。这可以降低管理和操作基于容器的工作负载的总拥有成本。与虚拟机相比，容器效率更高，因为它们允许您最大限度地放置应用程序，从而提高应用程序基础架构资源的利用率。

缩小技能差距

AWS 提供项目和沉浸式学习日，以提高客户开发团队在容器和技术 DevOps 方面的技能。这包括实际咨询和支持。

重构到 .NET 5+ 版本并使用 Linux 容器

虽然您可以通过将 .NET Framework 应用程序迁移到容器来降低成本，但如果将传统的 .NET 应用程序重构为云原生替代方案，则可以进一步节省成本。AWS

取消许可成本

将您的应用程序从 Windows 上的 .NET Framework 重构到 Linux 上的 .NET Core 可以节省大约 45% 的成本。

访问最新的增强功能

将应用程序从 Windows 上的 .NET Framework 重构为 Linux 上的 .NET Core，你可以访问最新的增强功能，例如 Graviton2。与同类实例相比，Graviton2 的性能价格高出 40%。

提高安全性和性能

将应用程序从 Windows 上的 .NET Framework 重构到 Linux 容器上的 .NET Core 可以提高安全性和性能。这是因为您可以获得最新的安全补丁，受益于容器隔离，并且可以访问新功能。

使用 Windows 容器，而不是在一个 IIS 实例上运行多个应用程序

请考虑使用 Windows 容器而不是使用互联网信息服务 (IIS) 在一个 EC2 Windows 实例上运行多个应用程序有以下优点：

- **安全性** — 容器提供了开箱即用的安全级别，这是通过 IIS 级别的隔离无法实现的。如果一个 IIS 网站或应用程序遭到入侵，则所有其他托管的网站都将暴露并容易受到攻击。容器逃脱的情况很少见，而且比通过网络漏洞控制服务器更难被利用。
- **灵活性** — 能够在进程隔离的情况下运行容器并拥有自己的实例，从而可以提供更精细的联网选项。容器还提供跨许多 EC2 实例的复杂分发方法。将应用程序整合到单个 IIS 实例上时，您将无法获得这些好处。
- **管理开销** — 服务器名称指示 (SNI) 会产生需要管理和自动化的开销。此外，您还必须努力处理典型的操作系统管理操作，例如修补、BSOD 故障排除（如果没有自动扩展）、端点保护等。根据[安全最佳实践](#)配置 IIS 站点是一项耗时且持续的工作。您甚至可能需要设置[信任级别](#)，这也会增加管理开销。容器被设计为无状态且不可变。最终，如果您改用 Windows 容器，您的部署将更快、更安全、更可重复。

后续步骤

投资现代基础架构来运行您的传统工作负载可为您的组织带来巨大好处。AWS 容器服务使您可以更轻松的管理您的底层基础架构，无论是在本地还是在云中，因此您可以专注于创新和业务需求。AWS 如今，云端所有容器中有将近80%在运行。AWS 为几乎所有用例提供了一组丰富的容器服务。要开始使用，请参阅[“容器”](#)，网址为 [AWS](#)。

其他资源

- [使用 ECS 容量提供商和 EC2 竞价型实例优化容器工作负载的成本](#) (AWS 博客)
- [Amazon ECS 的成本优化清单和 AWS Fargate](#) (AWS 博客)
- [AWS Graviton2 上的 Amazon EKS 现已上市：多架构应用程序的注意事项](#) (博客) AWS
- [Kubernetes 的成本优化](#) (博客) AWSAWS
- 通过 [Karpenter 整合优化你的 Kubernetes 计算成本](#) (博客) AWS

优化 Amazon ECS 上 AWS Fargate 任务的成本

概述

正确调整 AWS Fargate 任务规模是成本优化的重要一步。通常，应用程序是针对 Fargate 任务使用任意大小构建的，并且永远不会重新访问。这可能会导致 Fargate 任务过度配置和不必要的支出。本节向您展示如何使用 [AWS Compute Optimizer](#) 提供可操作的建议，以便您可以为在 Fargate 上运行的亚马逊弹性容器服务 (Amazon ECS) 服务优化任务 CPU 和内存。Compute Optimizer 还量化了采用这些建议所产生的成本影响。这使您能够根据节省机会的大小确定优化工作的优先顺序。Compute Optimizer 建议为缩小规模的任务提供了容器级的 CPU 和内存配置。

成本收益

在 Fargate 上正确调整 Amazon ECS 任务的大小可以将长时间运行的任务的成本降低 30-70%。无需查看应用程序性能指标来调整任务大小，您就可以将在 EC2 计算实例上使用的相同思维方式应用于容器的大小。这会导致 Fargate 任务过大，从而增加闲置资源的成本。您可以使用 Compute Optimizer 来被动地发现合适的规模机会。理想情况下，应用程序所有者会审查特定的应用程序性能指标并消除操作系统开销，以确保指定正确的任务大小。有关更多信息，请参阅本指南的“[将 Windows 应用程序移至容器](#)”部分。

成本优化建议

本节提供了有关使用 Compute Optimizer 在 Fargate 任务上正确调整亚马逊 ECS 规模的建议。

作为成本优化过程的一部分，我们建议您执行以下操作：

- 启用 Compute Optimizer
- 使用 Compute Optimizer 结果
- 将任务标记为合适的大小
- 启用成本分配标签以使用 AWS 计费工具
- 实施合适的尺码建议
- 在 Cost Explorer 中查看之前和之后的成本

启用 Compute Optimizer

您可以在组织 [AWS Compute Optimizer](#) 级别或单个账户级别启用 AWS Organizations。组织范围的配置为所有成员账户的整个队列中的新实例和现有实例提供持续报告。这样一来，正确调整大小就可以成为重复的活动，而不是一项 point-in-time 活动。

组织级别

对于大多数组织来说，使用 Compute Optimizer 的最有效方式是在组织层面。这为您的组织提供了多账户和多区域可见性，并将数据集中到一个来源供审查。要在组织级别启用此功能，请执行以下操作：

1. 使用具有[所需权限](#)的角色登录您的[AWS Organizations 管理账户](#)，然后选择加入该组织内的所有账户。您的组织必须[已启用所有功能](#)。
2. 启用管理账户后，您可以登录该账户，查看所有其他成员账户，并浏览他们的推荐。

Note

最佳做法是为 Compute Optimizer 配置[委托管理员帐户](#)。这使您能够行使最小权限原则，最大限度地减少对 AWS Organizations 管理帐户的访问权限，同时仍提供对组织范围服务的访问权限。

单一账户级别

如果您的目标账户成本高昂但无权访问，您仍然可以为该账户和 AWS Organizations 地区启用 Compute Optimizer。要了解选择加入流程，请参阅[入门](#)。AWS Compute Optimizer

Note

建议每天刷新，最多可能需要 12 小时才能生成。请记住，Compute Optimizer 需要在过去 14 天内提供 24 小时的指标才能为 Fargate 上的 Amazon ECS 生成建议。有关更多信息，请参阅 Compute Optimizer 文档中的[Fargate 上亚马逊 ECS 服务的要求](#)。

Compute Optimizer 会自动分析 Fargate 上 CloudWatch 亚马逊 ECS 服务的以下亚马逊和亚马逊 ECS 利用率指标：

- CPUUtilization— 服务中使用的 CPU 容量的百分比。
- MemoryUtilization— 服务中使用的内存百分比。

使用 Compute Optimizer 结果

举一个例子，该示例侧重于在单个账户和单个区域内进行正确的规模更改。在此示例中，Compute Optimizer 已在组织级别为所有账户启用。请记住，正确调整大小是一个破坏性的过程，在大多数情况下，该过程是由应用程序所有者在几周的计划维护窗口内精确执行的。

如果您从组织的管理帐户中导航到 Compute Optimizer（如以下步骤所示），则可以选择要调查的帐户。在此示例中，一个任务在中超额配置的单个账户中运行。us-east-1 重点是将大小调整为 Amazon ECS 服务的建议大小。

1. 打开 [Compute Optimizer 控制台](#)。
2. 在“控制面板”页面上，按 findings=overProvisioned 进行筛选，查看 Fargate 上的所有 Amazon ECS 服务。
3. 要查看 Fargate 上过度配置的 ECS 服务的详细建议，请向下滚动，然后选择查看建议。
4. 选择“导出”并保存文件以备将来使用。

Note

要保存建议以备将来查看，您必须有一个可用的 S3 存储桶供每个区域的 Compute Optimizer 写入内容。有关更多信息，请参阅 [Compute Optimizer AWS Compute Optimizer 文档中的亚马逊 S3 存储桶策略](#)。

要查看 Compute Optimizer 提供的建议，请执行以下操作：

1. 在 [Compute Optimizer 控制台](#) 中，进入导出建议页面。
2. 对于 S3 存储桶目标，请选择您的 S3 存储桶。
3. 在导出筛选器部分的资源类型中，选择 Fargate 上的 ECS 服务。
4. 在 Fargate 上的 ECS 服务建议页面上，深入研究 Fargate 上的一项 ECS 服务，查看 Compute Optimizer 提供的 CPU 和内存建议。例如，查看“将当前设置与建议的任务大小进行比较”和“将当前设置与建议的容器大小进行比较”部分中的建议。

要获取需要调整大小的 Fargate ECS 服务列表，请执行以下操作：

1. 打开 [Amazon S3 控制台](#)。
2. 在导航窗格中，选择 Buckets，然后选择您导出结果的存储桶。

3. 在“对象”选项卡上，选择您的对象并选择“下载”。
4. 在你下载的结果中，筛选结果列，仅显示 Fargate 上的 OVER_PROVISIONED Amazon ECS 服务。这显示了您计划针对哪些 Amazon ECS 服务进行适当规模调整。
5. 将任务定义存储在文本编辑器中以备日后使用。

正确调整标签任务的大小

为工作负载添加标签是组织资源的强大工具。AWS 您可以使用标签来获得成本的精细可见性并启用退款。有许多方法和策略可以向 AWS 资源添加标签以处理退款和自动化。有关更多信息，请参阅 AWS 白皮书《[为资源添加标签 AWS 的最佳实践](#)》。以下示例 [AWS CloudShell](#) 用于标记目标账户和中属于任何 Amazon ECS 服务的所有任务 AWS 区域。

```
#!/bin/bash
# Set variables
TAG_KEY="rightsizing"
TAG_VALUE="enabled"
# Get a list of ECS Clusters
ClustersArns=$( w secs list-clusters -query 'clusterArns' -output text)
for ClustersArn in $ClustersArns; do
  ServiceArns=$( w secs list-services -cluster $ClustersArn -query 'serviceArns' -output text)
  for ServiceArn in $ServiceArns; do
    TasksArns=$( w secs list-tasks -cluster $ClustersArn -service-name $ServiceArn -query 'taskArns' -output text)
    for TasksArn in $TasksArns; do
      w secs tag-resource -resource-arn $TasksArn -tags key=$TAG_KEY,value=$TAG_VALUE
    done
  done
done
```

以下代码示例显示了如何为所有 Amazon ECS 服务启用 [标签传播](#)。

```
#!/bin/bash
# Set variables
TAG_KEY="rightsizing"
TAG_VALUE="enabled"
# Get a list of ECS Clusters
ClustersArns=$(aws ecs list-clusters --query 'clusterArns' --output text)
for ClustersArn in $ClustersArns; do
  ServiceArns=$(aws ecs list-services --cluster $ClustersArn --query 'serviceArns' --output text)
```

```
for ServiceArn in $ServiceArns; do
  aws ecs update-service --cluster $ClustersArn --service $ServiceArn --propagate-tags
SERVICE &>/dev/null
  aws ecs tag-resource --resource-arn $ServiceArn --tags key=$TAG_KEY,value=$TAG_VALUE
done
done
```

启用成本分配标签以使用 AWS 计费工具

我们建议激活用户定义的成本分配标签。这样，就可以在 AWS 计费工具（例如和）中识别和筛选 Right sizing 标签。AWS Cost Explorer AWS 成本和使用情况报告如果您不启用此功能，则标签筛选选项和数据将不可用。有关使用成本分配标签的信息，请参阅 AWS Billing and Cost Management 文档中的[激活用户定义的成本分配标签](#)。

等待 24 小时后，您可以在 Cost Explorer 中看到标签，然后在下一节中实施正确的规模建议。为此，请在 Cost Explorer 中搜索 Right sizing 标签。

实施合适的尺码建议

Compute Optimizer 将提供任务或容器大小建议。要实施正确的尺码建议，请执行以下操作。

1. 打开 [Amazon ECS 控制台](#)。
2. 从导航栏中，选择包含您的任务定义的地区。
3. 在导航窗格中，选择 Task definitions（任务定义）。
4. 在 Task definitions（任务定义）页面上，选择任务，然后选择 Create new revision（创建新修订版）。
5. 在 Create new task definition revision（创建新任务定义修订版）页面上，进行更改。[要更新容器大小建议，请在 ECS 任务定义的 ContainerDefinitions 块 memory 下更新 cpu 和。](#) 例如：

```
"containerDefinitions": [
  {
    "name": "your-container-name",
    "image": "your-image",
    "cpu": 1024,
    "memory": 2048,
  }
],
```

6. 验证信息并选择 Create（创建）。

要更新 Amazon ECS 服务，请执行以下操作：

1. 打开 [Amazon ECS 控制台](#)。
2. 在 Clusters (集群) 页面上，选择集群。
3. 在 Cluster overview (集群概述) 页面上，选择服务，然后选择 Update (更新)。
4. 对于任务定义，选择要使用的任务定义族和修订。

对于高级操作员，您可以使用 CloudShell 更新 Amazon ECS 服务。例如：

```
bash
#!/bin/bash
# Set variables
ClustersName="workshop-cluster"
ServiceName="lab7-fargate-service"
TaskDefinition="lab7-fargate-demo:3"
# update the service
aws ecs update-service --cluster $ClustersName --service $ServiceName --task-definition
$TaskDefinition
```

审查费用前后

正确调整资源大小后，您可以使用 Right sizing 标签使用 Cost Explorer 来显示之前和之后的成本。回想一下，您可以使用[资源标签](#)来跟踪成本。通过使用多层标签，您可以精细地了解成本。在本指南中介绍的示例中，Rightsizing 标签用于将通用标签应用于所有目标实例。然后，使用团队标签来进一步组织资源。下一步是引入应用程序标签，以进一步显示运营特定应用程序的成本影响。

举一个例子，说明在单个账户级别使用 Rightsizing 标签可以降低成本。在此示例中，运营成本从每天 30.26 美元增加到每天 7.56 美元。假设每月 744 小时，则在正确调整之前，每年的费用为 11,044.9 美元。调整好尺寸后，年度成本降至 2,759.4 美元。这意味着该账户的计算成本降低了 75%。想象一下这对大型组织的影响。

在开始合适的尺码之旅之前，请考虑以下几点：

- AWS 提供了许多降低成本的选项。这包括 [AWS OLA](#)，AWS 它会在迁移到本地实例之前先检查您的本地实例 AWS。O AWS LA 还为您提供合适规模的建议和许可指南。
- [在购买 Savings Plans 之前，请完成所有合适的规模。](#)这可以帮助您避免超额购买 Savings Plans 承诺。

后续步骤

我们建议采取以下后续步骤：

1. 查看您的现有格局，并考虑将 Amazon EBS gp2 卷转换为 gp3 卷。
2. 查看 [Savings Plans](#)。

其他资源

- [Compute Optimizer 入门](#) (文档) AWS
- [标记 AWS 资源的最佳实践](#) (AWS 白皮书)
- [Windows 容器已开启 AWS](#) (工作AWS 室工作室)

了解您的 Amazon EKS 成本

概述

要有效监控 Kubernetes 部署的成本，就必须采用整体视图。唯一的固定和已知成本是亚马逊 Elastic Kubernetes Service (亚马逊 EKS) 控制平面的费用。这包括构成部署的所有其他组件，从计算和存储到网络，其数量根据您的应用程序需求而定。

您可以使用 [Kubecost](#) 来分析 Kubernetes 基础设施的成本，从[命名空间](#)和[服务](#)一直到各个 [Pod](#)，然后在仪表板中显示数据。Kubecost显示了计算和存储等集群内成本，以及[亚马逊简单存储服务 \(Amazon S3\) 存储桶](#)和[亚马逊关系数据库服务 \(Amazon RDS\)](#) 实例 out-of-cluster 等成本。Kubecost将根据这些数据提出适当规模的建议，并显示可能影响系统的关键警报。Kubecost可以与[集成 AWS 成本和使用情况报告](#)，以显示通过[计算储蓄计划](#)、[预留实例](#)和[其他折扣计划](#)节省的费用。

成本收益

Kubecost 提供报告和控制面板，用于可视化您的 Amazon EKS 部署成本。它使您能够从集群向下钻取到各个组件，例如控制器、服务、节点、容器和卷。这使您可以全面了解在 Amazon EKS 环境中运行的应用程序。通过启用这种可见性，您可以按照 Kubecost 的建议采取行动，或者精细地查看每个应用程序的成本。正确调整 Amazon EKS 节点组的大小与标准 EC2 实例相同的潜在节省。如果您可以调整容器和节点的大小，则可以从运行容器所需的实例大小和 auto Scaling 组中所需的 EC2 实例数量中消除计算膨胀。

成本优化建议

要充分利用 Kubecost，我们建议您执行以下操作：

1. 将 Kubecost 部署到你的环境中
2. 获取 Windows 应用程序的详细成本明细
3. 大小合适的群集节点
4. 大小合适的容器请求
5. 管理未充分利用的节点
6. 补救废弃的工作负载
7. 根据建议采取行动
8. 更新自管节点

将 Kubecost 部署到你的环境中

[亚马逊 EKS Finhack 研讨会](#) 教你如何部署配置为在自有账户中使用 Kubecost 的亚马逊 EKS 环境。AWS 这使您可以亲身体验该技术。如果您有兴趣在组织中举办此研讨会，请联系您的客户团队。

要使用 [Helm](#) 将 Kubecost 部署到你的 Amazon EKS 集群，请参阅博客上 [AWS 和 Kubecost 合作为 EKS 客户提供成本监控的](#) 帖子。AWS 或者，您可以参阅 [Kubecost 官方文档](#)，了解有关安装和配置 Kubecost 的说明。有关 Kubecost 对 Windows 节点的支持的信息，请参阅 Kubecost 文档中的 [Windows 节点支持](#)。

获取 Windows 应用程序的详细成本明细

尽管使用 [Amazon EC2 竞价型实例](#) 可以节省大量成本，但您也可以从 Windows 工作负载往往是有状态这一事实中受益的。竞价型实例的使用取决于应用程序，我们鼓励您验证它们是否适用于您的使用案例。

要获取 Windows 应用程序的详细成本明细，请[登录 Kubecost](#)。在导航页面中，选择“储蓄”。

大小合适的群集节点

在 [Kubecost](#) 中，从导航栏中选择 Savings，然后选择调整集群节点的大小。

举一个例子，Kubecost 报告集群在 vCPU 和 RAM 方面都被过度配置。下表显示了 Kubecost 提供的详细信息和建议。

	Current	建议：简单	建议：复杂
总数	每月 3462.57 美元	每月 137.24 美元	每月 303.68 美元
节点数	4	5	4
CPU	74 个 vCPU	10 个 vCPU	8 个 vCPU
RAM	152 GB	20GB	18 GB
实例细分	2 c5.xlarge + 另外 2 个	5 t3a.medium	2 c5n.large + 另外 1 个

正如 Kubecost 博客文章[为 Kubernetes 集群查找最佳节点集中](#)所述，简单选项使用单节点组，而复杂选项使用多节点组方法。“了解如何采用”按钮可以一键调整集群大小。它需要安装[Kubecost 集群控制器](#)。

如果你使用的是不是由 [eksctl](#) 创建的[自我管理的 Windows 节点](#)，请[参阅更新现有的自我管理节点组](#)。这些说明向您展示了如何在 [Auto Scaling 组](#) 使用的 Amazon EC2 启动模板中更改实例类型。

大小合适的容器请求

在 [Kubecost](#) 中，从导航栏中选择“储蓄”，然后转到“申请大小合适的推荐”页面。本页显示了 pod 的[效率](#)、合理调整大小的建议以及预计的成本节约。您可以使用“自定义”按钮按集群、节点、命名空间\ Controller 等进行筛选。

举个例子，假设 Kubecost 计算出你的某些 Pod 在 CPU 和 RAM (内存) 方面被过度配置。然后，Kubecost 建议您调整到新的 CPU 和 RAM 值，以实现其预计的每月节省。要更改 CPU 和 RAM 值，必须更新[部署清单](#)文件。

管理未充分利用的节点

在 [Kubecost](#) 中，从导航栏中选择“储蓄”，然后选择“管理未充分利用的节点”。

举一个例子，页面显示集群中的一个节点在 CPU 和 RAM (内存) 方面未得到充分利用，因此可能会被耗尽、终止或调整大小。选择未通过节点和 pod 检查的节点可以让你更多地了解为什么它们无法被耗尽。

补救废弃的工作负载

在 [Kubecost](#) 中，从导航栏中选择“储蓄”，然后选择“已放弃的工作负载”页面。在此示例中，您按名为 windows 的命名空间进行筛选。此页面显示了未达到流量阈值且被视为已放弃的 Pod。Pod 需要在定义的时间段内发送或接收一定数量的网络流量。

在仔细考虑放弃了一个或多个 Pod 之后，您可以通过缩减副本数量、删除部署、调整其大小以消耗更少的资源或通知应用程序所有者您认为部署已被放弃来节省成本。

根据建议采取行动

在“调整集群节点大小”部分中，Kubecost 分析了集群中工作节点的使用情况，并就如何正确调整节点大小以降低成本提出了建议。有两种类型的节点组可用于 Amazon EKS：[自我管理和托管](#)。

更新自管节点

有关更新自管理节点的信息，请参阅 Amazon EKS [文档中的自管理节点更新](#)。它指出，使用创建的节点组 eksctl 无法更新，必须使用新配置迁移到新的节点组。

举个例子，假设你有一个名为 ng-windows-m5-2xlarge（它使用 m5.2xlarge EC2 实例）的 Windows 节点组，并且你想将这些 pod 迁移到一个名为 ng-windows-t3-large（[该节点组由 t3.large EC2 实例支持以节省成本](#)）的新节点组。

要在使用由部署的节点组时迁移到新的节点组 eksctl，请执行以下操作：

1. 要查找 Pod 当前所在的节点，请运行 `kubectl describe pod <pod_name> -n <namespace>` 命令。
2. 运行 `kubectl describe node <node_name>` 命令。输出显示该节点正在一个 m5.2xlarge 实例上运行。它还与节点组名称 (ng-windows-m5-2xlarge) 相匹配。
3. 要将部署更改为使用节点组 ng-windows-t3-large，请删除节点组 ng-windows-m5-2xlarge 并运行 `kubectl describe svc,deploy,pod -n windows`。既然部署的节点组已被删除，则立即开始重新部署。

Note

删除节点组后，服务将停机。

4. 几分钟后再次运行该 `kubectl describe svc,deploy,pod -n windows` 命令。输出显示 Pod 都再次处于“运行”状态。

5. 要显示 Pod 现在正在节点组上运行 `ng-windows-t3-large`，请再次运行 `kubectl describe pod <pod_name> -n <namespace>` 和 `kubectl describe node <node_name>` 命令。

其他调整大小的方法

此方法适用于自行管理或托管节点组的任意组合。《[将工作负载从 EKS 自行管理的节点组无缝迁移到 EKS 管理的节点组](#)》博客文章提供了有关如何在不停机的情况下将工作负载从具有超大实例类型的节点组迁移到大小合适的节点组的指导。

后续步骤

Kubecost 可以轻松实现亚马逊 EKS 环境成本的可视化。Kubecost 与 Kubernetes 和 AWS API 的深度集成可以帮助你找到潜在的成本节约。你可以在 Kubecost 的储蓄控制面板中看到这些建议。Kubecost 还可以通过其[集群控制器功能为您实现其中的一些建议](#)。

我们建议您在 AWS 容器博客中查看 step-by-step 部署情况 [AWS , Kubecost 合作为 EKS 客户提供成本监控](#)。

其他资源

- [亚马逊 EKS 研讨会](#) (亚马逊 EKS 研讨会)
- [AWS 与 Kubecost 合作为 EKS 客户提供成本监控 \(AWS 博客 \)](#)
- [亚马逊 EKS Finhack AWS 研讨会](#) (工作室工作室)
- [Windows 容器已开启 AWS](#) (工作AWS 室工作室)

使用 App2Container 对 Windows 应用程序进行平台化

概述

[AWS App2Container](#) 是一款命令行工具，用于将 Java 和 .NET Web 应用程序迁移到容器中并实现现代化。App2Container 分析并生成在裸机、虚拟机、亚马逊弹性计算云 (Amazon EC2) 实例或其他云提供商中运行的所有应用程序的清单。您可以选择要容器化的应用程序。App2Container 将应用程序工件和依赖项打包到容器镜像中，配置网络端口，并生成必要的亚马逊弹性容器服务 (Amazon ECS) 和亚马逊弹性 Kubernetes Service (Amazon EKS) 部署工件，它们是基础设施即代码 (IaC) 模板。App2Container 提供了将容器化应用程序部署到生产环境所需的云基础设施和 CI\ CD 管道。有关更多信息，[请参阅 App2Container 文档中的 App2Container 的工作原理](#)。

借助 App2Container，您可以将应用程序迁移到容器 AWS 并对其进行现代化改造，同时还可以标准化应用程序的部署和操作。您可以使用 App2Container 来帮助快速构建概念验证 (PoC) 或加快在容器中部署生产工作负载。

在使用 Windows 应用程序时，需要记住几件事。App2Container 支持对部署在微软互联网信息服务 (IIS) 上的 ASP.NET 应用程序进行容器化，包括在 Windows Server 2016、Windows Server 2019 或 Windows Server Core 2004 上运行的 IIS 托管的 Windows 通信基金会 (WCF) 应用程序。有关更多信息，请参阅 App2Container [文档中支持的 Windows](#) 应用程序。App2Container 使用 Windows Server Core 作为其容器构件的基础映像，将 Windows Server Core 容器版本与你运行容器化命令的服务器的操作系统 (OS) 版本相匹配。这种方法将应用程序与底层操作系统分离，这样您就可以在不执行传统迁移的情况下升级操作系统。

如果您使用工作计算机对应用程序进行容器化，则容器基础映像（例如 Windows Server 2019 长期服务频道 (LTSC)）与您的工作计算机操作系统（例如 Windows Server 2019）相匹配。如果您直接在应用程序服务器上运行容器化，则版本与您的应用程序服务器操作系统相匹配。如果你的应用程序在 Windows Server 2008 或 2012 R2 上运行，你仍然可以通过为容器化和部署步骤设置工作计算机来使用 App2Container。App2Container 不支持在 Windows 客户端操作系统（例如 Windows 7 或 Windows 10）上运行的应用程序。App2Container 支持 Java 进程的 Tomcat、ToMee 和 JBoss（独立模式）框架。有关更多信息，请参阅 [App2Container 兼容性](#)。

成本收益

与 one-application-to-one 服务器部署设计模式相比，容器化和整合应用程序可以 [节省高达 60% 的计算成本](#)。App2Container 有助于加快应用程序容器化过程。以下是使用 App2Container 来满足现代化需求的一些好处：

- App2Container 不收取额外费用。
- App2Container 在一个容器镜像中支持多个应用程序。
- 使用 App2Container 将您的传统 .NET 应用程序迁移到容器，从而解决即将终止支持的操作系统。您可以迁移到更新的操作系统，无需为扩展支持付费，并降低安全风险。
- 容器是打包 .NET 应用程序的一种高效且经济实惠的方法。在 [MACO 建议——迁移到容器中查看容器的](#)好处。
- 应用程序整合和容器化通过更有效地使用计算资源来帮助减少计算、存储和许可占用空间。
- 迁移到容器可以降低运营开销和基础设施成本，并提高开发可移植性和部署灵活性。

成本优化建议

有关如何使用 App2Container 的说明，请参阅[入门](#)。AWS App2Container 有关 App2Container 命令的信息，请参阅 App2Container 命令参考。

后续步骤

App2Container 可以加快应用程序容器化以及部署到亚马逊 EKS 或 Amazon ECS 的过程。将应用程序部署到容器可以降低计算、网络 and 存储成本，并降低应用程序操作员的运营开销。

如需亲身体会 App2Container，请参阅[使用 AWS App2Container 现代化研讨会](#)。如果您想获得深入的学习体验，请让您的 AWS 客户团队安排一个 App2Container 沉浸式体验日。

其他资源

- [使用容器化复杂的多层 Windows 应用程序 AWS App2Container](#) (AWS 博客文章)
- [使用容器化旧版 ASP.NET 应用程序 AWS App2Container](#) (AWS 博客文章)
- [App2Container 支持的应用程序](#) (文档) AWS
- [利用@@ AWS App2Container 研讨会实现现代化](#) (AWS 工作室工作室)
- [AWS App2Container 常见问题](#) (AWS 网站)

存储

为你的 Microsoft 工作负载选择合适的存储是一项关键的架构决策。作为决策过程的一部分，我们建议您制定存储计划并确定应用程序和服务的功能要求。本章概述了以下存储选项，这些选项可能会影响您的规划。

部分：

- [Amazon EBS](#)
- [Amazon FSx](#)
- [AWS Storage Gateway](#)

Amazon EBS

Amazon Elastic Block Store (AmazonEBS) 是一项完全托管的块存储服务，可让您存储可用于亚马逊弹性计算云 (AmazonEC2) 实例的永久块级存储卷。您可以利用 Amazon 中的多项功能EBS来有效管理和优化云中 Windows 工作负载的存储资源。例如，您可以使用 Amazon EBS 为工作负载预配置所需的确切数量IOPS和吞吐量，从一系列卷类型中进行选择以满足您的工作负载要求，并使用工具来识别和消除浪费的存储资源。这种对存储性能和使用率的精细控制可帮助您优化存储资源，同时避免不必要的成本。

本节将介绍以下主题：

- [将亚马逊EBS卷从 gp2 迁移到 gp3](#)
- [修改亚马逊EBS快照](#)
- [删除未连接的 Amazon 卷 EBS](#)

将亚马逊EBS卷从 gp2 迁移到 gp3

概述

固态硬盘 (SSD) 是生产和高性能工作负载的标准存储选项。Amazon 为中高性能工作负载EBS提供了[通用SSD容量](#)。许多通用卷 AWS 服务（包括亚马逊EC2）的标准是 [gp2](#)，这是这些通用SSD卷的第二代。名为 [gp3 的第三代通用SSDs型于 2020 年 12 月发布](#)。

与上一代产品相比，gp3 产品在性能自定义方面进行了显著改进。对于 Amazon EBS gp2 卷，性能与卷的大小密切相关。每 1 GB 的容量，gp2 卷IOPS的性能就会提高 3。也就是说，一个 2,000 GB 的

gp2 卷能够容纳 6,000 个。IOPS 对于 gp3 卷，可以独立于存储容量自定义性能。这使得即使是小容量的卷也能实现高达 16,000 IOPS 和 1,000 MB/s 吞吐量的性能。

gp3 卷的另一个主要变化是基 IOPS 性能。gp3 的起始音量为 3,000。IOPS 相比之下，gp2 卷的大小必须达到 1 TiB 才能达到相同的性能。对于通常小于 1 TiB 的 C: 驱动器的 Windows Server 来说，从 gp2 升级到 gp3 是一项显著的性能改进。

最后，与 gp2 卷相比，gp3 卷的价格是最大的改进之一。gp3 卷提供所有增强的性能功能，成本仅为 gp2 卷的 20%。

成本影响

由于能够独立于容量扩展性能，因此了解增加额外容量 IOPS 和吞吐量的定价方面非常重要。对于 gp2 卷，定价基于每月每 GiB 0.10 美元的预配置容量。对于 gp3 卷，定价与高性能 [预配置 IOPS SSD 卷类似](#)，后者具有一种容量成本，而额外 IOPS 容量和吞吐量的成本是单独的。

如下表所示，gp3 卷的容量价格为每月 GiB 0.08 美元（比 gp2 低 20%），吞吐量单独的成本为 IOPS 每个预置月 0.005 美元，每个预配置 IOPS 月 0.04 美元，吞吐量超过 125 美元。MiBs MiBs

	gp3	gp2
卷大小	1 GiB – 16 TiB	1GiB – 16TiB
基线 IOPS	3000	3 IOPS /GiB (最小 100IOPS) 至最大 16,000 IOPS
		小于 1 TiB 的体积可以突增至 3,000 IOPS
最大 IOPS /音量	16000	16000
基准吞吐量	125 MiBs	吞吐量限制在 128 MiBs -250 之间 MiBs，具体取决于卷大小
每个卷的最大吞吐量	1,000 MiBs	250 MiBs
价格	每月 0.08 美元/Gib	0.10 USD/GiB-月

	gp3	gp2
	3,000 IOPS 免费，每月预配置 0.005 IOPS 美元 — 超过 3,000	
	125 美元 MiBs 免费，每月预配置 0.04 MiBs 美元，超过 125 美元 MiBs	

Important

尽管 gp3 卷的容量和性能成本各不相同，但如果配置为相同的性能级别，gp3 卷总是比 gp2 卷便宜。

下表显示了在各种容量和性能配置下将 gp2 卷转换为 gp3 卷可以节省成本的示例。

gp2 配置示例

卷大小 (GiB)	马克斯 IOPS	吞吐量 (MiBs)	成本 (USD/月)
30	3000	128	3.00 美元
100	3000	128	10.00 美元
500	3000	250	50.00 美元
1000	3000	250	100.00 美元
2000	6000	250	200.00 美元
6000	16000	250	600.00 美元

gp3 (基准) 配置示例

马克斯 IOPS	吞吐量 (MiBs)	成本 (USD/月)	降低成本 (与 gp2 相比)
3000	125	2.40 美元	20%
3000	125	8.00 美元	20%
3000	125	40.00 美元	20%
3000	125	80.00 美元	20%
3000	125	160.00 美元	20%
3000	125	480.00 美元	20%

gp3 (gp2 匹配) 配置示例

马克斯 IOPS	吞吐量 (MiBs)	成本 (USD/月)	降低成本 (与 gp2 相比)
3000	128	2.52 美元	16%
3000	128	8.12 美元	19%
3000	250	45.00 美元	10%
3000	250	85.00 美元	15%
6000	250	180.00 美元	10%
16000	250	550.00 美元	8%

有关成本分析，请参阅 [Amazon EBS资源中的 gp2 到 gp3 迁移成本节省计算器部分](#)。EBS 您可以下载计算器并使用它来了解将 gp2 卷迁移到 gp3 可以节省多少钱。

成本优化建议

有关如何完成迁移过程的说明，请参阅存储博客上的 [“将您的 Amazon EBS 卷从 gp2 迁移到 gp3 并节省高达 20% 的成本”](#) 一文。AWS

其他资源

- [将您的 Amazon EBS 卷从 gp2 迁移到 gp3，最多可节省 20% 的成本](#) (存储博客) AWS
- [制定 AWS Config 自定义规则以优化 Amazon EBS 卷类型](#) (AWS 云操作和迁移博客)
- [通过删除未使用的 Amazon EBS 卷来控制 AWS 成本](#) (AWS 云运营和迁移博客)
- [亚马逊EBS迁移工具](#) (GitHub)
- [从 2020 年 re: Invent 公告 \(AWS 云财务管理 \) 中节省开支](#)
- [成本优化研讨会](#) (Well-Archit AWS ected Labs)
- [gp2 到 gp3 迁移成本节省计算器](#) (下载)

修改亚马逊EBS快照

概述

删除EBS卷以及管理快照的保留和存档是从一开始就控制成本的重要方面。您可以通过拍摄快照将EBS卷上的数据备份到亚马逊简单存储服务 (Amazon S3)。point-in-time 快照是增量备份，因此它们仅保存最近一次快照后更改的设备上的块。由于无需复制数据，这将最大限度缩短创建快照所需的时间和增加存储成本节省。每个快照都包含将数据 (从创建快照时起) 恢复到新EBS卷所需的所有信息。

EBS快照费用按每月千兆字节计算。您需要为快照的大小和保留快照的时间付费。定价因存储层而异。对于[标准级别](#)，您只需为存储的已更改区块付费。对于存档层，您需要为存储的所有快照块付费。您还需要为从[存档](#)层检索快照付费。以下是每个存储层的示例场景：

- 标准层 — 您的卷可存储 100 GB 的数据。您需要为第一个快照 (快照 A) 的完整 100 GB 数据付费。下次拍摄快照 (快照 B) 时，您有 105 GB 的数据。然后，您只需为增量快照 B 额外的 5 GB 存储空间付费
- 存档层 — 您存档快照 B。然后，快照将移至存档层，您需要为完整的 105 GB 快照块付费。

您可以使用 [Amazon Data Lifecycle Manager](#) 来帮助您设置生命周期，以便按计划保留和管理快照。

成本影响

EBS卷和快照的费用是分开管理的。EBS快照的计费率低于活动EBS卷。当实例终止时，每个附加EBS卷的[DeleteOnTermination 属性](#)值决定是保留还是删除该卷。默认情况下，根卷True的DeleteOnTermination属性设置为。False对于所有其他音量类型，该值设置为。这会造

成这样的情况：操作员打算删除EC2实例，但除了根卷外，还会留下已添加到该实例的卷。有关检查不再需要的卷（及其关联快照）的说明，请参阅[亚马逊EBS文档中的查看有关亚马逊EBS卷的信息](#)。

默认情况下，当您创建快照时，快照将存储在 Amazon S EBS snapshot 标准套餐（标准套餐）中。存储在标准层中的快照可递增。这意味着只在卷上保存在最新快照之后发生更改的数据块。[Amazon S EBS snapshots Archive](#) 是一个新的存储层，您可以使用它来低成本、长期地存储您很少访问的快照，无需频繁或快速检索。标准版和存档版的价格差异很大，在设置快照策略时应该是一个关键的考虑因素。对于计划存储 90 天或更长时间且很少需要访问的快照，Amazon EBS Snapshots Archive 可将快照存储成本降低多达 75%。

亚马逊EBS快照存储	费用
Standard	每月 0.05 美元/GB
档案	每月 0.0125 美元/GB

在较小的环境中，成本节省可能不大。在大规模情况下，即使EBS卷已被删除，仍有多个账户和成千上万个带有EBS快照TBs的EC2实例，则节省的费用更为显著。

下表比较了每月使用量仅为 50 TB 的标准层和存档层。即使规模较小，每年仍可节省数千美元。

亚马逊EBS快照存储	每月费用	每年的成本
标准 50 TB	312.50 美元	3,750 美元
存档 50 TB	78.13 美元	937.60 美元
	每年节省开支	2,812.40 美元

成本优化建议

删除快照可能不会降低组织的数据存储成本。其他快照可引用已删除快照的数据，已引用的数据总是会被保留。例如，当您为具有 10GiB 数据的卷创建首个快照时，快照大小也为 10GiB。由于快照是增量式的，您为同一个卷创建的第二个快照仅包含在第一个快照创建后发生变更的数据块。第二个快照还会引用第一个快照中的数据。如果您更改 4 GiB 的数据并拍摄第二张快照，则第二个快照的大小为 4 GiB。此外，第二个快照将引用第一个快照中没有发生变更的 6GiB 数据。有关更多信息，请参阅[删除卷快照然后删除EBS卷本身，为什么我的存储成本没有降低？](#) 在 AWS 知识中心中。

请考虑以下事项：

- 您无需为他人 AWS 账户 拥有并与您的账户共享的快照付费。只有当您共享快照复制到您的账户时，才会向您收费。您还需要为通过共享快照创建的EBS卷付费。
- 如果快照（快照 A）被另一个快照（快照 B）引用，则删除快照 B 可能不会降低存储成本。删除快照时，只会删除该快照所特有的数据。其他快照引用的数据仍然存在，您需要为此引用的数据付费。要删除增量快照，请参阅 Amazon EBS 文档中的[增量快照删除](#)。

在中运行工作负载时，快照清洁度是标准操作规范 AWS。随着时间的推移，对于你不需要的数据，快照会增加昂贵的费用。

其他 资源

- [通过删除未使用的 Amazon EBS 卷来控制 AWS 成本](#)（AWS 云运营和迁移博客）
- [删除亚马逊EBS快照](#)（亚马逊EBS文档）
- [成本优化研讨会](#)（Well-Archit AWS ected Labs）
- 使用 [Amazon Data Lifecycle Manager 自动存档亚马逊EBS快照](#)（AWS 存储博客）

删除未连接的 Amazon 卷 EBS

概述

未连接（孤立）的EBS卷可能会给您的 AWS 环境带来不必要的存储成本。必须将定期审查和删除未使用和未使用的EBS卷作为 AWS 环境卫生的一部分。最佳做法是制定持续审查EBS卷使用情况的流程。您可以使用来查看未充分利用[AWS Compute Optimizer](#)的实例。本节可帮助您识别、管理和删除未连接或未充分利用的EBS卷。

Amazon EBS

[Amazon Elastic Block Store \(AmazonEBS\)](#) 是一款块级设备，为亚马逊弹性计算云 (AmazonEC2) 实例提供存储量。EBS提供永久存储，可以灵活地连接和分离EC2实例。这意味着，即使EC2实例终止，EBS卷的生命周期也会持续下去。该[DeleteOnTermination](#)属性是一项功能，用于控制在实例终止时是保留还是删除附加的EBS卷。默认情况下，根卷True的属性设置为，从而导致删除。False对于其他卷，则将其设置为，因此可以保存。

成本影响

未连接的EBS卷，也称为未使用的卷或孤立的卷，根据预配置的存储大小和存储类型，其费用与连接的卷相同。尽管 Amazon EBS 收费的平均成本似乎微乎其微，仅为每月 0.10 美元，但必须认识到，随着时间的推移，未使用EBS量的积累可能会导致巨额成本。

例如，考虑保留 50 个未使用的EBS卷所产生的后果，每个卷的存储大小均为 100 GB，如下表所示。

存储卷数量	卷类型	大小	每月总费用
50 卷	gp2 (0.10 美元) USD	100 GB	100 GB 50.00 EBS 卷 月 0.10 美元 USD = 500.00 美元 USD

根据上表中的设想，每月可减少约500美元或每年6,000美元的成本。这是朝着降低成本迈出的有效一步。请务必将删除未连接的EBS卷作为 AWS 环境卫生的常规做法。

成本优化建议

您可以使用轻松 AWS 地自动删除未连接的EBS卷。例如，您可以使用 AWS Lambda、AWS Config CloudWatch、Amazon 和，根据年限、标签和其他规格 AWS Systems Manager 来定义删除独立卷的标准。您还可以使用它们 AWS 服务 来大规模自动执行清理过程。

为避免意外后果，我们建议您在删除未连接EBS的卷之前进行尽职调查。

管理未连接的卷 EBS

我们建议您考虑以下最佳实践：

- 满足合规性要求 — 验证删除独立EBS卷是否符合贵组织的治理和合规性要求。
- 设置数据备份和保留策略-在删除未连接的EBS卷之前，请将所有重要数据备份到另一个存储库（例如 [Amazon S3](#)）。就数据保留而言，[Amazon EBS 快照](#)是比EBS卷更具成本效益的保留数据的方式，将来如果需要，它们可以恢复卷。有关有效管理快照的更多信息，请参阅本指南的[修改 Amazon EBS 快照](#)部分。
- 检查依赖关系-检查未连接的EBS卷与其他 AWS 资源之间是否存在任何依赖关系。您可以使用[AWS Management Console](#) 或[API](#)来收集有关您的EBS卷的描述性信息，例如大小、状态和关联的资源。这是防止删除任何暂时分离的资源的重要步骤。
- 创建保留策略 — 为未连接的EBS卷设定保留期。这可以帮助您确定删除未连接卷的合适时间，从而确保您的 AWS 环境保持优化。例如，您可以创建 A [mazon EventBridge](#) 规则来按计划启动 Lambda

函数。Lambda 函数可以使用主动识别任何未连接的EBS卷，应用标记机制 AWS SDK以便于跟踪，并在未连接的EBS卷达到或超过定义的阈值时发出通知。

- 标记未连接的EBS卷 — [标记EBS卷](#)是一种有用的做法，可以帮助根据环境、应用程序或所有者等属性组织和识别卷。这在决定要删除哪些未连接的卷时特别有用，因为它使您能够根据标签快速识别不再需要的卷。
- 确保安全删除 — 查看上次连接EBS卷的时间可以帮助您确定删除该卷是否安全。有关更多信息，请参阅[如何使用 AWS CLI 命令列出特定 Amazon EBS 卷的附件或分离历史记录？](#)在 AWS 知识中心中。
- 识别未充分利用的EBS卷 — 强烈建议识别并删除未充分利用的EBS卷，以降低存储成本和维护优化的环境。AWS Trusted Advisor 并[AWS Compute Optimizer](#)可以帮助您识别未充分利用的EBS容量，并提供降低成本和提高效率的建议。例如，请参阅使用 [\(\) 设置自动化以优化EBS卷](#)、[建立 Trusted Advisor 组织](#) [AWS Trusted Advisor\(GitHub\)](#) [控制面板 \(TAOWorkshop AWS p Studio\)](#) 和[使用 AWS Compute Optimizer \(AWS 存储博客 \) 优化亚马逊EBS卷的成本](#)。

自动清理未连接EBS的卷

我们建议您考虑使用以下工具来帮助您自动清理未连接的EBS卷：

- [AWS APIs\(DescribeVolumes\)](#) — 您可以使用AWSSDKs或 AWS Command Line Interface (AWS CLI) 筛选和查找未连接的EBS卷。您可以使用按计划运行的脚本或 [Lambda](#) 函数自动执行此过程，从而节省时间和精力。中的[示例脚本](#) GitHub 演示了其工作原理。该脚本使用 Lambda 来分析 AWS CloudTrail 日志并识别未EBS连接的卷。
- [AWS Systems Manager 自动化](#) — 这使您能够自动执行基础架构中的日常维护和补救任务。首先，请[创建自动化运行手册](#)，其中定义了要按特定顺序执行的一系列步骤。例如，您可以创建一个 runbook，它首先创建未连接的EBS卷的快照，然后删除该卷本身。这可以帮助您自动执行任务，否则如果手动完成，这些任务将非常耗时且容易出错。
- [AWS Config](#)— 这使您能够评估、审计和跟踪 AWS 资源随时间推移而发生的变化。通过捕获配置更改，您可以使用 AWS Config 来评估环境中的合规性、治理和资源利用率。例如，AWS Config 可以识别[未使用的EBS卷](#)。此外，您可以将 AWS Systems Manager Automation 与关联 AWS Config 以自动修复未使用EBS卷的删除。

其他资源

- 使用@@ [AWS Config](#) 和 [\(AWS 规范性指南EBS \) 删除未使用的亚马逊 Elastic Block Store AWS Systems Manager\(Amazon\) 卷](#)
- [通过删除未使用的 Amazon EBS 卷来控制 AWS 成本](#) (AWS 云运营和迁移博客)

- [AWSConfigRemediation-DeleteUnused EBSVolume](#) (AWS Systems Manager 自动化运行手册参考)

Amazon FSx

Amazon FSx for Windows File Server 是一项完全托管的文件存储服务，已针对 Windows 工作负载进行了优化。它为您提供了一个简单且可扩展的解决方案，无需复杂的存储基础架构管理即可运行基于 Windows 的应用程序和工作负载。您可以使用 FSx Windows File Server 轻松配置和访问原生支持您的 Windows 应用程序的共享文件存储，包括微软SQL服务器 SharePoint、微软和自定义。NET应用程序。此外，FSxfor Windows File Server 通过提供灵活的定价选项（例如 pay-as-you-go 和存储配额）和自动重复数据删除来减少存储占用空间并优化性能和成本，从而帮助您管理成本。

本节将介绍以下主题：

- [选择合适SMB的文件存储空间](#)
- [在 Amazon 中启用重复数据删除 FSx](#)
- [了解 Windows 文件服务器中的FSx数据分片](#)
- [了解 Amazon 的HDD容量使用情况 FSx](#)
- [使用单个可用区](#)

选择合适SMB的文件存储空间

概述

AWS 提供各种完全托管的存储服务，为您提供业界领先的文件服务的丰富功能，同时将最新的 AWS 基础架构创新与安全性相结合。您可以将 AWS 服务整合到基础设施即代码 (IaC) 工作流程中，并将其与 AWS 计算、监控和数据保护服务集成。对于 Windows 工作负载，您可以从两种完全托管的文件服务中进行选择，以满足您的应用程序需求：适用FSx于 Windows 文件服务器和适用于 Amazon FSx 的服务 NetApp ONTAP。

FSx适用于 Windows 文件服务器

Amazon FSx for Windows File Server 提供基于 Windows Server 的完全托管的共享存储，并提供各种数据访问、数据管理和功能。FSxfor Windows File Server 可轻松与 Windows 环境集成，因为它是一项 Windows 原生服务。FSx对于用户和组共享，我们建议使用适用于 Windows 文件服务器，为 SQL服务器、Windows 应用程序和虚拟桌面基础架构使用始终开启的故障转移群集实例 (VDI)。FSx

适用于 Windows 的文件服务器还可以很好地与亚马逊FSx文件网关、亚马逊 Kendra、亚马逊 S3 的审计日志和亚马逊 Data Firehose 集成。

FSx for ONTAP

FSx for ONTAP 或基于 NetApp 的专有 ONTAP 文件系统。这需要一定程度的技能提升，主要推荐给现有的本地 NetApp 用户。典型的用例包括用户和组共享、SQL 服务器的 Always On Failover 群集实例以及 Windows 应用程序。FSx for ONTAP 支持多种协议、大于 64 TB 的文件系统（不带 DFS 命名空间服务器的 PB 规模）、克隆、复制、快照、压缩（存储效率）和数据的智能分层。

成本影响

FSx 适用于 Windows 文件服务器

FSx 适用于 Windows File Server 是第一个 AWS 用于为 SQL 服务器部署故障转移群集实例的共享存储解决方案。对于 FSx 适用于 Windows File Server，您可以使用 SQL 标准版许可启动故障转移群集实例。但是，这可以防止您依赖需要 SQL 服务器企业版许可证的 Always On 可用性组。通过从 SQL Server Enterprise SQL 标准版切换到服务器标准版，您可以节省 65-75% 的 [SQL 服务器](#) 许可费用。

您可以使用 FSx 适用于故障转移群集实例的 Windows 文件服务器来卸载典型 EBS 存储中的存储 I/O。通过将 I/O 卸载到 FSx Windows 文件服务器，您可以缩小依赖高 Amazon EBS 吞吐量的 EC2 实例 IOPS，而不会影响存储吞吐量。

FSx for ONTAP

您可以使用 FSx for 在块协议上运行 Microsoft 故障转移群集，SCSI 并受益于 SQL 服务器即时文件初始化、跨区域复制使用 SnapMirror、防病毒支持和克隆。ONTAP 如果您创建多个数据库副本进行测试，则克隆可以在空间消耗和创建这些数据库副本的速度方面产生显著差异。此外，您可以使用 FSx for NetApp SnapCenter 来管理 SQL 服务器 EC2 实例的备份、还原和克隆功能 ONTAP。FSx for ONTAP 还提供从低成本容量池存储 SSD 到低成本容量池存储的自动分层，兼具性能和成本效益。

FSx for NetApp 或 ONTAP 支持的文件系统 (ONTAP)，与 FSx 支持 Windows 本机文件系统的 Windows NTFS 文件服务器不同。的最小大小 FSx for ONTAP 为 1024 GB，而 Windows File Server 的最小大小可以低至 32 GB。

与微软分布式文件系统集成

FSx 适用于 Windows 文件服务器，也 FSx 适用于 ONTAP 与微软的 [分布式文件系统 \(DFS\)](#) 集成，可无缝集成到现有部署中。在规划架构时，请记住以下几点：

- FSx适用于 Windows 文件服务器，FSxONTAP支持两种部署类型 ([多个可用区和单个可用区 DFSN](#)) 上的DFS命名空间 ()。
- 仅FSx适用于 Windows 文件服务器支持[DFS复制 \(DFSR\)](#)，并且仅在使用单个可用区时才支持复制。

成本优化建议

Windows File Server 和 Windows 文件服务器FSx的性能都FSxONTAP非常依赖于配置，它们的定价也是如此。FSx适用于 Windows File Server 的定价主要取决于存储容量和存储类型、吞吐容量、备份和传输的数据。使用 FSx fo ONTAP r，您需要为SSD存储SSDIOPS、容量池使用量、吞吐容量和备份付费。

文件服务	5 TB 存储空间的成本	配置	区域
FSx适用于 Windows 文件服务器	982.78 美元	单可用区 SSD(15,000IOPS) 32 MBps 5 TB 的备份 (不节省重复数据删除费用)	美国东部 (弗吉尼亚州北部)
FSx for ONTAP	979.28 美元	单可用区 100% SSD 15,000 个读写容量层 15000 SSD IOPS 128 MBps 5 TB 的备份 (不节省重复数据删除费用)	美国东部 (弗吉尼亚州北部)

记住以下内容：

- 重复数据删除和压缩使您能够通过缩小数据大小在物理设备上存储更多数据，但您需要为预配置的固态硬盘 (SSD) 或硬盘驱动器 (HDD) 存储付费。
- 您可以使用 `f` FSx or ONTAP 对数据进行分层。定期访问您的全部数据并需要 SSD 存储的情况极为罕见。您可以将冷数据和不常访问的数据移至容量层，以节省成本。
- 此处提到的价格是根据该等级的 100% 数据和该 SSD 等级的 15,000 IOPS % 数据计算得出 SSD 的。

备份

默认情况下，无论 FSx 是 Windows 文件服务器还是 FSx 适用于 Windows 文件服务器，都将其完全托管的备份存储在 Amazon S3 上。ONTAP 但是，使用 FSx for ONTAP 还有一个额外的备份选项 SnapVault，它可以将备份配置为驻留在容量层中。使用备份 SnapVault 是一种自我管理的机制，它比默认的完全托管备份选项更具成本效益。完全托管的备份选项为每月每 GB 0.05 美元。ONTAP (容量池存储空间为 10:1) FSx 的 SnapVault 备份费用为 0.03221 SSD 美元 ($0.9 \times 0.0219 + 0.1 \times 0.125$)。

记住以下内容：

- AWS 托管备份的间隔时间为一小时。[SnapVault](#) 可以让你低至五分钟。
- 您可以使用 NetApp 的工具 (例如 CLI 和 API) 来配置 SnapVault 关系和快照复制。
- 在卷上启用 all 分层策略，以使用容量 SnapVault 量层作为备份数据的存储。
- SnapVault 目的地可以位于同一个区域 AWS 区域、跨区域或本地。这通常是指向单个可用区或多个可用区文件系统的备份目标。相比之下，AWS Backup 它得益于 Amazon S3 的区域弹性。

合适的尺寸

您还可以通过调整大小和防止过度配置来节省成本并充分利用文件系统。

要正确调整尺寸，请执行以下操作：

1. 根据数据确定您当前的需求。对于典型的 Windows 工作负载，您可以使用[性能监视器](#)等内置操作系统工具。
2. 在性能监视器中，使用以下计数器来衡量您当前的性能需求。捕获间隔设置为一秒，最大日志大小为 1,000 MB，并启用覆盖。

```
Logman.exe create counter PerfLog-Short -o "c:\perflogs\PerfLog-Long.blg" -f bincirc
-v mmdhmm -max 1024 -c "\LogicalDisk(*)\*" "\Memory\*" "\.NET CLR Memory(*)\*"
"\Cache\*" "\Network Interface(*)\*" "\Paging File(*)\*" "\PhysicalDisk(*)\*"
"\Processor(*)\*" "\Processor Information(*)\*" "\Process(*)\*" "\Thread(*)\*"

```

```
"\Redirector\*" "\Server\*" "\System\*" "\Server Work Queues(*)\*" "\Terminal Services\*" -si 00:00:01
```

3. 要开始捕获日志，请运行 `logman start PerfLog-Short` 命令。要停止日志捕获，请运行 `logman stop PerfLog-Short` 命令。

Note

可以在运行捕获的服务器上的 `c:\perlogs` 中找到性能日志文件。有关更多信息，请参阅 Microsoft 文档中的 [Windows 性能监视器概述](#)。

4. 确定正确的配置后，使用诸如微软之类的磁盘压力工具，在亚马逊 FSx 文件系统上测试您的估计是否正确 [DISKSPD](#)。
5. 如果您对性能感到满意，请切换到文件共享。

我们建议采用保守的存储容量方法，因为存储容量只能向上扩展。吞吐容量可以根据需要向上和向下扩展。

其他资源

- [Amazon FSx for NetApp ONTAP FAQs](#) (AWS 网站)
- [使用新指标优化 Amazon FSx 的 Windows 文件服务器性能](#) (AWS 存储博客)

在 Amazon 中启用重复数据删除 FSx

概述

重复数据删除功能使您能够以更低的容量需求更高效地存储数据。它涉及在不影响数据保真度或完整性的前提下查找和删除数据中的重复项。重复数据删除使用子文件可变大小的分块和压缩，一般文件服务器的优化率为 2:1，虚拟化数据的优化率高达 20:1。重复数据删除比 NTFS 压缩有效得多。重复数据删除架构的固有特点是硬件故障期间的弹性——对数据和元数据进行全面校验和验证，包括元数据冗余和访问量最大的数据块。

FSx 适用于 Windows 文件服务器完全支持重复数据删除。使用它可以为通用文件共享平均节省 50-60%。在股票范围内，用户文档可节省 30-50%，软件开发数据集最多可节省 70-80%。重要的是要明白，通过重复数据删除可以节省的存储空间取决于数据集的性质，包括文件之间存在多少重复数据。如果存储的数据本质上是动态的，则重复数据删除不是一个好选择。

成本影响

为了应对企业中数据存储的增长，管理员整合了服务器，并将容量扩展和数据优化作为关键目标。重复数据删除的默认设置可以立即节省开支，或者管理员可以微调设置以获得更多收益。例如，您可以将其配置为仅在特定文件类型上运行重复数据删除，也可以创建自定义作业计划。

总体而言，重复数据删除有三种类型的作业：优化、垃圾收集和清理。请注意，在优化后运行垃圾回收作业之前，空间不会被释放。您可以安排作业，也可以手动运行它。计划重复数据删除作业时可用的所有设置也可在您手动启动作业时使用（计划特定的设置除外）。

即使重复数据删除仅有效节省了25%，但Windows文件服务器还是可以节省大量成本。FSx这些预计节省的费用基于中的[估计](#) AWS Pricing Calculator。

成本优化建议

默认情况下，Windows FSx 文件服务器文件系统未启用重复数据删除。要使用[远程管理](#)启用重复数据删除 PowerShell，必须运行Enable-FSxDedup命令，然后使用该Set-FSxDedupConfiguration命令来设置配置。有关更多信息，请参阅 Windows [文件服务器文档中的 FSx管理文件系统](#)。

要启用重复数据删除，请运行以下命令：

```
PS C:\Users\Admin> Invoke-Command -ComputerName amznfsxzzzzzzzzz.corp.example.com -  
ConfigurationName FSxRemoteAdmin -ScriptBlock {Enable-FsxDedup }
```

要验证您的重复数据删除配置，请运行以下命令：

```
Invoke-Command -ComputerName amznfsxzzzzzzzzz.corp.example.com -ConfigurationName  
FSxRemoteAdmin -ScriptBlock {  
Set-FSxDedupSchedule -Name "CustomOptimization" -Type Optimization -Days  
Mon,Tues,Wed,Sat -Start 09:00 -DurationHours 7  
}
```

通过运行 PowerShell Measure-DedupFileMetadata cmdlet，您可以确定如果删除一组文件夹、一个文件夹或单个文件，然后运行垃圾收集作业，则可以回收卷上多少潜在的磁盘空间。具体而言，该DedupDistinctSize值告诉您在删除这些文件后可以获得多少空间。文件中的区块通常在其他文件夹之间共享，因此重复数据删除引擎会计算哪些区块是唯一的，哪些区块将在垃圾收集任务结束后被删除。

默认的[重复数据删除任务计划](#)旨在很好地适用于推荐的工作负载，并且尽可能保持非侵入性（不包括为备份使用类型启用的优先级优化作业）。如果工作负载需要大量资源，我们建议您安排作业仅在空闲时间运行，或者减少或增加允许重复数据删除作业消耗的系统资源量。

默认情况下，重复数据删除会占用可用内存的 25%。但是，可以通过使用来增加此值-memory switch。对于优化任务，我们建议您将范围设置为 15 到 50。对于计划作业，您可以使用更高的内存消耗。例如，对于垃圾收集和清理作业（通常安排在非工作时间运行），您可以设置更高的内存消耗（例如 50）。

有关重复数据删除设置的更多信息，请参阅 Windows File Server 文档中的[使用重复数据删除降低存储成本](#)。FSx

其他资源

- [了解重复数据删除](#)（微软文档）
- 通过@@ [重复数据删除降低存储成本](#)（FSx适用于 Windows 文件服务器文档）

了解 Windows 文件服务器中的FSx数据分片

概述

FSx对于 Windows 文件服务器，性能取决于配置。它主要基于存储类型、存储容量和吞吐量配置。您选择的吞吐量决定了文件服务器可用的性能资源，包括网络 I/O 限制、CPU和内存以及文件服务器施加的磁盘 I/O 限制。您选择的存储容量和存储类型决定了存储卷的可用性能资源，即存储磁盘施加的磁盘 I/O 限制。除了性能外，配置选择还会影响成本。FSx适用于 Windows File Server 的定价主要取决于存储容量和存储类型、吞吐量、备份和传输的数据。

如果您的文件存储空间和性能要求相对较大，则可以从数据分片中受益。数据分片涉及[将文件数据划分为较小的数据集](#)（分片），然后将其存储在不同的文件系统中。从多个实例访问您的数据的应用程序可以通过并行读取和写入这些分片来实现高水平性能。同时，您仍然可以在公共命名空间下向应用程序呈现统一视图。此外，它还可以帮助将文件数据存储扩展到每个文件系统对大型文件数据集所支持的容量 (64 TB) 之外，最高可达数百 PB。

成本影响

对于大型数据集，通常更有效的做法是为 Windows File Server 文件系统部署多个小型FSx数据集，而不是部署一个大型SSD共享以实现相同的性能级别。将FSx适用于 Windows File Server HDD 和SSD 存储类型的组合使用可以更好地节省成本，并使您能够将工作负载与最佳底层磁盘子系统相匹配。在下

表中，您可以看到单个 17 TB 文件系统之间的区别，并将其与添加相同容量的多个较小文件系统进行比较。

具有多个工作负载的大型SSD文件系统

服务器名称	费用	配置	区域
FSx适用于 Windows 文件服务器的亚马逊	5,716 美元 USD	17 TB SSD 30% 的重复数据删除率 256 Mbps 17 TB 的备份	美国东部 (弗吉尼亚州北部)

使用对工作负载进行分区 DFSN

服务器名称	费用	配置	区域	共享
FSx适用于 Windows 文件服务器的亚马逊	1,024 美元 USD	2 TB SSD 重复数据删除率为 20% 128 Mbps 2 TB 的备份 多可用区	美国东部 (弗吉尼亚州北部)	分享 1
FSx适用于 Windows 文件服务器的亚马逊	2,132 美元 USD	5 TB SSD 30% 的重复数据删除 256 Mbps 5 TB 的备份	美国东部 (弗吉尼亚州北部)	分享 2

服务器名称	费用	配置	区域	共享
		多可用区		
FSx适用于 Windows 文件服务器的亚马逊	1,036 美元 USD	10 TB HDD 40% 重复数据删除 128 Mbps 10 TB 的备份 多可用区	美国东部 (弗吉尼亚州北部)	分享 3
DFSNWindows EC2 实例	27 美元 USD	t3a.medium 2 vCPUs 4 GiB 内存	美国东部 (弗吉尼亚州北部)	DFSN实例

大型SSD文件系统的年成本为 68,592 美元。分区工作负载的年度成本为 50,640 美元。在此示例中，将工作负载与适当的后端存储相匹配的同时，可以节省 26%。有关定价估算的更多信息，请参阅[AWS Pricing Calculator](#)估算。

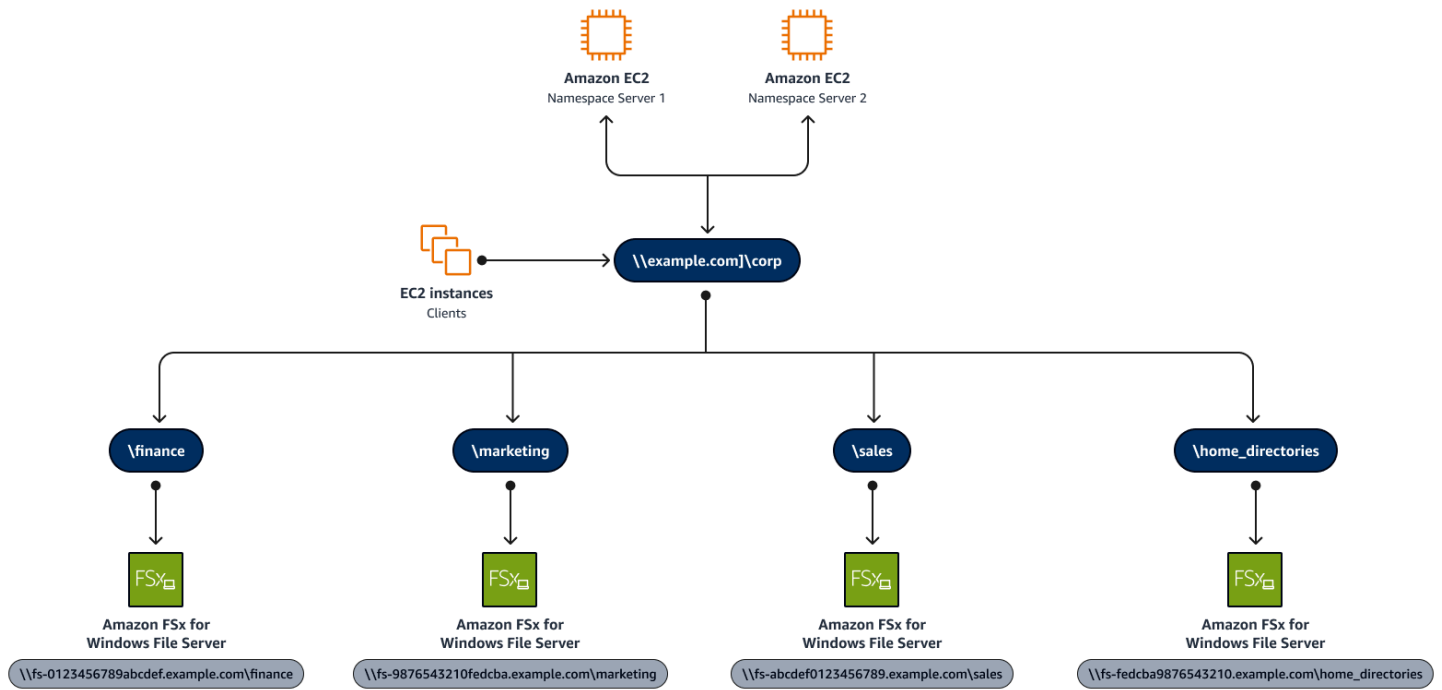
成本优化建议

要部署重复数据删除解决方案，必须根据数据类型、I/O 大小和 I/O 访问模式设置 [Microsoft DFS 命名空间](#)。每个命名空间总共支持多达 50,000 个文件共享和数百 PB 的存储容量。

选择一种在计划使用的所有文件系统之间均匀分布 I/O 的分片约定是最有效的。监控您的工作负载将有助于进一步优化或降低成本。如果您在衡量亚马逊FSx文件系统的性能信息时需要帮助，[FSx请参阅 Windows 文件服务器文档中的 Window FSx s 文件服务器性能](#)。

选择分片策略后，您可以使用DFS命名空间对文件系统进行分组，以便轻松访问您的共享。这使用户能够看到同质文件系统，而实际上他们正在使用专门构建的用例访问各种不同的文件系统。使用正确的命名约定创建共享非常重要，这样您的最终用户就可以轻松解读共享是为何种工作负载而设计的。为生产和非生产共享添加标签也很重要，这样最终用户就不会错误地将文件放在错误的文件系统中。

下图显示了如何使用单个DFS命名空间作为多个 Amazon FSx 文件系统的访问点。



记住以下内容：

- 您可以将现有FSx的 Windows 文件服务器共享添加到DFS树中。
- FSx无法将 Amazon 添加到DFS共享路径的根目录。您只有一个子文件夹。
- 您必须部署一个EC2实例才能为DFS命名空间配置提供服务。

有关 DFS-N 配置的更多信息，请参阅 Microsoft [DFS文档中的命名空间概述](#)。有关使用DFS命名空间的更多信息，请观看上的“在 [Amazon 上使用DFS命名空间 for Windows 文件服务器](#)”视频。
YouTube

其他资源

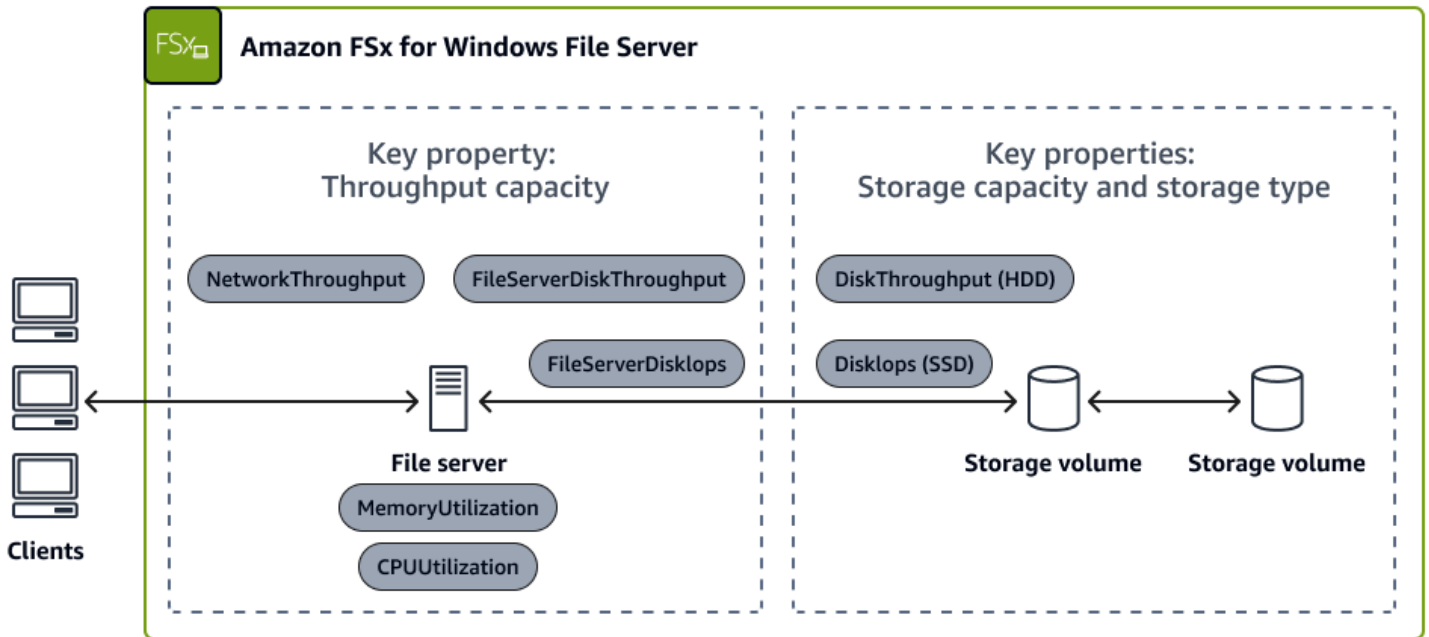
- [使用DFS命名空间对多个文件系统进行分组](#) (Ama FSx zon 文档)
- [演练 6：使用分片扩展性能](#) (Ama FSx zon 文档)
- 在 [Amazon 上使用DFS命名空间FSx适用于 Windows 文件服务器 \(实验室 \)](#) AWS

了解 Amazon 的HDD容量使用情况 FSx

概述

Amazon FSx for Windows File Server 提供了独立于文件系统容量选择吞吐量的灵活性。有两种容量设置可供选择：HDD（硬盘驱动器）和SSD（固态硬盘）。[EBSst1 驱动器](#)用于中的文件系统存储。HDD[EBSio1 驱动器](#)用于。SSD

下图显示了吞吐量和存储设置之间的关系。



对于HDD基于基础的存储，您将获得 12 的IOPS基准，其中有 80 个突发磁盘IOPS（IOPs每 TiB 的存储），吞吐量为 12 兆字节/秒，80个突发兆字节/秒（每 TiB 存储）。例如，如果您的共享大小为 50 TB，则可以获得 $50 * 12 = 600$ 作为吞吐量和的基准IOPS。

亚马逊版 Window FSx s 文件服务器提供 80 burst IOPS。当您的使用率低于基准率时，突发积分会自动充值，当您的使用率高于基准率时，会自动消耗突发积分。例如，如果您的工作负载在一小时内仅使用 10 IOPS /TB（比基准速率低 2 IOPS /TB），则可以在接下来的一个小时内使用 14 IOPS /TB（比基准高 2 IOPS /TB），然后再用完突发积分。

对于文件操作，Amazon f FSx or Windows File Server 在存储方面提供一致的亚毫秒延迟和SSD存储的个位数毫秒延迟。HDD对于所有文件系统，包括带有HDD存储空间的文件系统，Amazon FSx for Windows File Server 在文件服务器上提供了快速（内存中）缓存，因此无论存储类型如何，您都可以为主动访问的数据获得高性能和亚毫秒级的延迟。

在适当的情况下，HDD存储的使用可以帮助降低总体存储容量的成本，并为您的需求提供可靠的存储平台。

成本影响

Amazon FSx for Windows 文件服务器的性能取决于三个因素：存储容量、存储类型和吞吐量。网络 I/O 性能和内存中缓存大小完全由吞吐容量决定，而磁盘 I/O 性能则由吞吐容量、存储类型和存储容量的组合决定。

虽然SSD建议用于 I/O 密集型工作负载，但HDD性能规格可以满足各种工作负载的需求。HDD存储专为各种工作负载而设计，包括主目录、用户和部门共享以及内容管理系统。例如，如果您的用户只需要低延迟访问支持当前项目的数据，那么您存储的大部分数据都不会经常被访问。

您可以使用[AWS Pricing Calculator](#)将 20 TB 与中的HDD文件系统SSD进行比较us-east-1。如下表所示，即使不节省重复数据消除费用，在比较HDD文件系统和文件系统时，成本差异也很大。SSD

Amazon FSx 文件系统配置	每月费用
20 TB 多可用区 SSD () us-east-1	4,699.30 美元
20 TB 多可用区 HDD () us-east-1	542.88 美元
预计每月可节省的费用	4,156.42 美元

Note

FSx有关 Windows 文件服务器节省的其他费用，请参阅本指南的“[在 Amazon 中启用重复数据删除FSx](#)”部分。

通过正确识别您的性能需求，您可以为工作负载选择合适的存储并降低成本。

成本优化建议

如果您决定使用HDD存储，请测试您的文件系统以确保它可以满足您的性能要求。HDD与存储相比，SSD存储的成本较低，但磁盘吞吐量和IOPS每单位存储的磁盘较低。它可能适用于 I/O 要求较低的通用用户共享和主目录、不经常检索数据的大型内容管理系统或包含少量大文件的数据集。

现有文件系统的存储类型无法更改。要转换 Amazon FSx for Windows 文件服务器文件系统的存储类型，您必须备份现有文件系统并将其恢复到具有所需存储类型的新文件系统。如果您要将现有 SSD 文件系统转换为 HDD 文件系统，请注意其最低容量要高得多，HDD 为 2 TB。

要使用不同的存储类型恢复备份，请执行以下操作：

1. [备份您现有的文件系统](#)。
2. 使用 HDD 存储类型@@ [创建新 FSx 的 Amazon 文件系统](#)。
3. 将备份还原到具有所需存储类型的新文件系统。
4. 验证新文件系统的存储类型是否正确，并且您的数据完好无损。

在将更改转移到生产环境之前，我们建议您分析您的 Amazon FSx 文件系统的性能并确认更改是否可以接受。有关更多指导，请参阅 AWS 存储博客上的“[使用新指标优化 Amazon FSx for Windows 文件服务器性能](#)”一文。

其他资源

- 利用@@ [亚马逊优化成本 FSx](#) (亚马逊 FSx 文档)

使用单个可用区

概述

本节说明何时使用[亚马逊 FSx for Windows 文件服务器](#)的单一可用区实现更有利。它涵盖了迁移到单个可用区可以降低成本，同时仍允许您使用 FSx 适用于 Windows 的 Amazon File Server 作为托管文件存储服务的场景。我们建议您为生产工作负载实施单一的 Amazon FSx 可用区。这可以帮助确保您拥有多个可用区的冗余。

成本影响

与实施多个可用区相比，单个可用区文件系统可以降低大约 40% 的成本。使用多可用区文件系统，您每月支付每千兆字节 0.230 美元，每月每 GB 0.025 美元，SSD 而单个可用区文件系统每月的费用为每 GB 0.130 美元，HDD 而单个可用区文件系统的费用为 SSD 每 GB 每月 0.013 美元。HDD 您可以使用查看成本比较并创建自己的估算值[AWS Pricing Calculator](#)。

对于 10 TB 的文件系统，这可能与每月为多个可用区支付大约 1200 美元或为单个可用区每月支付 680 美元的差异。此[示例](#)使用 10 TB FSx 的 Windows 文件服务器文件和 SSD。据估计，重复数据删除可节省 50%。总体而言，单个可用区的入门成本较低，但有一些注意事项，将在下一节中介绍。

成本优化建议

单可用区部署

为确保单个可用区恰到好处，请考虑您自己的内部可用区，SLAs以便存储 Windows File Server 的数据。FSx这需要了解您是否必须SLAs向客户（内部和外部）提供商品，以及亚马逊FSx单一可用区的三个九个可用性是否仍然允许您满足这些SLAs需求。FSx对于具有单个可用区的 Windows 文件服务器，其正常运行时间仍为 99.9%。SLA对于亚马逊FSx来说，多个可用区域的比例大于 99.99%。对于任务关键型工作负载，我们建议您在单个可用区上使用多个可用区，即使需要支付额外费用。

单可用区部署非常适合SQL服务器数据库备份等工作负载。它们可以提供低成本的HDD分层存储，同时仍能为您提供稳定的正常运行时间。如果您需要为生产工作负载提供更高级别的可用性，例如高可用性SQL服务器或生产应用程序访问权限，那么单个可用区不适合您的工作负载。对于备份、非生产测试和开发环境，实施 Amazon FSx 单一可用区可以降低您的运营成本。

Amazon FSx 单可用区文件系统运行良好的一个用例是在生产环境中，使用多个 Amazon FSx 单可用区文件系统，作为使用 Always On 可用性组的高可用性SQL服务器集群中的每台服务器的存储。有关更多信息，请参阅 AWS 存储博客上[AWS发布的“优化高可用性SQL服务器部署的成本”](#)。

多区域复制

如果您想利用 Amazon FSx 的多区域复制，那么使用单个可用区文件系统（只有一个可用区文件系统可以工作的文件系统）来降低成本的一个潜在选择是。你可以部署支持使用原生 Microsoft DFS-R 的[单可用区文件系统](#)。DFS-R 能够在区域和多个站点之间自动复制数据。有关使用亚马逊配置 DFS-R 的更多信息FSx，请参阅亚马逊FSx文档中的[使用 Microsoft 分布式文件系统复制](#)。

另一种节省多区域成本的替代方案是使用 AWS Storage Gateway。这使您能够在另一个区域实施[亚马逊FSx文件网关](#)，以实现亚马逊FSx的多区域访问。有关更多信息，请参见本指南的[AWS Storage Gateway](#) 部分。

如果您跨区域工作，则必须考虑跨区域数据流量的数据传输成本。跨区域的流量会产生 0.02 美元/Gb 的费用。因此，如果您在高容量下保持一致的数据更改，这将增加您的总体成本。[例如](#)，1 TB 的数据传输等于大约 20.48 美元。

维护时段

如果您在 Amazon 上使用单一可用区，则维护时段是一个关键考虑因素FSx。在维护时段内，由于对底层 Windows 服务器进行例行软件修补，Amazon FSx 文件系统将在大约 20 分钟内不可用。如果您使用文件系统进行隔夜备份，请相应地调整 Amazon FSx 维护时段，以免备份期间出现中断。您可以在创建 Amazon FSx 文件系统后调整[维护](#)时段。

其他资源

- [可用性和持久性：单可用区和多可用区文件系统](#) (Ama FSx zon 文档)
- [亚马逊 Window FSx s 文件服务器定价](#) (AWS 网站)

AWS Storage Gateway

AWS Storage Gateway 是一种混合云存储服务，可将本地环境与 AWS 云存储连接起来。它允许您将现有的本地基础设施与无缝集成 AWS，使您能够从云中存储和检索数据，并在混合环境中运行应用程序。对于 Windows 工作负载，您可以使用 Storage Gateway 使用原生 Windows 协议（例如和）来存储 SMB 和访问数据 NFS。您可以使用 Storage Gateway AWS 通过使用本地硬件和软件作为通往云端的桥梁，来降低与运行 Windows 工作负载相关的成本。这使您无需对现有基础架构进行重大更改 AWS 即可利用的可扩展性和成本效益。

在 Storage Gateway 的保护下，您可以获得 Amazon S3 FSx 文件网关、亚马逊文件网关、磁带网关和卷网关。S3 文件网关和 FSx 文件网关最常用于 Microsoft 工作负载。

亚马逊 S3 文件网关

[Amazon S3 文件网关](#) 允许您将文件存储在 Amazon S3 中，同时使用传统 SMB 共享向用户提供访问权限。这提供了熟悉的用户界面，并通过将您的数据存储在 Amazon S3 中并利用各种 Amazon S3 存储层来帮助降低成本。您可以实施 Storage Gateway 和 S3 智能分层，以帮助您将生命周期文件自动移动到成本最低的存储层，从而进一步降低成本。我们建议 S3 File Gateway 用于横向扩展、只读访问、快速重复读取（从缓存）和数据库转储。对于高性能或高可用性写入、编辑文件或部门共享，通常不建议这样做。

Amazon FSx 文件网关

使用 [亚马逊 FSx Windows FSx 文件系统时](#)，[Amazon File Gateway](#) 还可以节省成本。您可以搭建 FSx 文件网关以提供对另一个地区的 Amazon FSx 文件系统的本地化访问权限，从而避免拥有两个独立文件系统的成本。如果您有多个本地文件服务器，并且想要整合这些服务器以避免为多台硬件设备付费，这也可能很有用。

成本影响

亚马逊 S3 文件网关

设置 S3 文件网关非常简单，因为您可以使用 Storage Gateway 的启动向导。通过在您的 AWS 环境中使用 EC2 实例，您可以在几分钟内部署网关。设置网关后，您可以将 Storage Gateway 共享配置为

可通过SMB和NFS协议进行访问。对于典型的 Windows 工作负载，您还可以使用此设置来利用 Active Directory 环境并设置文件共享权限。您可以有效地将 Storage Gateway 集成到正常使用中，因为它可以用作典型的 Windows 文件共享。文件和文件夹存储为对象，NTFS访问控制列表 (ACLs) 存储为元数据。

下表将 10 TB 存储的成本与三种可用存储选项进行了比较：

- FSx适用于 Windows 文件服务器
- 亚马逊 S3 文件网关
- 亚马逊 Elastic Block Store (亚马逊EBS)

如果您使用 Amazon S3，则存储 10 TB 存储空间的价格要低得多，因为您可以将数据分成不同的使用等级。在定价估算中，使用 S3 智能分层是因为其定价灵活性。这包括S3标准版中的80%，不频繁访问中的10%和Amazon S3 Glacier中的10%。尽管您可以使用 S3 Glacier，但重要的是要设置正确的生命周期规则，以确保无需立即访问任何移至 S3 Glacier 的文件。S3 Glacier 纯粹用于存档用途，而不是常规访问用途。

存储系统	10 TB 存储空间的成本	区域
FSx适用于 Windows 文件服务器 (假设重复数据删除可节省 50%)	683.20 美元 USD SSD	美国东部 (弗吉尼亚州北部)
亚马逊 S3 文件网关	449.51 美元智能分层 USD	美国东部 (弗吉尼亚州北部)
Amazon EBS	1,335.69 美元 USD GP3	美国东部 (弗吉尼亚州北部)

请考虑以下事项：

- 在 S3 Glacier 中，除非您使用将对象恢复[RestoreObject](#)API到 Amazon S3，否则您会收到一般 I/O 错误。我们建议您使用 Amazon Ev CloudWatch ents 来通知此 I/O 错误。这样，您的运营团队就可以对用户可能在需要访问的文件上遇到此错误做出反应。有关这些错误的更多信息，请参阅 Amazon S3 文件网关文档 [InaccessibleStorageClass](#) 中的[错误](#)：。
- 除了 S3 Glacier 的访问限制外，Storage Gateway 上[每个对象/文件夹只ACLs允许 10](#) 个。在决定使用 Storage Gateway 之前，请确保所需的ACL条目不超过 10 个。

Amazon FSx 文件网关

与 Amazon S3 文件网关类似，FSx 文件网关提供对可长期保留数据的文件系统的访问。在 Amazon S3 文件网关中，数据位于亚马逊 S3 中。对于 FSx 文件网关，您的数据驻留在 Windows FSx 文件服务器上。尽管多可用区选项可用 FSx 于 Windows 文件服务器，但没有多区域选项。如果您有一家跨国公司或远程办公室，则可能需要提供一个地理位置更接近最终用户的共享存储平台，以避免延迟。如果您要部署另一个 Amazon FSx 文件系统，则会增加全新 Amazon FSx for Windows 文件服务器文件系统的成本和必要的存储空间。为避免创建全新的文件系统和重复成本，您可以在辅助区域部署 FSx File Gateway。这为用户提供了对文件的本地化访问权限，同时有助于降低总体成本。

存储系统	10 TB 存储空间的成本	区域
FSx 适用于 Windows 文件服务器的亚马逊	683.20 美元 USD SSD	美国东部 (弗吉尼亚州北部)
Amazon FSx 文件网关	503.70 美元/单网关	美国东部 (弗吉尼亚州北部)

Note

上表中的价格基于 [Storage Gateway 的定价](#)。

记住以下内容：

- FSx File Gateway 可以帮助您每月为多区域工作负载节省大约 180 美元 (或每年 2100 美元)。
- FSx File Gateway 的数据传输费用要低得多，因为它只需要缓存定期访问的文件，而不是完整的辅助副本。
- 尽管您可以在不同的区域部署两个 Windows File Server，并使用 AWS Backup 或更新它们 AWS DataSync，但这两个选项都不是实时的。FSx

成本优化建议

亚马逊 S3 文件网关

S3 File Gateway 提供了一种低成本的存储文件选项，但是在如何实现和使用文件系统方面需要考虑一些问题。例如，S3 文件网关需要使用虚拟机才能运行 Storage Gateway 软件。在中 AWS，默认情况

下，Storage Gateway 是使用 m5.xlarge 实例在亚马逊EC2中部署的。如果您想降低本地存储成本，可以将 Storage Gateway 作为虚拟设备部署在虚拟化平台（例如VMware和 Hyper-V）上。

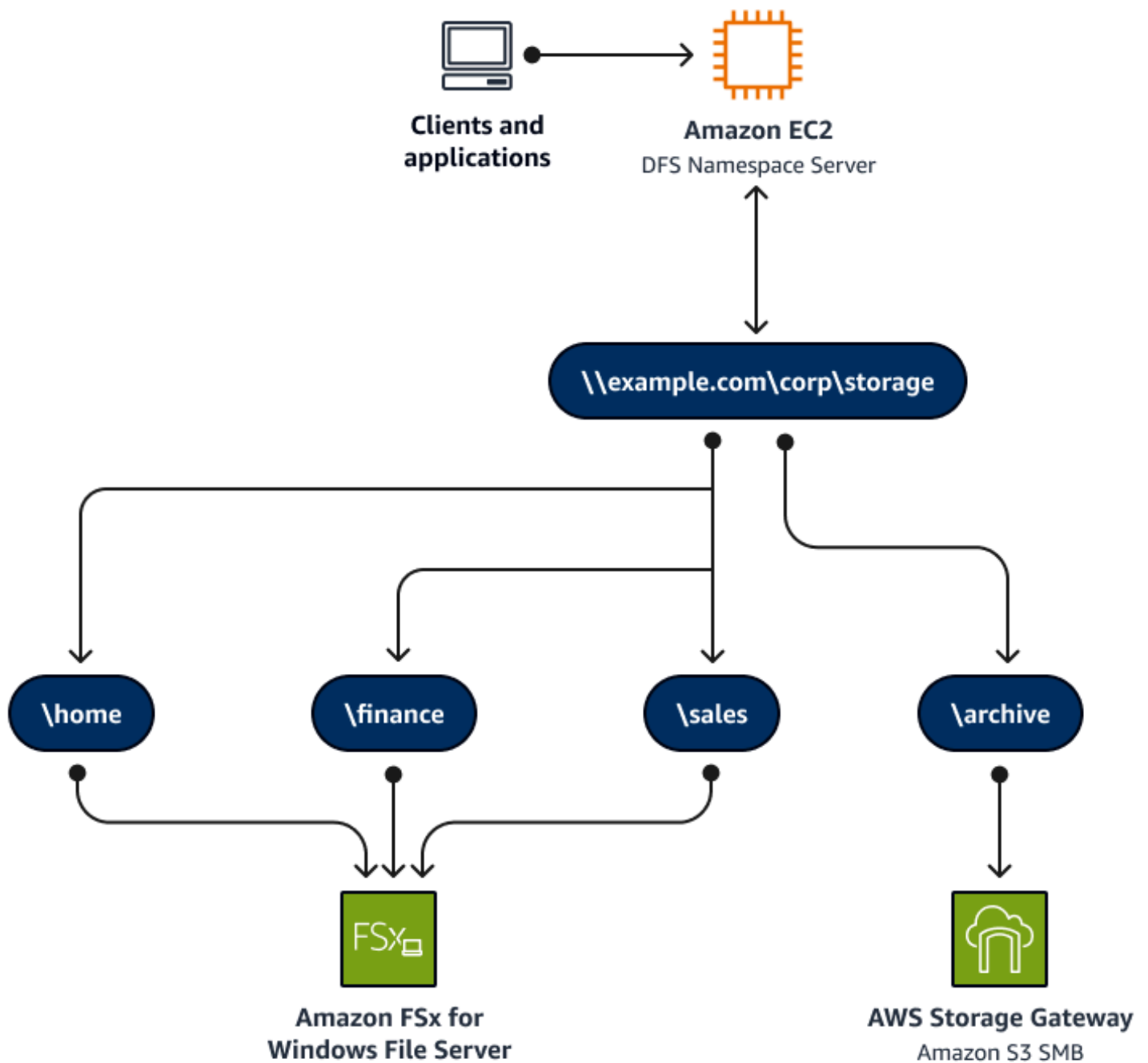
高可用性注意事项

运行 Storage Gateway 是访问文件的单点故障。为防止不必要的停机，我们建议您实施严格的访问控制，用户可以对其进行更改或停止和启动 Storage Gateway 实例。此外，对于上的部署 AWS，使用 Amazon Data Lifecycle Manager 创建路由快照以快速恢复您的 Storage Gateway 实施也是有益的。如果您使用在本地运行 Storage GatewayVMware，则可以对其进行配置以实现[高可用性](#)。

运行多个文件系统

将日常使用的文件工作负载与存档工作负载分开可以帮助您避免不必要的存储成本。Storage Gateway 能够与FSx适用于 Windows 的文件服务器文件系统一起部署。通过使用[DFS命名空间](#)，您可以呈现在 Windows 文件服务器上FSx运行的主要日常使用存储空间和在 Amazon S3 中运行的存储（可通过 Storage Gateway 访问）。

下图显示了如何使用单个DFS命名空间作为不同后端存储选项的前端接入点。



客户端会被定向到文件夹结构，例如\\example.com\storage。此主目录包含子目录。FSx适用于 Windows 的文件服务器文件系统包含正常访问的文件共享。您可以将在 Storage Gateway 上创建的文件共享用于存档数据。用户可以手动将项目存档到存档文件夹，也可以构建一个流程来自动将某些文件从普通文件共享移动到存档文件夹。

请考虑以下事项：

- 查看您的存储需求并为[缓存提供足够的存储空间](#)。
- 将您的网关添加到 Active Directory 配置中，然后使用[标准 Windows ACLs 来访问文件](#)。

FSx文件网关

FSx文件网关的部署与 S3 文件网关的部署类似，但如果使用启动向导，则会更容易。有关详细说明，请参阅[亚马逊FSx文件网关文档中的步骤 3：创建并激活](#)亚马逊FSx文件网关。在您的环境中部署 FSx File Gateway 后，您可以将其与现有 Amazon FSx 文件系统关联并访问您的文件。

部署FSx文件网关时，存储是首要考虑因素。默认存储空间提供 150 GB，这对于缓存文件来说是一个不错的空间。为可用空间不足创建监控警报有助于在不过度分配的情况下调整存储容量。

其他 资源

- [AWS Storage Gateway 资源](#) (AWS 文档)

Active Directory

运行 Windows Server 的亚马逊弹性计算云 (Amazon EC2) 是一个安全、可靠和高性能的环境，用于部署基于 Windows 的应用程序和工作负载。您可以快速配置实例，并根据需要扩大或缩小规模，同时只需为实际使用量付费。在 Windows 服务器环境中，Active Directory 服务用作身份管理的主要来源。

本节将介绍以下主题：

- [亚马逊 EC2 上自行管理的活动目录](#)
- [AWS Managed Microsoft AD](#)
- [AD Connector](#)

亚马逊 EC2 上自行管理的活动目录

概述

本节为降低在亚马逊弹性计算云 (Amazon EC2) 上运行 Active Directory 的成本提供了建议。主要重点是确保您可以适当地调整 Active Directory 域控制器的大小，并利用的 AWS Cloud 灵活性根据环境的需要进行调整。AWS 可以帮助您轻松停止实例并调整其大小以满足不断变化的需求，或者在扩展速度过快时缩小实例的大小。选择正确的实例大小和类型可以节省大量资金。

成本影响

下表显示了选择突发性能实例系列实例而不是通用实例之间的区别。这种选择每月可以为您节省大量资金。对您的实例进行适当的规划和规模可以帮助您管理成本。

实例类型	实例的数量	vCPU	内存	费用
t3a.medium	2	2	8	81.76 美元/月
m5a.large	2	2	8	每月 259.88 美元

有关成本的更多信息，请参阅 AWS Pricing Calculator [估算值](#)。

每月节省 178.12 美元，您的域控制器每年可节省超过 2,000 美元。请记住，在一个账户中只有两个域控制器的占用空间很小。如果有多个账户和额外的域控制器，这样的节省加起来可以显著降低成本。

成本优化建议

当您部署活动目录环境时，Microsoft 会提供[容量规划建议](#)。我们建议您在规划或扩展 Active Directory 环境时考虑以下主要组成部分：

- 内存
- 网络
- 存储
- 处理器

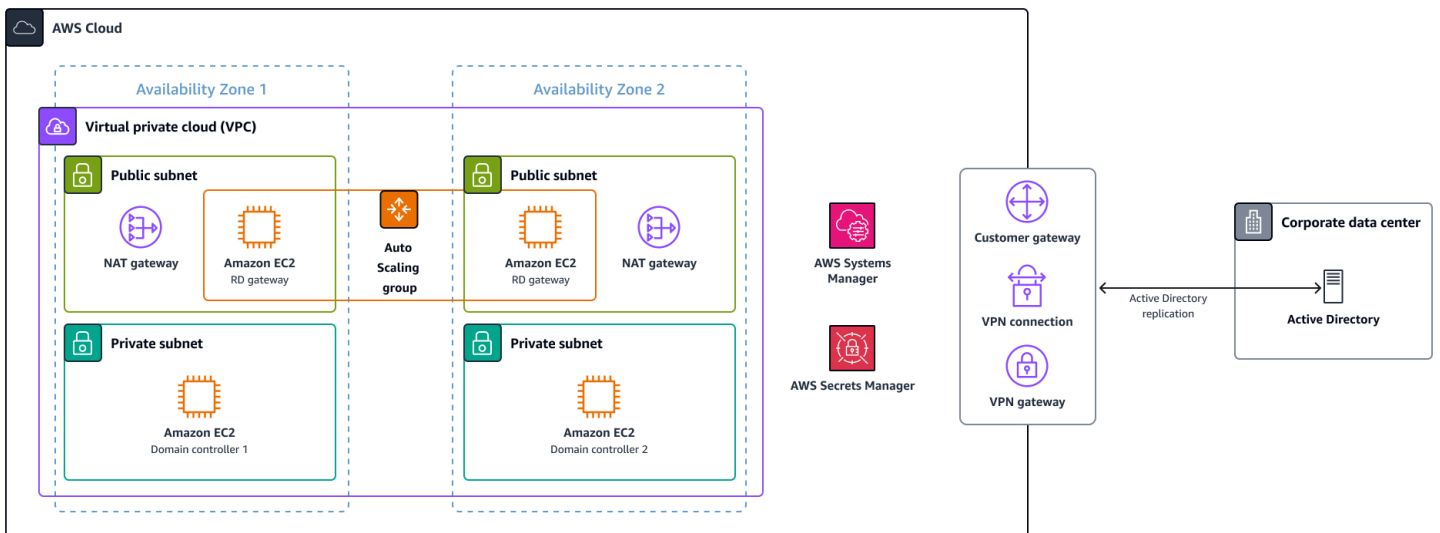
在记住这些主要组件的同时，您可以选择一种对您的 Active Directory 环境有意义的实例类型 AWS。本节介绍 AWS 部署场景的几个活动目录示例。这些场景清楚地表明，如果您不打算像在本地环境中 AWS 那样处理相同数量的用户和计算机，则没有必要在中复制本地环境。

下表重点介绍了与 vCPU、内存和磁盘有关的重要组件，以占用您的 AWS 占用空间。

组件	估计
存储/数据库大小	每位用户 40—60 KB
RAM	数据库大小 基本操作系统建议 第三方应用程序
网络	1 GB
CPU	每个内核有 1,000 个并发用户

混合部署场景

下图显示了 Active Directory 混合部署的示例架构。



如图所示，您通常拥有本地占地面积，然后将其扩展到 AWS Cloud。在迁移的初始阶段，通常不会部署所有用户和服务器 AWS。这就是为什么最初部署较小的占地面积以节省迁移费用很重要的原因。

如果您要保持本地占用空间，让服务器和用户在本地进行身份验证，那么您就不需要为域控制器使用相同的占用空间。AWS通过遵循 Active Directory 最佳实践，您可以实施适当的 [Active Directory 网站和服务](#)，对本地占用空间的用户和计算机进行身份验证，同时仅向中的 AWS域控制器验证您的 AWS 足迹。这使您能够 AWS 通过仅使用 AWS 资源而不是所有本地基础架构来避免 Active Directory 占用空间过大。有关设计混合设置的指南，请参阅 Microsoft 文档中的 [正确放置域控制器和站点注意事项](#)。

通过调整大小进行 AWS 迁移优化

如果您要为用户部署新的 Active Directory 实例，或者计划完全迁移到 AWS Active Directory 基础架构，我们建议您根据上表中实例选择的 microsoft 对 vCPU、内存和磁盘空间的建议来规划规模。

如果这是一个新的占用空间，你可以从小处着手，利用轻松 [更改实例类型的](#) 功能，随着环境的发展，调整环境的大小 AWS。本指南的“[亚马逊 EC2 上的 Windows](#)”部分向您展示了如何监控和查看您的 CPU 和内存使用率 AWS。这样，您就可以知道何时增加 EC2 实例的大小。

如果您要将本地 Active Directory 环境完全迁移到 AWS，则可以实施相同的规模调整计划以确保适当的性能。在复制本地内容之前 AWS，我们建议您完成对 Active Directory 环境的全面审查。这可以帮助您防止过度配置。请务必使用性能监控器来收集有关现有域控制器的流量和利用率的信息。这可以让你了解总体使用情况，这样你就可以调整规模并最终降低成本。

在“优化活动目录”AWS

如果您在上运行 Active Directory AWS，则还必须持续监控利用率并根据需要更改实例大小以减少开支。您可以使用 AWS Compute Optimizer 来获取有关您正在运行的资源的信息 AWS。有关使用

Compute Optimizer 调整您的 Windows 工作负载大小的信息，请参阅本指南的“[亚马逊 EC2 上的 Windows](#)”部分。要进行更全面的深入研究，您可以使用性能监控器来监控 Active Directory 域控制器的利用率，评估性能，然后相应地调整大小。

您还可以 CloudWatch 使用监控域控制器的性能。要优化您的域控制器（向上或向下扩展），您可以使用中提供的指标 CloudWatch 来帮助您做出正确的决策。您可以使用 CloudWatch 代理配置要发送以收集数据的自定义性能监控指标。有关说明，请参阅[如何使用 CloudWatch 代理在 Windows 服务器上查看性能监控器的指标？](#) 在 AWS 知识中心中。

部署 CloudWatch 代理后，可以在代理配置文件中的下配置以下指标 `metrics_collected`：

指标类别	指标名称
数据库到实例 (NTDSA)	数据库缓存% 命中
I/O 数据库读取平均延迟	
I/O 数据库读取量/秒	
I/O 日志写入平均延迟	
DirectoryServices (NTDS)	LDAP 绑定时间
DRA 待处理的复制操作	
DRA 待处理的复制同步	
DNS	递归查询次数/秒
递归查询失败/秒	
每秒收到的 TCP 查询次数	
每秒收到的查询总数	
每秒发送的响应总数	
收到的 UDP 查询/秒	
LogicalDisk	磁盘队列的平均长度

指标类别	指标名称
% 可用空间	
内存	已使用的已提交字节百分比
长期平均待机缓存寿命 (秒)	
网络接口	每秒发送的字节数
Bytes Received/sec	
当前带宽	
NTDS	ATQ 估计队列延迟
ATQ 请求延迟	
DS 目录读取/秒	
DS 目录搜索量/秒	
DS 目录写入次数/秒	
LDAP 客户端会话	
LDAP 搜索次数/秒	
LDAP 成功绑定次数/秒	
处理器	% 处理器时间
安全系统范围的统计数据	Kerberos 身份验证
NTLM 身份验证	

其他资源

- [Active Directory 域名服务 AWS : 合作伙伴解决方案部署指南](#) (AWS 文档)
- [Active Directory 域服务的容量规划](#) (微软文档)

- 在 [EC2 实例上运行 Active Directory 的设计注意事项](#) (AWS 白皮书)

AWS Managed Microsoft AD

概述

AWS Directory Service for Microsoft Active Directory，也称为 AWS Managed Microsoft AD，由 Windows 服务器 Active Directory 提供支持并由管理 AWS。您可以使用将各种支持 A AWS Managed Microsoft AD ctive Directory 的应用程序迁移到。AWS Cloud AWS Managed Microsoft AD 适用于各种本机 Active Directory 应用程序和服务。它还支持[AWS 托管应用程序和服务](#)。尽管 AWS Managed Microsoft AD 由于该服务及其计费机制，成本优化杠杆不多，但有一些设计原则可以帮助您将成本降至最低。

成本影响

由于 AWS Managed Microsoft AD 是基于当前 SKU 的托管服务，因此调整规模是一个相对简单的过程。目前有两种规模 SKU 可供选择：标准版和企业版。其他 SKU 包括目录共享、添加其他域控制器（包括其他区域）和跨区域数据传输。

成本优化建议

AWS Managed Microsoft AD 标准版和 AWS Managed Microsoft AD 企业版有区别。企业版最多支持 500,000 个 Active Directory 对象、125 个账户份额（软限制），并且支持多区域。标准版最多支持 30,000 个 Active Directory 对象、五个账户共享（软限制为最大约 30 个），并且不支持多区域。

在选择目录类型之前需要考虑的问题是：

- 是否需要多区域支持？
- 该目录是否会与 30 多个账户共享？
- Active Directory 对象数量会超过 30,000 吗？

如果以上任何一个问题的答案为是，则需要使用企业版。如果所有问题的答案都是否定的，我们建议您从标准版开始。

Note

您可以将目录从标准版升级到企业版，但目录无法降级。部署标准版并不是单向的。如果您想将目录升级到企业版，请联系 AWS。

在 AWS Managed Microsoft AD 企业版中共享目录时，每次共享都需要付费。这低于在每个账户中部署目录的成本，但请记住，如果不加以控制，共享成本可能会攀升。我们建议您仅与包含亚马逊关系数据库服务 (Amazon RDS) 和适用于 Windows File Server 的 Amazon FSx 的账户共享目录，因为只有这些服务支持此功能。请记住，你可以选择将 FSx for Windows File Server 与自行管理的 Active Directory 集成，包括。AWS Managed Microsoft AD 如果其他账户只需要使用 Amazon FSx，则无需共享目录即可针对 AWS Managed Microsoft AD 该账户进行自我管理的 Amazon FSx 部署。

在决定何时部署其他域控制器时，请记住，在同一 VPC 的不同可用区中仅 AWS Managed Microsoft AD 支持两个子网。添加其他域控制器不允许您添加其他子网。要确定是否由于性能问题而必须添加其他域控制器，请查看[中的域控制器性能指标 CloudWatch](#)。这会告诉您是否有一个或所有域控制器不堪重负。如果您确定只有一个域控制器不堪重负，那么添加额外的域控制器并不能减轻负载，您需要更深入地研究没有在当前可用的域控制器之间进行负载平衡的应用程序。如果所有域控制器都被大量使用，则添加额外的域控制器可以减少现有域控制器的负载。有关如何自动扩展的说明，请参阅 AWS 安全博客中的[如何根据利用率指标自动 AWS Managed Microsoft AD 扩展](#)。

如果您将目录扩展到多个区域，我们建议您不要使用 NETLOGON 或 SYSVOL 共享的目录来存储文件。所有域控制器都会复制这些共享的内容。不使用共享文件存储可以将数据传输成本降至最低。

您还可以选择通过注册企业协议 AWS。企业协议使您可以选择定制最适合自己需求的协议。有关更多信息，请参阅[企业客户](#)。

其他资源

- [AWS Managed Microsoft AD 配额](#) (AWS Directory Service 文档)
- [AWS Directory Service 定价](#) (AWS 网站)
- [活动目录域名服务 AWS](#) (AWS 白皮书)

AD Connector

概述

[AD Connector](#) 是一项代理服务，它提供了一种简单的方法，可以将你现有的本地 Microsoft Active Directory 连接到兼容的 [AWS 应用程序](#)，例如亚马逊 WorkSpaces QuickSight、亚马逊，以及亚马逊弹性计算云 (Amazon EC2) 实例的无缝域加入，而无需在云中缓存任何信息。您可以使用 AD Connector 将一个服务帐户添加到您的活动目录。AD Connector 无需进行目录同步，也无需托管联合基础架构的成本和复杂性。尽管由于服务的性质及其计费机制的原因，AD Connector 的成本优化杠杆并不多，但您可以按照本节中的设计建议将成本降至最低。

成本影响

AD Connector 是一项基于预设 SKU 的托管服务。这使得调整大小成为一个简单的过程。有两种尺寸 SKU 可供选择：小号和大号。您可以使用来估算涉及 AD [AWS Pricing Calculator](#) Connector 的成本。

成本优化建议

除了后端计算资源之外，小型和大型连接器大小没有区别。

在选择目录类型之前需要考虑的问题是：

- 是否有大量（10,000 多个）活跃用户在使用与 AD Connector 集成的 AWS 应用程序？
- 用户是多个、深层嵌套或循环嵌套组的成员？

如果两个问题的答案都是否定的，我们建议您从小号开始。如果你对以上任何一个问题的回答为是，那么大尺寸可能值得考虑。您可以从小尺寸的 AD Connector 开始，如果目录因性能而受损，则可以请求将该目录升级到大容量。

Note

您可以将 AD 连接器从小升级到大型，但是 AD 连接器无法降级。

大多数性能问题与 AD Connector 无关，而是由于许多用户是许多、深层或循环嵌套群组的成员，本地 Active Directory 域控制器不堪重负。

您还可以选择通过注册企业协议 AWS。企业协议使您可以选择定制最适合自己需求的协议。有关更多信息，请参阅 [企业客户](#)。

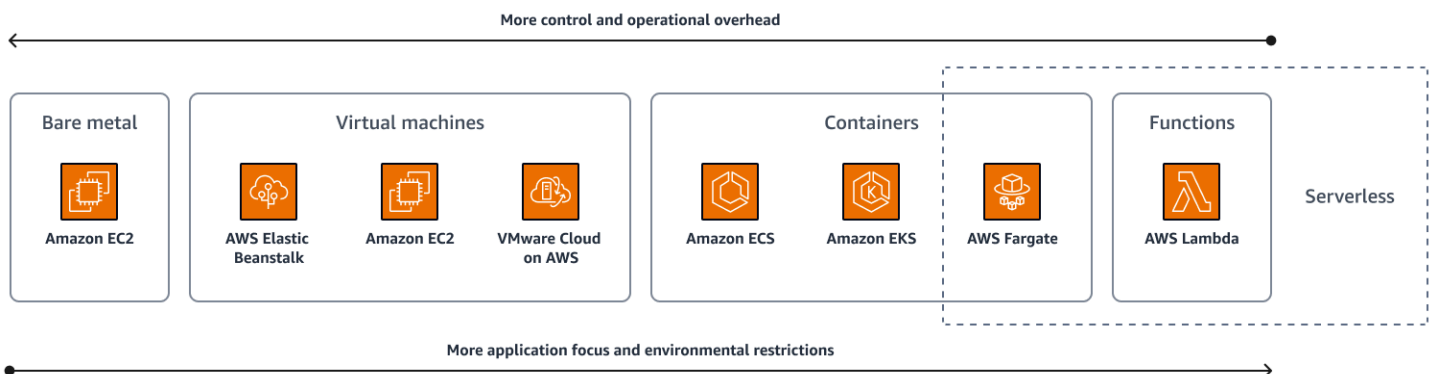
其他资源

- [AD Connector 配额](#) (AWS Directory Service 文档)
- [其他目录类型定价](#) (AWS 网站)
- [活动目录域名服务 AWS](#) (AWS 白皮书)

.NET

开发和部署。 .NET 应用程序是帮助您实现云计算所提供的规模和敏捷性的重要关键。对于许多遗产。 .NET 应用程序，在中运行应用程序的最合适计算选择 AWS 是通过 AWS Elastic Beanstalk 或亚马逊弹性计算云 (Amazon EC2) 使用虚拟机。也可以跑步。 .NET Windows 和 Linux 容器中的应用程序。

的介绍 .NET Core 使您能够进行现代设计。 .NET 充分利用云端所有优势的应用程序。现代应用程序可以使用传统的计算选择集，也可以针对各种类型的无服务器环境，包括 AWS Fargate 或 AWS Lambda。 .NET 6+ 现在在 G EC2 Graviton2 系列等 ARM64 EC2 实例上提供高性能的工作负载托管。这使您可以访问亚马逊上可用的最新一代处理器 EC2。这意味着您的应用程序可以托管在专门用于您的工作负载类型的计算上，例如视频编码、Web 服务器和高性能计算 (HPC)。



本节提供了一些建议，以帮助您调整自己的 .NET 利用云优势的应用程序，重点是成本效益。

本节将介绍以下主题：

- [重构为现代。 .NET 然后迁移到 Linux](#)
- [容器化。 .NET 应用程序](#)
- [使用 Graviton 实例和容器](#)
- [Support 支持静态的动态缩放。 .NET 框架应用程序](#)
- [使用缓存来减少数据库需求](#)
- [以无服务器为例。 .NET](#)
- [考虑专门构建的数据库](#)

重构为现代。NET然后迁移到 Linux

概述

对遗产进行现代化改造。NET框架应用程序可以帮助您提高安全性、性能和可扩展性。实现现代化的有效方法。NET框架应用程序是将它们迁移到现代的。NET版本 (6+)。以下是将这些应用程序迁移到开源的一些主要好处。NET:

- 通过在 Linux 操作系统上运行 Windows 来降低许可成本
- 充分利用现代语言的可用性
- 获得经过优化的性能，可在 Linux 上运行

许多组织仍在运行旧版本的。NET框架。这可能会带来安全风险，因为Microsoft不再解决旧版本中的漏洞。Microsoft 已终止对最新版本的支持。NET框架 4.5.2、4.6 和 4.6.1。评估继续运行旧版本框架的风险和收益非常重要。为了降低风险和降低成本，值得花时间和精力重构到的现代版本。NET。

成本影响

以通用EC2实例类型 (m5) 为例，它提供了计算、内存和网络资源的平衡。这些实例适用于各种应用程序，例如 Web 服务器、中型数据库和源代码存储库。

例如，美国东部 (弗吉尼亚北部) 的 Windows Server (含许可证) 上内存为 4 vCPUs 和 16 GB 的按需 m5.xlarge 实例每月花费 274.48 美元。在 Linux 服务器上使用同样的资源每月花费 140.16 美元。在此示例中，当您从中迁移应用程序时，成本降低了49%。NET框架改为现代版本。NET然后在 Linux 服务器上运行您的应用程序。[您的费用可能会有所不同，具体取决于您在选择实例时选择的选项 \(例如，实例类型、操作系统、存储\)](#)。EC2您可以使用 Savings Plans 或[预留实例](#)进一步优化成本。有关更多详细信息，请使用[AWS Pricing Calculator](#)进行成本估算。对于包含 Windows 的实例，无论定价模式如何，许可成本均为[每 v CPU 每小时 0.046 美元](#)。

移植这些。NET框架应用程序到现代。NET需要开发人员的努力。您必须评估您的应用程序及其依赖关系，以查看它们是否与目标平台版本兼容。AWS 的@@ [移植助手](#)。NET是一款扫描辅助工具。NET构架应用程序并生成.NET兼容性评估，帮助您更快地移植应用程序，使其与 Linux 兼容。的移植助手.NET标识与的不兼容之处。NET，查找已知的替代品，并生成详细的兼容性评估。移植解决方案后，必须手动更改代码，才能使用依赖项成功编译项目。这样可以减少将应用程序现代化为 Linux 所涉及的手动工作。如果您的应用程序支持ARM处理器，则迁移到 Linux 可以解锁使用 Graviton 实例的能力。这可以帮助您进一步降低20%的成本。有关更多信息，请参阅[通电](#)。[NET5 使用 AWS Graviton2：计算博客中的基准测试](#)。AWS

还有其他工具，例如[用于的AWS 工具包](#)、[NET重构](#)和[.NET升级助手](#)，它可以帮助您移植旧版。NET 框架应用程序到现代。NET。

成本优化建议

要迁移。NET框架应用程序，请执行以下操作：

1. 先决条件-要使用移植助手。NET，你必须安装。NET在您计划分析应用程序源代码的计算机上 5+。计算机上的资源必须至少具有 1.8 的GHz处理速度、4 GB 的内存和 5 Gb 的存储空间。有关更多信息，请参阅移植助手中的[先决条件](#)。NET文档。
2. 评估 — 下载移植助手.NET作为[可执行](#)（下载）文件。您可以将该工具下载并安装到您的计算机上，以开始评估您的应用程序。评估页面包含移植的项目、软件包APIs，这些项目和包与现代版本不兼容。NET。因此，评估结束后，解决方案中会出现构建错误。您可以查看评估结果或将其下载到CSV文件中。有关更多信息，请参阅“移植助手”中的移植[解决方案](#)。NET文档。
3. 重构 — 评估应用程序后，您可以将项目移植到目标框架版本。移植解决方案时，移植助手将修改您的项目文件和部分代码。您可以查看日志，查看源代码的更改。在大多数情况下，代码需要付出额外的努力才能完成迁移和测试，从而为生产做好准备。根据应用程序的不同，某些更改可能包括实体框架、身份和身份验证。有关更多信息，请参阅“移植助手”中的移植[解决方案](#)。NET文档。

这是将应用程序现代化为容器的第一步。可能有许多业务和技术驱动因素可以帮助您实现现代化。NETLinux 容器的框架应用程序。其中一个重要的驱动因素是通过从 Windows 操作系统迁移到 Linux 来降低总拥有成本。在将应用程序迁移到的跨平台版本时，这可以降低许可成本。NET以及用于优化资源利用率的容器。

将应用程序移植到 Linux 后，您可以使用[AWS App2Container](#)对应用程序进行容器化。App2Container 使用ECS亚马逊或亚马逊EKS作为终端节点服务，您可以直接部署到这些服务。App2Container 提供了所有必要的基础设施即代码 (IaC) 部署工件，可重复对您的应用程序进行容器化。

其他注意事项和资源

- 如果您有基于 VB 构建的应用程序。NET (2002 年的遗留框架)，并希望将它们移植到。NET6、参见 [Port legacy VB。NET申请到。NET6.0 带有“移植助手”的功能](#)。[NET](#)在 Microsoft 工作负载 AWS 博客上发帖。
- 如果你在 Windows Communication Foundation (WCF) 上有旧版应用程序，并且想在现代版上运行它们。NET，你可以采用酷睿WCF。有关更多信息，请参阅[WCF使用移植助手将旧版WCF应用程序现代化为核心版](#)。[NET](#)在 Microsoft 工作负载 AWS 博客上发帖。

- 您可以将移植助手作为扩展添加到 Visual Studio 中 IDE。这使您能够执行转换代码所需的所有任务，而无需在您的 IDE 和移植助手之间切换。NET 工具。有关更多信息，请参阅[加速。NET 使用移植助手实现应用程序现代化。NET 微软工作负载 AWS 博客上的 Visual Studio IDE 扩展帖子](#)。
- AWS 的[@@ 移植助手。NET 现在是开源工具](#)，包含评估的源代码和兼容性分析组件。这可以鼓励您的开发者使用和共享。NET 移植知识和最佳实践。
- 你可以移植。NET 框架应用程序到现代。NET 在 Linux 上，使用 AWS 工具包适用于。NET 重构。有关更多信息，请参阅[加速。NET 使用 AWS Toolkit 实现现代化。NET 在 Microsoft 工作负载博客上重构帖子](#)。AWS
- 您可以[加快容器化和迁移。ASP.NET 要 AWS 使用的核心应用程序 AWS App2Container](#)。

容器化。NET 应用程序

概述

容器是一种以一致且可复制的方式打包和部署应用程序的轻量级高效方法。本节介绍如何使用 AWS Fargate 无服务器容器服务来降低您的成本。NET 应用程序，同时还提供可扩展和可靠的基础架构。

成本影响

影响使用容器节省成本的有效性的一些因素包括应用程序的大小和复杂性、需要部署的应用程序数量以及应用程序的流量和需求水平。对于小型或简单的应用程序，与传统的基础设施方法相比，容器可能无法显著节省成本，因为管理容器和相关服务的开销实际上可能会增加成本。但是，对于更大或更复杂的应用程序，使用容器可以提高资源利用率和减少所需实例的数量，从而节省成本。

为了节省成本，我们建议您在容器化时记住以下几点：

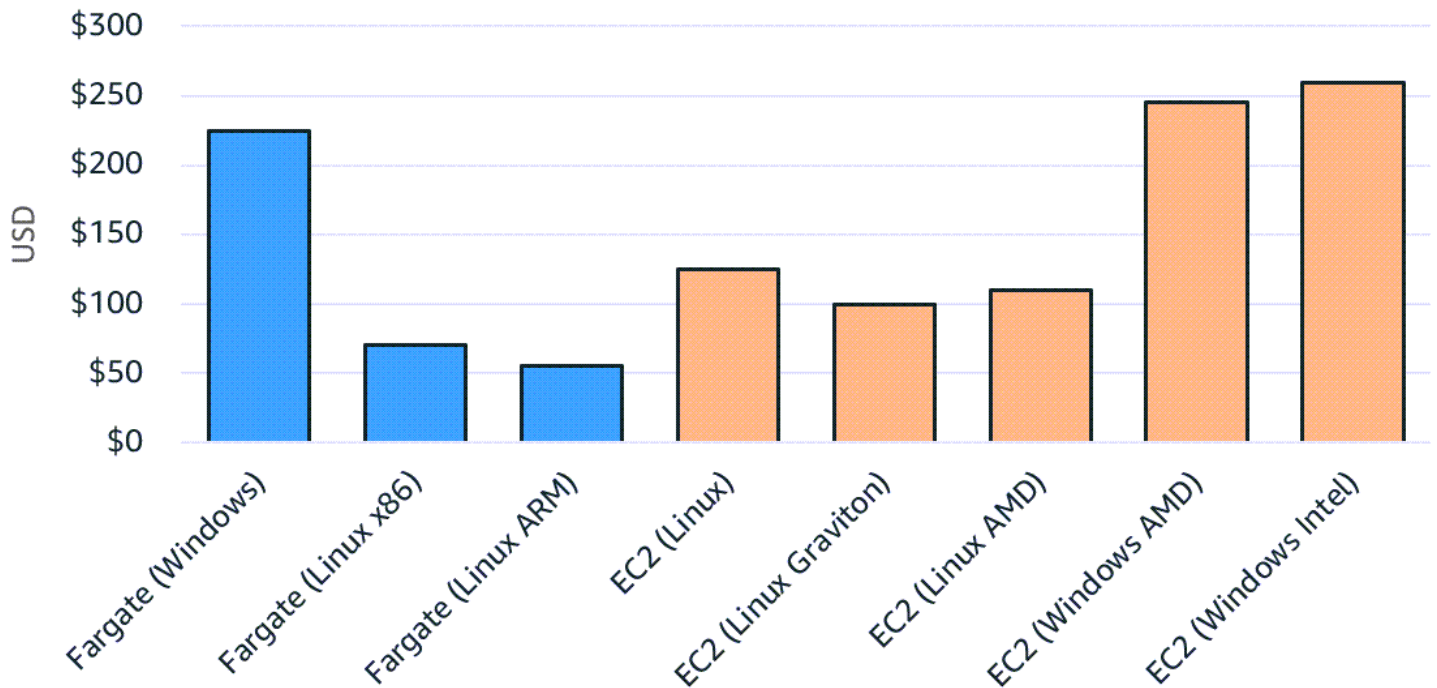
- 应用程序规模和复杂性 — 更大、更复杂的应用程序更适合容器化，因为它们往往需要更多的资源，并且可以从提高的资源利用率中受益更多。
- 应用程序数量 — 您的组织必须部署的应用程序越多，通过容器化节省的成本就越多。
- 流量和需求 — 具有高流量和高需求的应用程序可以从容器提供的可扩展性和弹性中受益。这可以节省成本。

不同的架构和操作系统会影响容器成本。如果您使用的是 Windows 容器，则出于许可方面的考虑，成本可能不会降低。Linux 容器的许可成本较低或不存在。下图使用了美国东部（俄亥俄州）地区的基本配置，设置如下：每月 30 个任务运行 12 小时，分配了 4 GB vCPUs 和 8 GB 的内存。AWS Fargate

您可以选择两个主要的计算平台来运行您的容器 AWS：[EC2 基于容器的主机和无服务器或 AWS Fargate](#)。如果您使用亚马逊弹性容器服务 (Amazon ECS) 而不是 Fargate，则必须保持计算（实例）的运行，以允许放置引擎在需要时实例化容器。如果您改用 Fargate，则仅配置所需的计算容量。

下图显示了使用 Fargate 的等效容器与使用 Amazon 的容器的区别。EC2 由于 Fargate 的灵活性，应用程序的任务每天可以运行 12 个小时，而非工作时间的利用率为零。但是，对于亚马逊 ECS，您必须使用 [Auto Scaling EC2 实例组](#) 来控制计算容量。这可能导致容量每天 24 小时运行，最终会增加成本。

Monthly costs of Fargate and Amazon EC2



成本优化建议

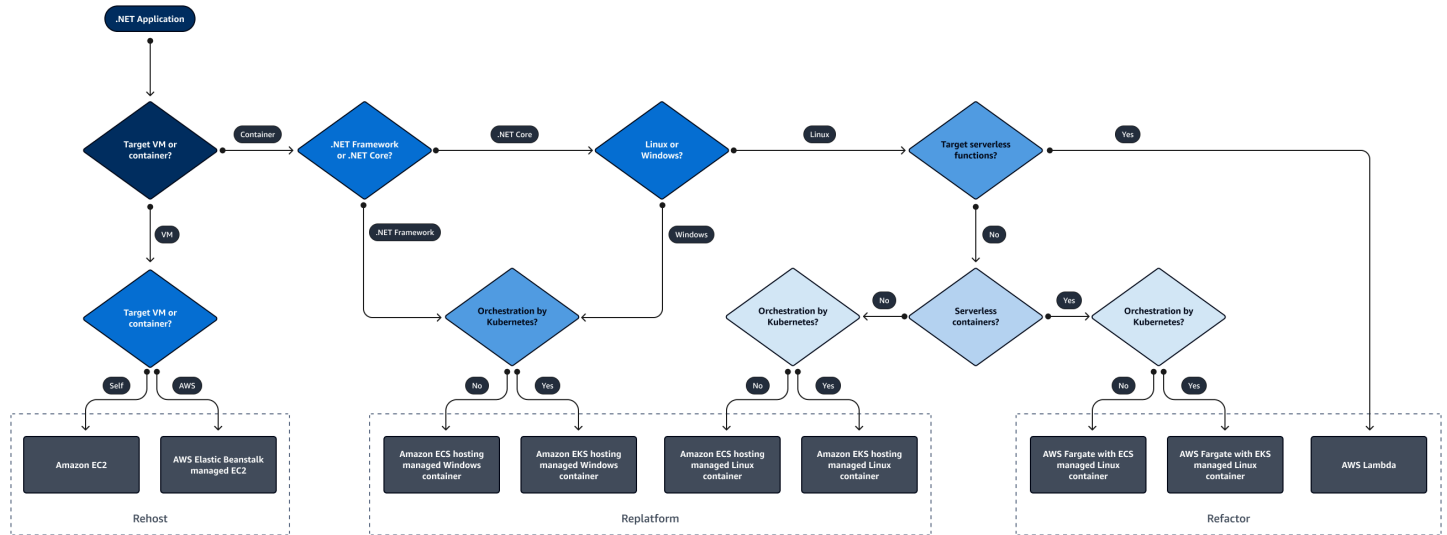
使用 Linux 容器而不是 Windows

如果您使用 Linux 容器而不是 Windows 容器，则可以节省大量资金。例如，如果您运行 .NET 应用程序，则可以节省大约 45% 的计算成本。NET 在 EC2 Linux 上运行核心，而不是运行 .NET 在 EC2 Windows 上的框架。如果您使用 ARM 架构（AWS Graviton）而不是 x86，则可以额外节省 40%。

如果您计划为现有容器运行基于 Linux 的容器。NET 框架应用程序，您必须将这些应用程序移植到现代的跨平台版本中。NET（[比如。NET 6.0](#)）以便使用 Linux 容器。一个主要的考虑因素是权衡重构成本与通过降低 Linux 容器成本而节省的成本。有关将应用程序移植到现代版本的更多信息。NET，请参阅 [“移植助手”](#) 了解相关信息。NET 在 AWS 文档中。

迁移到现代化的另一个好处。NET (也就是说，远离 .NET 框架) 是可以获得更多的现代化机会。例如，您可以考虑将应用程序重新架构为基于微服务的架构，该架构具有更高的可扩展性、灵活性和成本效益。

下图说明了探索现代化机会的决策过程。



利用 Savings Plans

容器可以帮助您利用 [Compute Savings Plans](#) 来降低 Fargate 成本。灵活的 discount 模式提供的折扣与可转换预留实例相同。Fargate 定价基于从您开始下载容器映像到亚马逊 ECS 任务终止 (四舍五入到最接近的秒数) 所使用的 v CPU 和内存资源。[Fargate](#) 的 Savings Plans 可为 Fargate 的使用量节省高达 50%，以换取承诺在一年或三年内使用特定数量的计算使用量 (以每小时美元计)。您可以使用 [AWS Cost Explorer](#) 来帮助您选择 Savings Plan。

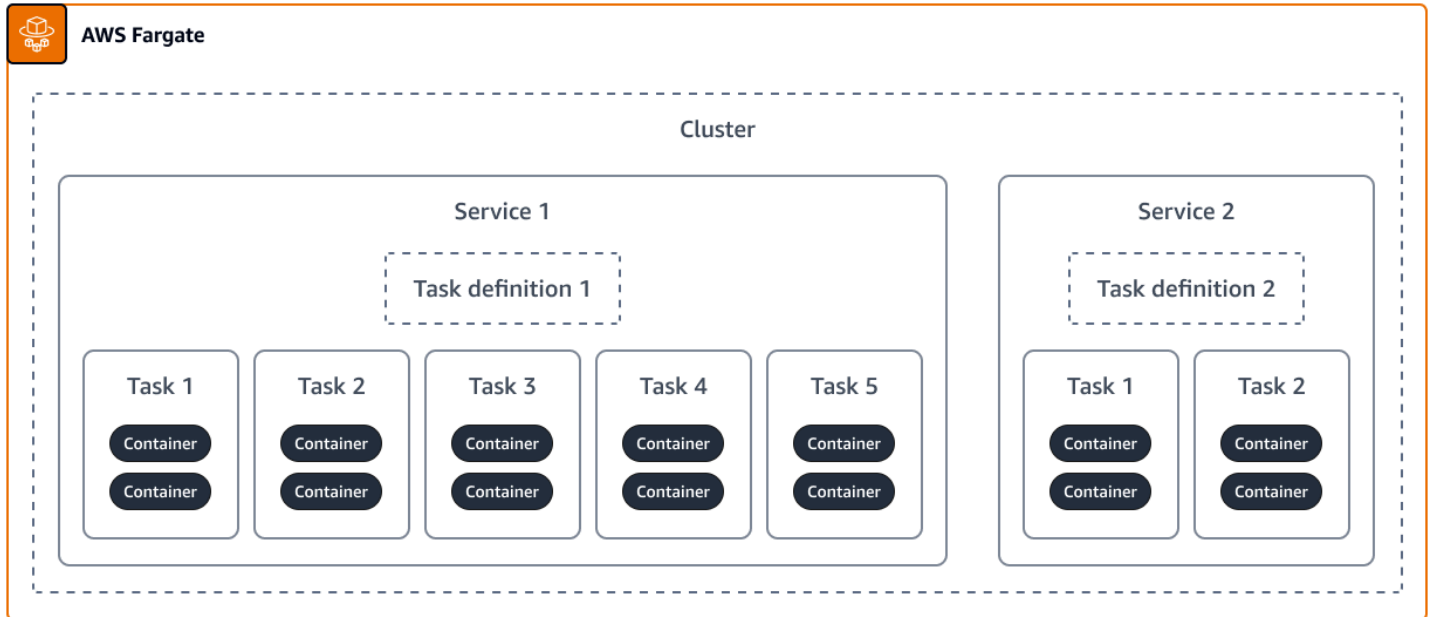
重要的是要明白，Compute Savings Plans 适用于首先为您节省最大成本的使用量。例如，如果你在云中运行一个 t3.medium Linux 实例 us-east-2 和一个相同的 Windows t3.medium 实例，那么 Linux 实例将首先获得 Savings Plan 的好处。这是因为 Linux 实例具有节省 50% 的潜力，而同一 Windows 实例的节约潜力为 35%。如果您有其他符合储蓄计划条件的资源 AWS 账户，例如亚马逊 EC2 或 Lambda，则无需先将您的储蓄计划应用于 Fargate。有关更多信息，请参阅本指南的 [储蓄计划文档中的了解储蓄计划如何适用于您的 AWS 使用量](#)，以及本指南的“在 [亚马逊上优化 Windows 上的支出 EC2](#)”部分。

大小合适的 Fargate 任务

必须确保正确调整 Fargate 任务的大小，以实现最大程度的成本优化。通常，开发人员在最初确定其应用程序中使用的 Fargate 任务的配置时，并不掌握所有必要的使用信息。这可能导致任务过度配置，进而导致不必要的支出。为避免这种情况，我们建议您对在 Fargate 上运行的应用程序进行负载测试，

以了解特定任务配置在不同使用场景下的表现。您可以使用负载测试结果 v CPU、任务的内存分配和 auto Scaling 策略在性能和成本之间找到适当的平衡。

下图显示了 Compute Optimizer 如何针对最佳任务和容器大小生成建议。



一种方法是使用负载测试工具（例如[上 AWS 的“分布式负载测试”](#)中描述的工具）来建立 v CPU 和内存利用率的基准。运行负载测试以模拟典型的应用程序负载后，您可以微调任务的 v CPU 和内存配置，直到达到基准利用率。

其他资源

- [Amazon ECS 和 AWS Fargate \(AWS 容器博客文章 \) 的成本优化清单](#)
- [亚马逊的理论成本优化ECS发布类型：Fargate vs EC2 \(AWS 容器博客文章 \)](#)
- [的@@ 移植助手。NET \(AWS 文档 \)](#)
- [分布式负载测试 AWS \(AWS 解决方案库 \)](#)
- [AWS Compute Optimizer 在 AWS Fargate \(AWS 云财务管理博客文章 \) 上推出对亚马逊ECS服务的支持](#)

使用 Graviton 实例和容器

概述

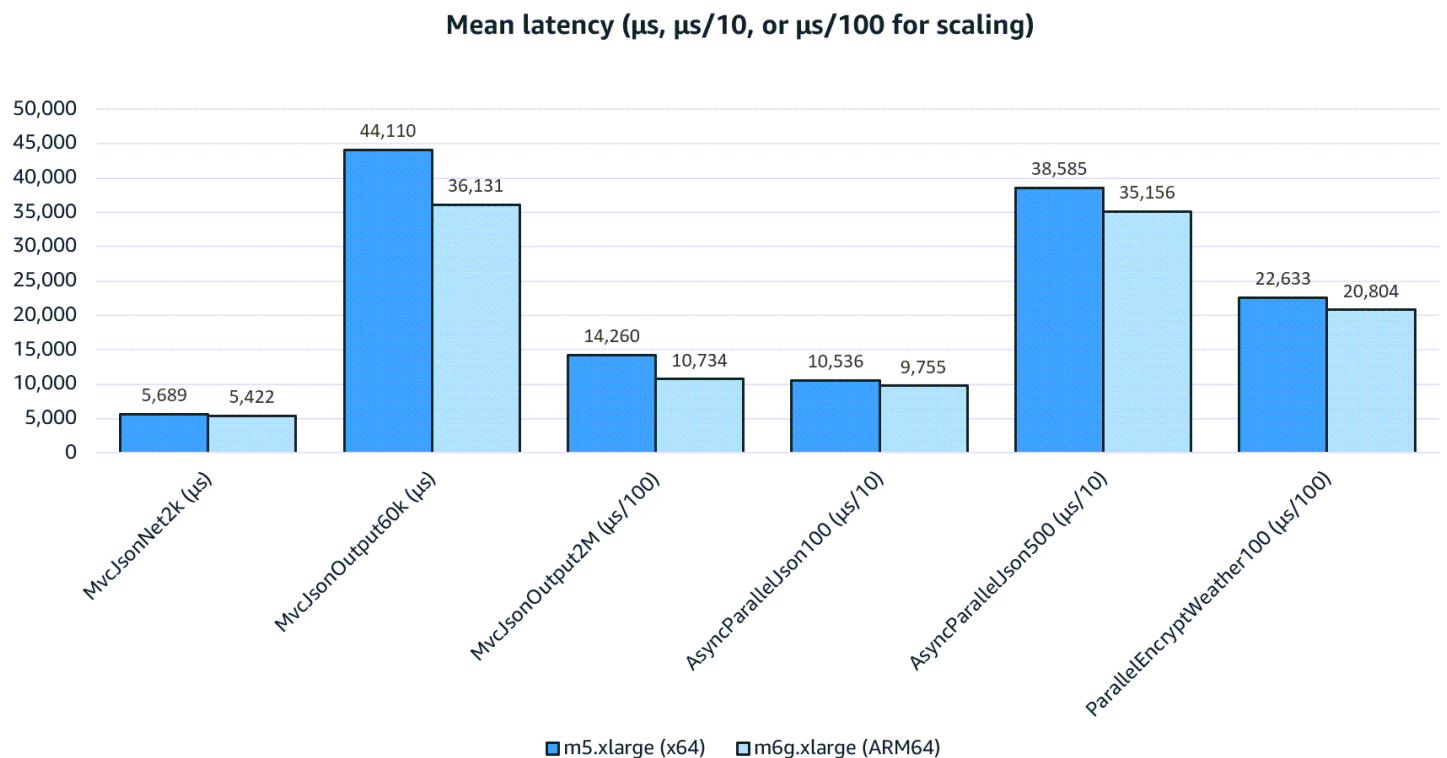
AWS Graviton 实例由设计的ARM处理器提供支持，旨在为在亚马逊弹性计算云 (Amazon) 中运行的云工作负载（包括在其中运行的容器EC2）提供最佳的性价比。AWS目前，亚马逊上有三代 Graviton 可供使用。EC2本指南重点介绍如何使用 Graviton 2 和 3。NET应用程序，因为使用最新版本的 Graviton 可以节省大量成本。请记住，Graviton 实例只能运行 Linux 操作系统。因此，Graviton 实例是一款强劲的产品。NET在 Linux 上运行，但不适用于 Windows 操作系统或传统操作系统。NET 框架应用程序。

与同类EC2实例相比，Graviton 3的效率提高了60%，性能提高了40%。本指南重点介绍使用Graviton 的成本效益，但值得注意的是，Graviton还提供了性能改进和环境可持续性改善的额外好处。

成本影响

当你切换到Graviton时，你最多可以节省45%。在你重构任何遗产之后。NET将应用程序框架化为现代.NET版本，你可以解锁使用 Graviton 实例的功能。迁移到Graviton是一种有效的成本优化技术。NET开发者。

下表中的示例显示了通过迁移到 Graviton 实例可能实现的性能改进。



有关上图中用于创建结果的基准测试方法的完整细分和解释，请参阅 [Powering 。NET5 使用 AWS Graviton2：计算博客中的基准测试](#)。AWS

效率提高的原因之一是 x86 和 Graviton CPU 之间 v 的含义不同。在 x86 架构中，v CPU 是通过超线程实现的逻辑内核。在 Graviton 中，v CPU 等同于物理核心，它允许 v CPU 完全投入工作负载。

Graviton2的结果是，与同类x86/x64实例相比，其性价比提高了40%。与 Graviton2 相比，Graviton3 提供了以下功能：

- 性能得到提高，性能提高了 25%
- 浮点性能最高可提高两倍
- 加密工作负载性能最高可提高两倍
- 机器学习性能提高多达三倍

此外，Graviton3是云中第一个以内存为特色的实例。DDR5

下表显示了基于 Graviton 的实例和基于 x86 的同等实例在成本节省方面的差异。

下表显示Graviton节省了19.20%。

实例类型	架构	v CPU	内存 (GB)	每小时成本 (按需)
t4g.xlarge	ARM	4	16	0.1344 美元
t3.xlarge	x86	4	16	0.1664 美元

下表显示Graviton节省了14.99%。

实例类型	架构	v CPU	内存 (GB)	每小时成本 (按需)
c7g.4xlarge	ARM	16	32	0.5781 美元
c6i.4xlarge	x86	16	32	0.6800 美元

在考虑 Graviton 时，测试应用程序的性能状况非常重要。Graviton 不能取代扎实的软件开发实践。您可以使用测试来验证是否充分利用了底层计算资源。

成本优化建议

有几种方法可以利用 Graviton 处理器/实例。本节将引导您完成从使用 x86 架构计算机迁移到 Graviton () 实例所需的更改。ARM

在 Lambda 中更改运行时设置

我们建议您在中切换运行时设置 AWS Lambda。有关更多信息，请参阅 Lambda [文档中的修改运行时环境](#)。NET 是一种编译语言，您必须遵循构建过程才能使其正常工作。有关如何执行此操作的示例，请参阅 [NET 在 Graviton 上](#)。GitHub

容器

对于容器化工作负载，请创建多架构容器镜像。您可以通过在 Docker build 命令中指定多个架构来做到这一点。例如：

```
docker buildx build -t "myImageName:latest" --platform linux/amd64,linux/arm64 --push .
```

您也可以使用诸如之类的工具 AWS Cloud Development Kit (AWS CDK) 来帮助[编排构建](#)。有关 Docker 的示例，请参阅 Docker 文档中的[使用 Docker 桌面为 Arm 和 x86 构建多架构映像](#)。

Amazon EC2

要 ARM 从 x86/x64 迁移到，请在编译步骤中定位 ARM 架构。在 Visual Studio 中，您可以创建一个 ARM64 CPU。有关说明，请参阅 Microsoft 文档中的[将项目配置为以 Arm64 和其他平台为目标](#)。

如果你使用的是 .NET CLI，然后在 ARM 机器上运行该版本会生成与 Graviton 兼容的版本。要观看演示，请观看[加速 .NET 在 Graviton2 上开启 Arm64 时性能为 6](#)。YouTube 依赖问题将导致编译时错误，然后可以单独解决这些错误。只要存在任何依赖关系的 ARM 库，过渡就应该相对简单。

其他资源

- [如何使用亚马逊上的 Graviton ARM 和 Spot 实例构建容器并进行保存 ECS \(博客 \) AWS](#)
- [AWS Lambda 由 AWS Graviton2 处理器提供支持的功能 — 在 Arm 上运行您的功能，性价比最高可提高 34% \(博客 \) AWS](#)
- [将 AWS Lambda 函数迁移到基于 ARM 的 AWS Graviton2 处理器 \(博客 \) AWS](#)

- [构建和部署。NET使用 AWS CDK \(博客 \) ARM支持 AWS Graviton 2 Amazon Clu ECS sters 的 Web 应用程序AWS](#)
- [Graviton Fast Start — 一项帮助将工作负载转移到 AWS Graviton 的新计划 \(博客 \) AWS](#)
- [供电。NET5 使用 AWS Graviton2 : 基准测试 \(博客 \) AWS](#)

Support 支持静态的动态缩放。NET框架应用程序

概述

将云用于应用程序的主要好处之一是弹性，或者能够根据需求向内或向外扩展计算。这使您只需为所需的计算容量付费，而不必为峰值使用量进行预配置。网络星期一，在线零售商可以快速获得比平时多很多倍的流量（例如，[在几分钟内达到数千%](#)），这是弹性的一个很好的例子。

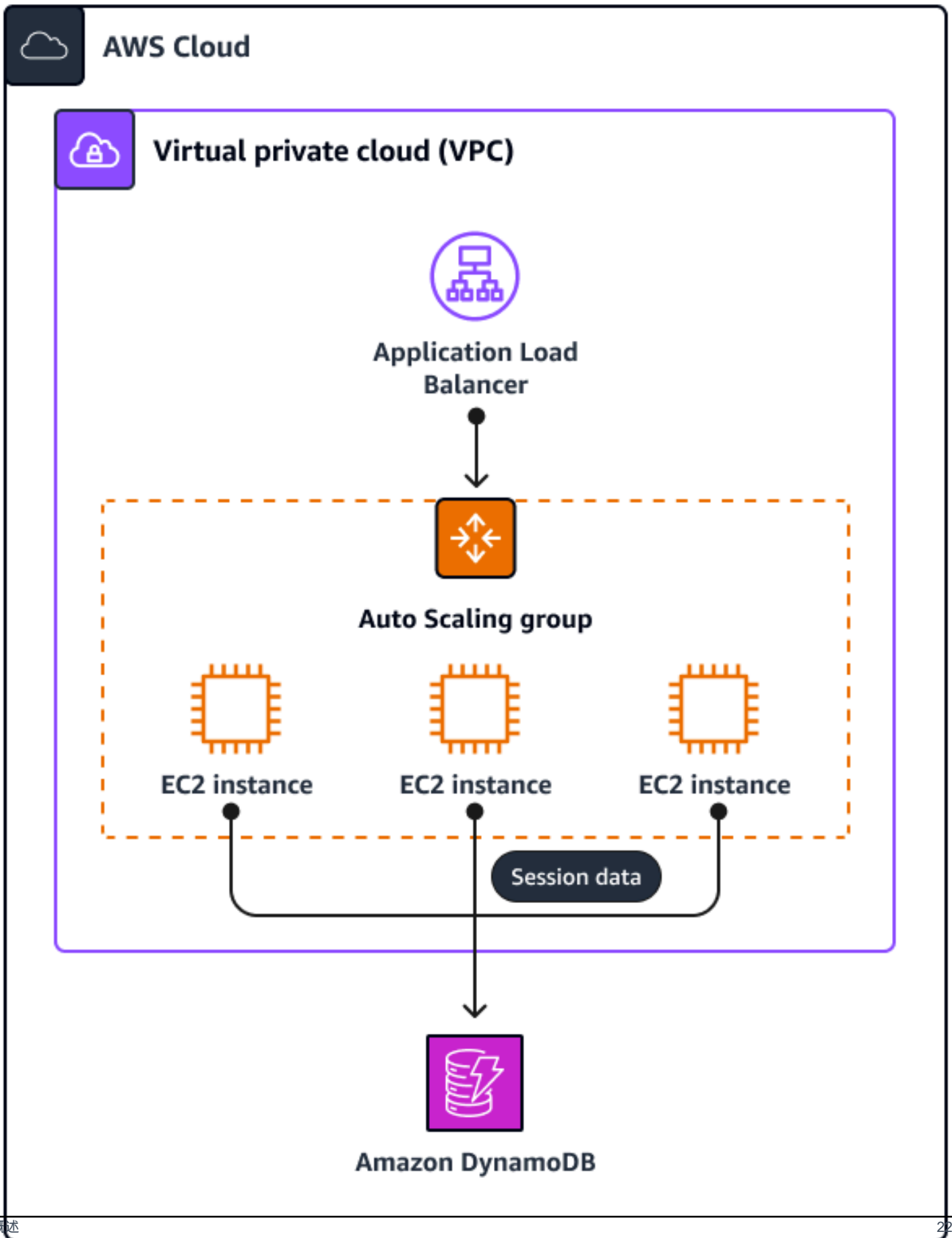
如果你要带来遗产。NET云端的 Web 应用程序（例如，ASP.NET框架应用程序在上运行IIS），但由于应用程序的状态性质，快速扩展负载平衡服务器群的能力可能很困难或不可能。用户会话数据存储在应用程序的内存中，通常存储在[ASP.NET会话状态](#)或静态变量，用于保存必须保留的跨请求数据。用户会话关联性通常通过负载均衡器粘性会话来维持。

事实证明，这在操作上具有挑战性。当需要增加容量时，必须有意识地配置和添加服务器。这可能是一个缓慢的过程。在修补或意外故障时使节点停止服务可能会给最终用户体验带来问题，所有与受影响节点关联的用户都将失去状态。充其量，这需要用户重新登录。

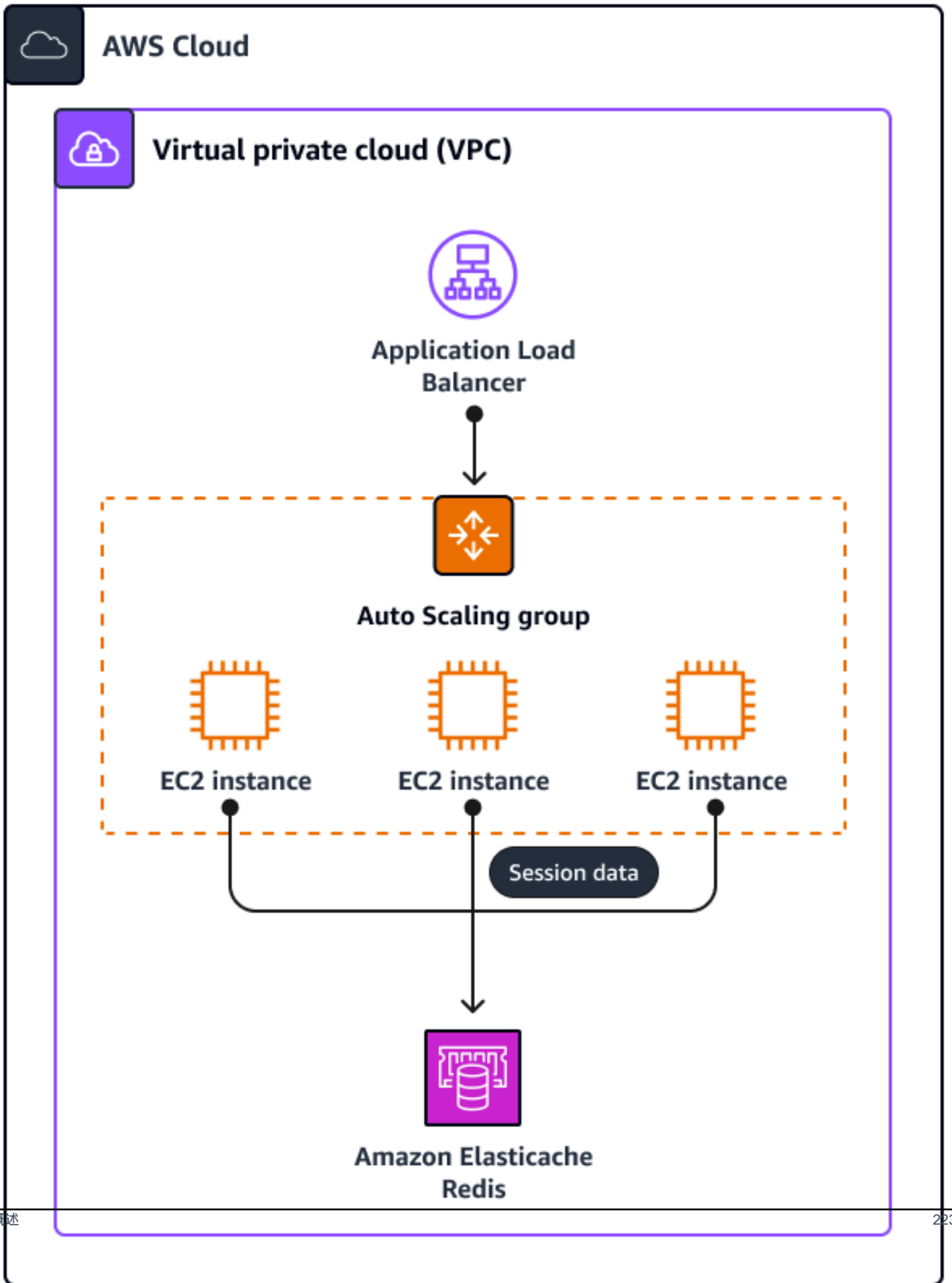
通过集中会话状态。ASP.NET应用程序并将自动缩放规则应用于旧版ASP.NET应用程序，您可以利用云的弹性，并有可能在运行应用程序时利用成本节约的优势。例如，您可以通过计算可扩展性降低成本，但也可以从不同的定价模式中进行选择，例如减少[预留实例使用量](#)和使用 [Amazon Spot 实例定价](#)。

两种常见的技术包括使用[亚马逊 DynamoDB 作为会话状态](#)提供程序和使用 [ElastiCache 亚马逊 \(OSSRedis\) 作为 ASP.NET会话存储](#)。

下图显示了使用 DynamoDB 作为会话状态提供程序的架构。



下图显示了使用 ElastiCache (RedisOSS) 作为会话状态提供者的架构。

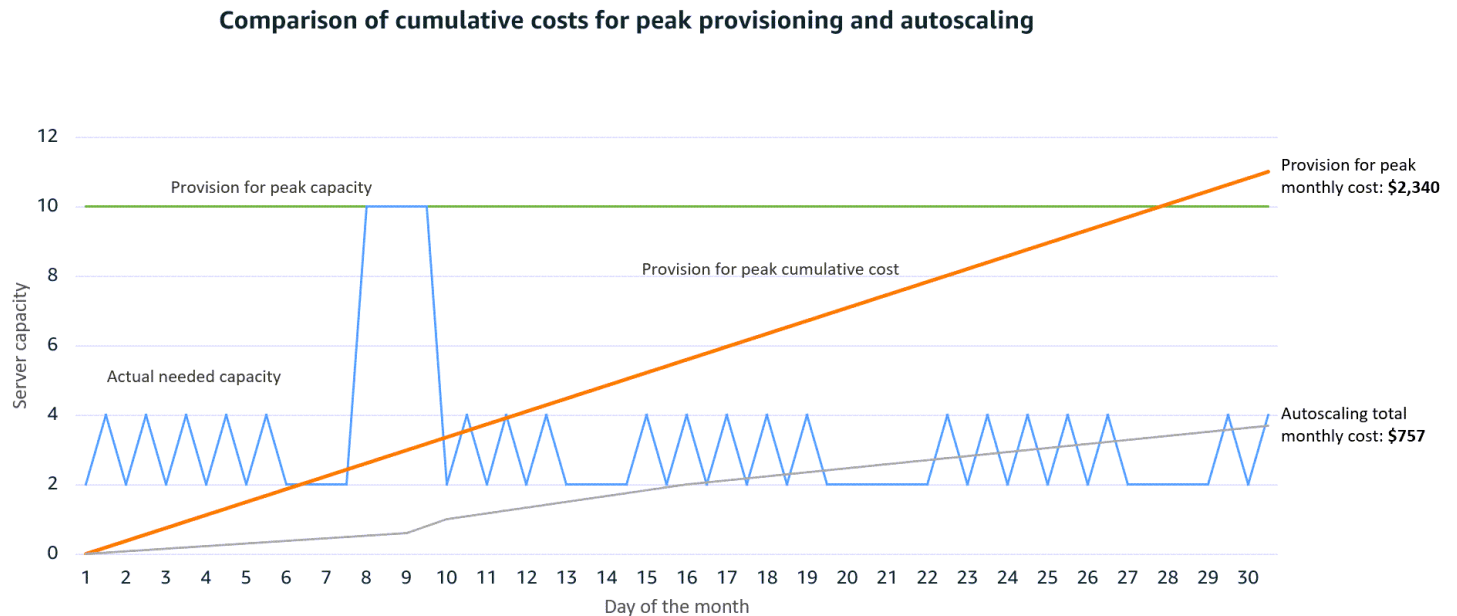


成本影响

为了确定扩展生产应用程序的好处，我们建议您对实际需求进行建模。本节假设以下假设来对示例应用程序进行建模：

- 在旋转中添加和移除的实例是相同的，并且不会引入任何实例大小变化。
- 为了保持应用程序的高可用性，服务器利用率永远不会降至两台活动服务器以下。
- 服务器的数量随流量线性扩展（也就是说，两倍的流量需要两倍的计算量）。
- 以六小时为增量对一个月的流量进行建模，其中包含当天差异和一个异常的流量峰值（例如促销活动），一天的流量为10倍。周末流量是根据基础利用率建模的。
- 夜间流量以基本利用率建模，而工作日流量按利用率的 4 倍建模。
- 预留实例定价使用为期一年的免费预付定价。普通日间定价使用按需定价，而突发需求使用竞价型实例定价。

下图说明了此模型如何利用中的弹性。NET应用程序，而不是为高峰使用量进行配置。这样可以节省大约 68% 的费用。



如果您使用 DynamoDB 作为会话状态存储机制，请使用以下参数：

```
Storage: 20GB
Session Reads: 40 million
Session Writes: 20 million
```

```
Pricing Model: On demand
```

这项服务的估计每月费用约为每月 35.00 美元。

如果您使用 ElastiCache (RedisOSS) 作为会话状态存储机制，则使用以下参数：

```
Number of Nodes: 3
Node size: cache.t4g.medium
Pricing Model: 1y reserved
```

这项服务的估计每月费用约为每月91.00美元。

成本优化建议

第一步是在旧版中实现会话状态。NET应用程序。如果您使用 ElastiCache 作为状态存储机制，请按照 AWS SDK for .NET 文档中的[内容 AWS SDK for .NET中的](#)指导进行操作。[如果您使用的是 DynamoDB，请按照中的指导ElastiCache 进行操作。](#)[ASP NET AWS 开发者工具博客中的会话存储。](#)

如果应用程序一开始就使用InProc会话，请确保计划存储在会话中的所有对象都可以序列化。为此，请使用SerializableAttribute属性来装饰其实例将存储在会话中的类。例如：

```
[Serializable()]
public class TestSimpleObject {
    public string SessionProperty {get;set;}
}
```

此外，.NETMachineKey所有正在使用的服务器之间必须相同。从常见的 Amazon 系统映像 (AMI) 创建实例时，通常会出现这种情况。例如：

```
<machineKey
validationKey="some long hashed value"
decryptionKey="another long hashed value"
validation="SHA1"/>
```

但是，重要的是要确保在更改基础映像时，其配置为相同的映像。NET计算机映像（可以在服务器级别IIS或服务器级别进行配置）。有关更多信息，请参阅[SystemWebSectionGroup. MachineKey 微软文档中的属性](#)。

最后，您必须确定将服务器添加到 Auto Scaling 组以响应扩展事件的机制。有几种方法可以做到这一点。我们建议使用以下方法进行无缝部署。NET在 Auto Scaling 组中为EC2实例构建框架应用程序：

- 使用 [EC2Image Builder](#) 配置AMI包含完全配置的服务器和应用程序的。然后，您可以使用它AMI来配置 [Auto Scaling 组的启动模板](#)。
- [AWS CodeDeploy](#)用于部署您的应用程序。CodeDeploy 允许直接与 [Amazon A EC2 uto Scaling 集成](#)。除了为应用程序的每个版本创建新版本AMI之外，这提供了另一种选择。

其他资源

- [使用 Im EC2 age Builder 创建EC2图像](#) (Image Builder 文档)
- [正在部署。NET使用 AWS CodeDeploy Visual Studio 团队服务的 Web 应用程序](#) (AWS 开发者工具博客)

使用缓存来减少数据库需求

概述

您可以使用缓存作为有效的策略来帮助降低成本。NET应用程序。当应用程序需要频繁访问数据时，许多应用程序都使用后端数据库，例如SQL服务器。维护这些后端服务以满足需求的成本可能很高，但是您可以使用有效的缓存策略，通过降低大小和扩展要求来减少后端数据库的负载。这可以帮助您降低成本并提高应用程序的性能。

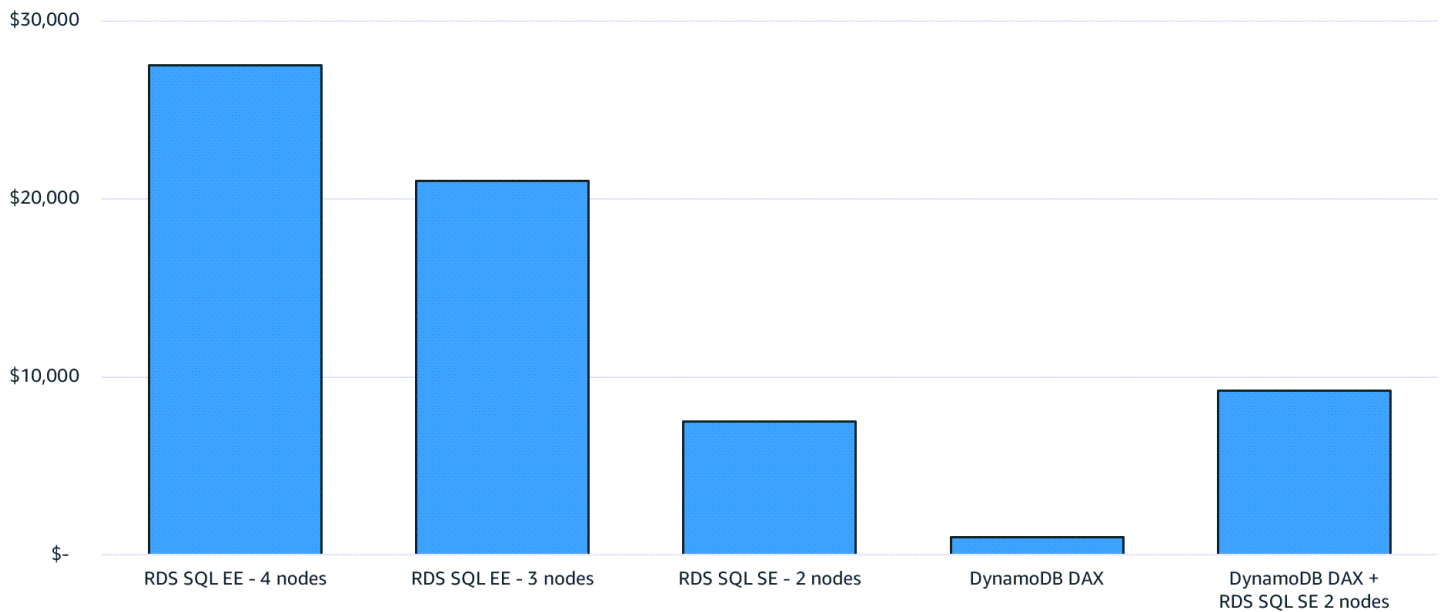
缓存是一种有用的技术，可以节省与读取使用SQL服务器等更昂贵资源的繁重工作负载相关的成本。为你的工作量使用正确的技巧很重要。例如，本地缓存不可扩展，需要您为应用程序的每个实例维护一个本地缓存。您应该权衡性能影响与潜在成本的比较，以便底层数据源的较低成本抵消与缓存机制相关的任何额外成本。

成本影响

SQL服务器要求您在调整数据库大小时考虑读取请求。这可能会影响成本，因为您可能需要引入只读副本以适应负载。如果您使用的是只读副本，请务必了解它们仅在 SQL Server Enterprise 版上可用。此版本需要比SQL服务器标准版更昂贵的许可证。

下图旨在帮助您了解缓存的有效性。它显示了 Amazon f RDS or SQL Server 有四个 db.m4.2xlarge 节点运行服务器企业版。SQL它部署在具有一个只读副本的多可用区配置中。独占读取流量 (例如SELECT查询) 将定向到只读副本。相比之下，亚马逊 DynamoDB 使用的是 r4.2xlarge 双节点 DynamoDB 加速器 () 集群。DAX

下图显示了不再需要处理高读取流量的专用只读副本的结果。



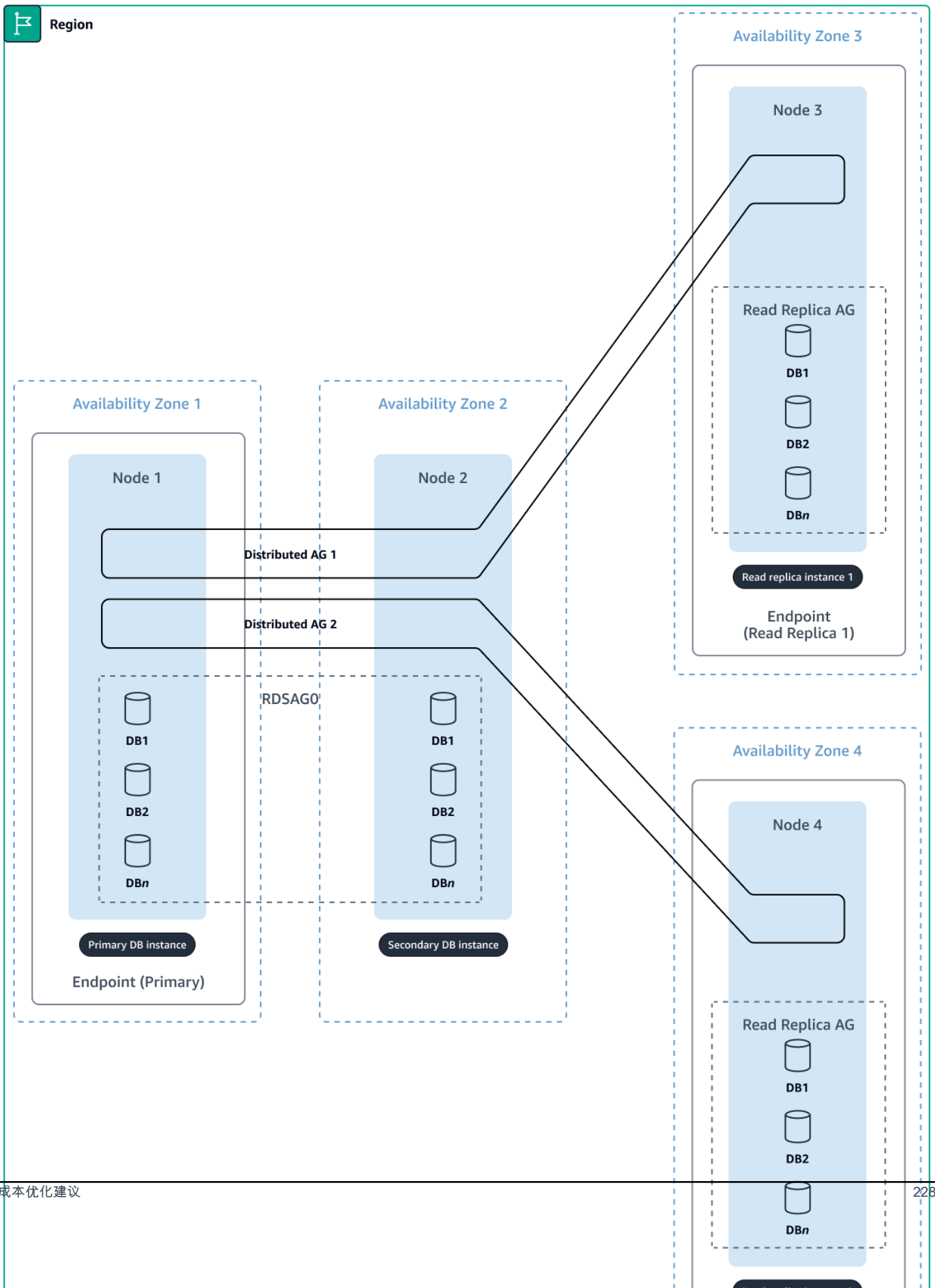
您可以通过使用不带只读副本的本地缓存或与 SQL Server on Amazon RDS 并行引入DAX缓存层来实现显著的成本节约。该层从SQL服务器卸载并减小了运行数据库所需的SQL服务器的大小。

成本优化建议

本地缓存

本地缓存是为本地环境或云端托管的应用程序缓存内容的最常用方法之一。这是因为它实现起来相对容易和直观。本地缓存包括从数据库或其他来源获取内容，然后在内存或磁盘上本地缓存以加快访问速度。这种方法虽然易于实现，但对于某些用例来说并不理想。例如，这包括缓存内容需要随着时间的推移而持续存在的用例，例如保留应用程序状态或用户状态。另一个用例是需要从其他应用程序实例访问缓存的内容。

下图说明了具有四个节点和两个只读副本的高可用性SQL服务器群集。



使用本地缓存，您可能需要对多个EC2实例的流量进行负载平衡。每个实例都必须维护自己的本地缓存。如果缓存存储有状态信息，则需要定期向数据库提交，并且可能需要将用户转发到每个后续请求（粘性会话）的同一个实例。在尝试扩展应用程序时，这会带来挑战，因为有些实例可能被过度利用，而有些实例则由于流量分布不均而未得到充分利用。

您可以将本地缓存（内存中缓存或使用本地存储）用于.NET应用程序。为此，您可以添加功能来将对象存储在磁盘上并在需要时检索它们，或者从数据库中查询数据并将其保存在内存中。例如，要在C#中对来自SQL服务器的数据进行内存和本地存储的本地缓存，可以使用MemoryCache和LiteDB库的组合。MemoryCache提供内存缓存，而LiteDB是一个嵌入式的No SQL基于磁盘的数据库，既快速又轻巧。

要执行内存缓存，请使用.NET图书馆System.Runtime.MemoryCache。以下代码示例说明如何使用该System.Runtime.Caching.MemoryCache类在内存中缓存数据。该类提供了一种将数据临时存储在应用程序内存中的方法。这可以减少从更昂贵的资源（例如数据库或）获取数据的需求，从而帮助提高应用程序的性能API。

以下是代码的工作原理：

1. 已创建一个 named 的私MemoryCache_memoryCache有静态实例。给缓存起一个名字 (dataCache) 来标识它。然后，缓存存储和检索数据。
2. 该GetData方法是一个通用方法，它有两个参数：一个string键和一个名为的Func<T>委托getData。密钥用于标识缓存的数据，而getData委托代表当缓存中不存在数据时执行的数据检索逻辑。
3. 该方法首先使用该_memoryCache.Contains(key)方法检查缓存中是否存在数据。如果数据在缓存中，则该方法使用检索数据，_memoryCache.Get(key)并将其转换为预期的类型 T。
4. 如果数据不在缓存中，则该方法会调用getData委托来获取数据。然后，它使用将数据添加到缓存中_memoryCache.Add(key, data, DateTimeOffset.Now.AddMinutes(10))。此调用指定缓存条目应在 10 分钟后过期，此时数据将自动从缓存中删除。
5. 该ClearCache方法将string密钥作为参数，并使用从缓存中删除与该密钥关联的数据_memoryCache.Remove(key)。

```
using System;
using System.Runtime.Caching;

public class InMemoryCache
{
    private static MemoryCache _memoryCache = new MemoryCache("dataCache");
```

```
public static T GetData<T>(string key, Func<T> getData)
{
    if (_memoryCache.Contains(key))
    {
        return (T)_memoryCache.Get(key);
    }

    T data = getData();
    _memoryCache.Add(key, data, DateTimeOffset.Now.AddMinutes(10));

    return data;
}

public static void ClearCache(string key)
{
    _memoryCache.Remove(key);
}
}
```

您可以使用以下代码：

```
public class Program
{
    public static void Main()
    {
        string cacheKey = "sample_data";

        Func<string> getSampleData = () =>
        {
            // Replace this with your data retrieval logic
            return "Sample data";
        };

        string data = InMemoryCache.GetData(cacheKey, getSampleData);
        Console.WriteLine("Data: " + data);
    }
}
```

以下示例向您展示如何使用 [LiteDB](#) 在本地存储中缓存数据。您可以使用 LiteDB 作为内存缓存的替代或补充。以下代码演示了如何使用 LiteDB 库将数据缓存在本地存储中。该 `LocalStorageCache` 类包含用于管理缓存的主要函数。

```
using System;
```



```
using LiteDB;

public class LocalStorageCache
{
    private static string _liteDbPath = @"Filename=LocalCache.db";

    public static T GetData<T>(string key, Func<T> getData)
    {
        using (var db = new LiteDatabase(_liteDbPath))
        {
            var collection = db.GetCollection<T>("cache");
            var item = collection.FindOne(Query.EQ("_id", key));

            if (item != null)
            {
                return item;
            }
        }

        T data = getData();

        using (var db = new LiteDatabase(_liteDbPath))
        {
            var collection = db.GetCollection<T>("cache");
            collection.Upsert(new BsonValue(key), data);
        }

        return data;
    }

    public static void ClearCache(string key)
    {
        using (var db = new LiteDatabase(_liteDbPath))
        {
            var collection = db.GetCollection("cache");
            collection.Delete(key);
        }
    }
}

public class Program
{
    public static void Main()
    {
```

```
string cacheKey = "sample_data";

Func<string> getSampleData = () =>
{
    // Replace this with your data retrieval logic
    return "Sample data";
};

string data = LocalStorageCache.GetData(cacheKey, getSampleData);
Console.WriteLine("Data: " + data);
}
}
```

如果您有不经常更改的静态缓存或静态文件，也可以将这些文件存储在亚马逊简单存储服务 (Amazon S3) Service 对象存储中。应用程序可以在启动时检索静态缓存文件以供本地使用。有关如何使用从 Amazon S3 检索文件的更多详细信息。NET，请参阅 Amazon S3 文档中的[下载对象](#)。

使用缓存 DAX

您可以使用可在所有应用程序实例之间共享的缓存层。D@@@ [ynamoDB Accelerator DAX \(\)](#) 是一款适用于 DynamoDB 的完全托管、高度可用的内存缓存，可以将性能提高十倍。您可以使用 DAX 减少在 DynamoDB 表中过度配置读取容量单位的需求，从而降低成本。这对于读取量大、需要重复读取单个密钥的工作负载特别有用。

DynamoDB 按需定价或按预配置容量定价，因此每月的读取和写入次数会影响成本。如果您有大量的读取负载，DAX 集群可以减少 DynamoDB 表的读取次数，从而帮助降低成本。有关如何设置的说明 DAX，请参阅 DynamoDB 文档中的[使用 DynamoDB 加速器 DAX \(\) 进行内存加速](#)。有关信息。NET 应用程序集成，请观看[将 Amazon DynamoDB 集成到 DAX 你的。ASP NET 申请已启用](#) YouTube。

其他资源

- 使用 Dynamo@@@ [DB 加速器进行内存加速 \(\) DAX-亚马逊 DynamoDB \(DynamoDB 文档 \)](#)
- [将亚马逊 DynamoDB 集成到 DAX 你的。ASP NET 应用程序](#) (YouTube)
- [正在下载对象](#) (亚马逊 S3 文档)

以无服务器为例。NET

概述

无服务器计算已成为构建和部署应用程序的流行方法。这主要是因为无服务器方法在构建现代架构时提供了可扩展性和敏捷性。但是，在某些情况下，必须考虑无服务器计算的成本影响。

Lambda 是一个无服务器计算平台，使开发人员无需专用服务器即可运行代码。Lambda 是一个特别有吸引力的选择。NET 希望降低基础架构成本的开发人员。有了 Lambda，。NET 开发人员可以开发和部署高度可扩展且可能具有成本效益的应用程序。通过使用无服务器方法，开发人员不再需要配置服务器来处理应用程序请求。相反，开发人员可以创建按需执行的函数。这使得无服务器方法比运行、管理和扩展虚拟机更具可扩展性、可管理性，而且可能更具成本效益。因此，您只需为应用程序使用的资源付费，而不必担心未充分利用的资源或服务器维护成本。

开发人员可以使用现代的跨平台。NET 用于构建快速、高效且经济实惠的无服务器应用程序的版本。的。NET 核心和更新版本是一个免费的开源框架，与以前相比，它更适合在无服务器平台上运行。NET 框架版本。这使开发人员能够减少开发时间并提高应用程序性能。现代。NET 还支持一系列编程语言，包括 C# 和 F#。因此，对于希望在云端构建现代架构的开发人员来说，这是一个有吸引力的选择。

本节介绍如何通过使用 Lambda 作为无服务器选项来节省成本。您可以通过微调 Lambda 函数的执行配置文件、调整您的 Lambda 函数的内存分配大小、[使用 AOT Native](#) 以及迁移到基于 Graviton 的函数来进一步优化成本。

成本影响

您可以降低多少成本取决于多个因素，包括您的无服务器函数将执行多少次以及分配的内存量和每个函数的持续时间。AWS Lambda 提供免费套餐，包括每月 100 万个免费请求和每月 400,000 GB 秒的计算时间。对于达到或接近这些免费套餐限制的工作负载，您可以显著降低每月费用。

使用以 Lambda 函数为目标的负载均衡器时，可能还会产生额外费用。这是根据负载均衡器为 [Lambda](#) 目标处理的数据量计算得出的。

成本优化建议

正确调整您的 Lambda 函数的大小

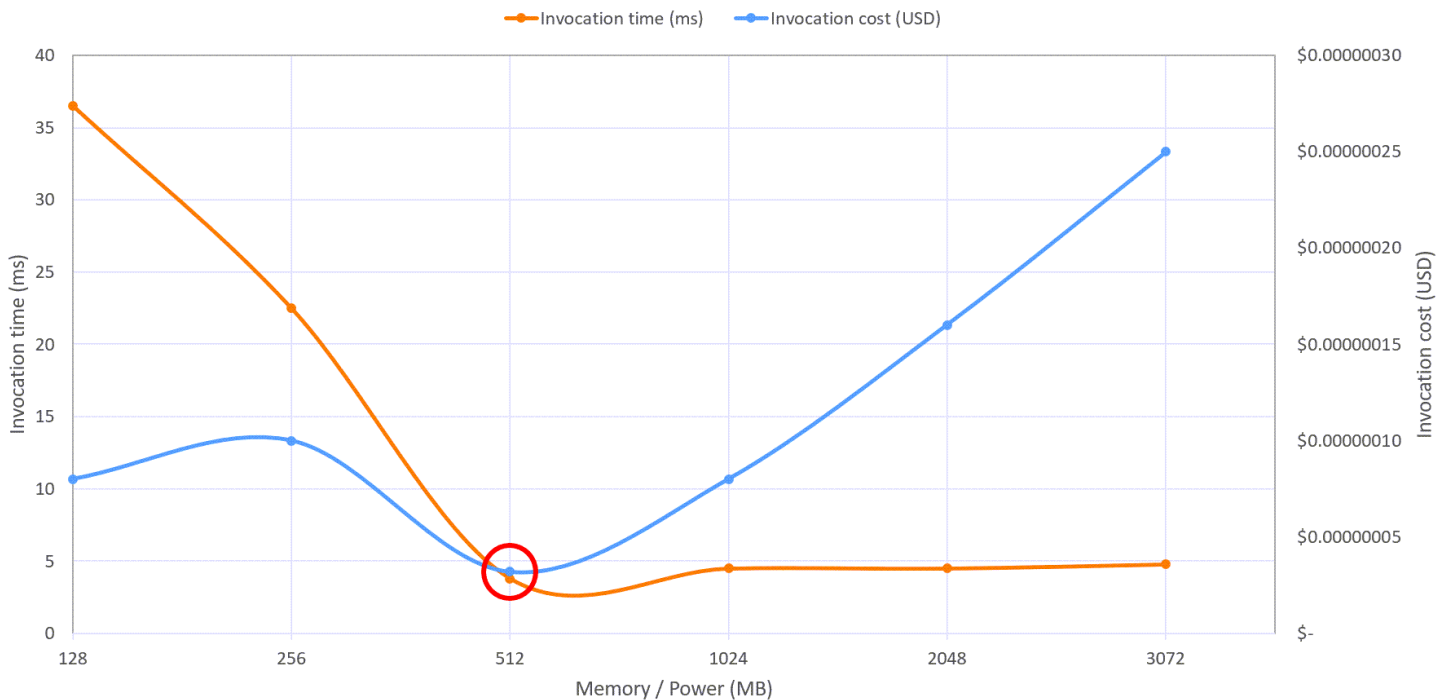
正确调整大小是成本优化的基本做法。NET 基于 Lambda 函数。该过程涉及确定在性能与成本效益之间取得平衡的最佳内存配置，而无需更改代码。

通过为 Lambda 函数配置内存（从 128 MB 到 10,240 MB 不等），您还可以调整调用期间 CPU 可用的 v 量。这使内存或 CPU 受限的应用程序能够在执行期间访问更多资源，从而有可能缩短调用持续时间和总体成本。

但是，请确定最适合您的配置。NET 基于 Lambda 函数可能是一个手动且耗时的过程，尤其是在频繁更改的情况下。[AWS Lambda Power Tuning 工具](#)可以根据示例负载分析一组内存配置，从而帮助您确定适当的配置。

例如，增加 a 的内存。NET 基于 Lambda 函数可以在不影响性能的情况下缩短总调用时间并降低成本。功能的最佳内存配置可能有所不同。AWS Lambda Power Tuning 工具可以帮助确定每项功能的最具成本效益的配置。

在下面的示例图表中，随着此 Lambda 函数内存的增加，总调用时间会缩短。这可以降低总执行成本，而不会影响函数的原始性能。对于此函数，该函数的最佳内存配置为 512 MB，因为对于每次调用的总成本，这是资源利用率最高的地方。这因函数而异，在您的 Lambda 函数上使用该工具可以确定它们是否受益于正确的大小。



我们建议您定期完成此练习，作为发布新更新时任何集成测试的一部分。如果不经常更新，请定期进行此练习，以确保功能经过调整和大小合适。确定了 Lambda 函数的相应内存设置后，您可以为进程添加合适的大小。AWS Lambda Power Tuning 工具会生成编程输出，在新代码发布期间，CI/CD 工作流程可使用这些输出。这使您能够自动配置内存。

您可以免费下载[AWS Lambda 功率调整工具](#)。有关如何使用该工具的说明，请参阅中的[如何执行状态机](#) GitHub。

Lambda 还支持原生AOT，这会启用。NET要预编译的应用程序。这可以通过缩短执行时间来帮助降低成本。NET函数。有关创建原生AOT函数的更多信息，请参阅。[NETLambda 文档中包含原生AOT编译的函数](#)。

避免空闲等待时间

Lambda 函数持续时间是用于计算账单的一个维度。当函数代码发出阻塞调用时，您需要按等待收到响应的的时间计费。当 Lambda 函数链接在一起或某个函数充当其他函数的协调器时，等待时间可能会增加。如果您有批量操作或订单交付系统等工作流程，则会增加管理开销。此外，可能无法在 15 分钟的最大 Lambda 超时时间内完成所有工作流程逻辑和错误处理。

我们建议您不要在函数代码中处理此逻辑，而是重新架构您的解决方案，以[AWS Step Functions](#)用作工作流程的协调者。使用标准工作流程时，您需要为工作流程中的每个[状态](#)转换付费，而不是按工作流程的总持续时间付费。此外，您可以将对重试、等待条件、错误工作流程和[回调](#)的支持转移到状态条件中，以允许您的 Lambda 函数专注于业务逻辑。有关更多信息，请参阅 AWS Compute 博客中的[优化 AWS Lambda 成本 — 第 2 部分](#)。

转到基于引力的函数

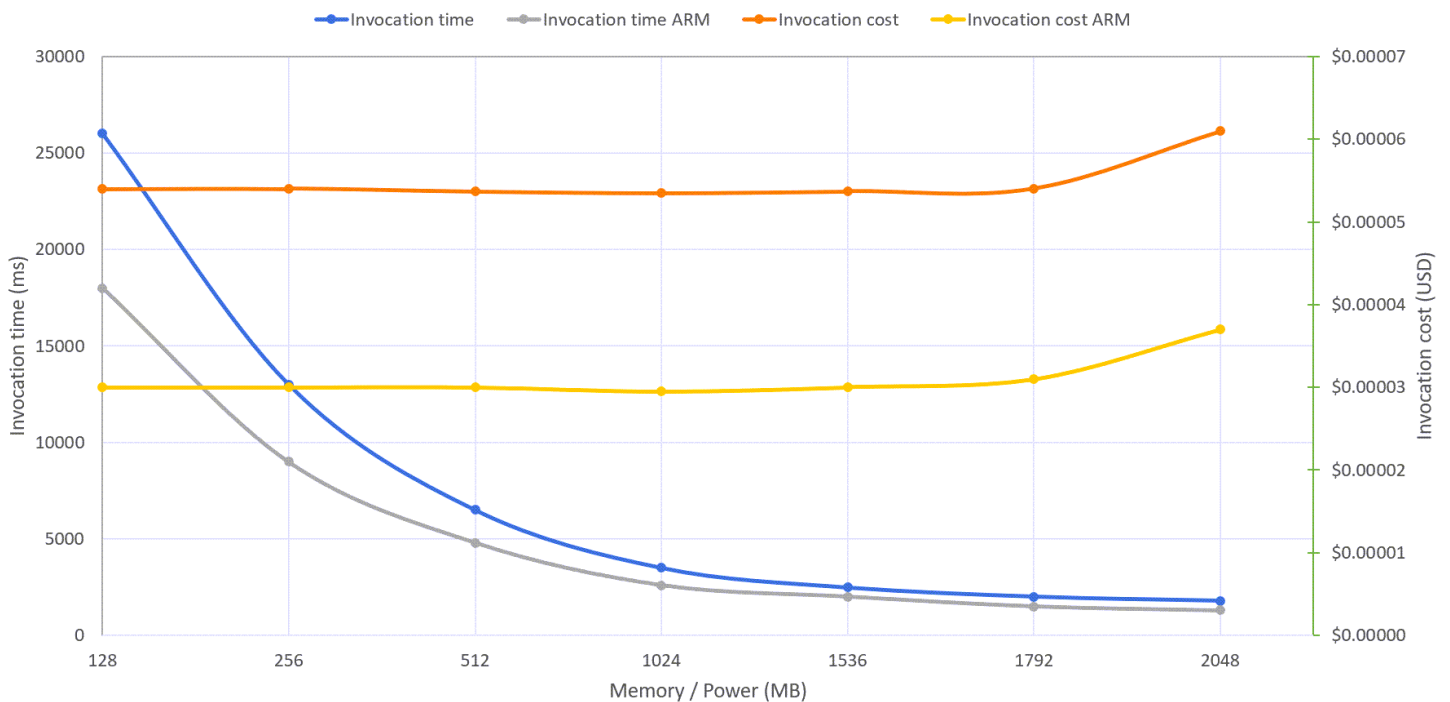
由下一代 Graviton2 处理器提供支持的 Lambda 函数现已正式上市。Graviton2功能使用ARM基于处理器的架构，旨在以低20%的成本为各种无服务器工作负载提供高达19%的性能提升。由Graviton2处理器提供支持的函数具有更低的延迟和更好的性能，非常适合为任务关键型无服务器应用程序提供动力。

迁移到基于 Graviton 的 Lambda 函数可能是一个经济实惠的选择。NET希望优化 Lambda 成本的开发人员。基于 Graviton 的函数使用ARM基于引力的处理器而不是传统的 x86 处理器。这可以在不牺牲性能的情况下显著节省成本。

虽然迁移到基于 Graviton 的函数有几个好处，但我们建议您考虑一些挑战和注意事项。例如，基于 Graviton 的函数需要使用 Amazon Linux 2，但可能无法与所有函数兼容。NET应用程序。此外，第三方库可能存在兼容性问题，或者依赖项与ARM基于处理器的处理器不兼容。

如果你在跑步。NET框架应用程序并希望通过 Lambda 利用无服务器功能，可以考虑将应用程序移植到现代版本。NET通过使用[移植助手](#)。[NET](#)。这可以帮助您加快旧版的移植。NET应用到现代。NET，使应用程序能够在 Linux 上运行。

下图比较了计算素数的函数的 x86 和 ARM /Graviton2 架构结果。



该函数使用的是单线程。当内存配置为 1.8 GB 时，会报告两种架构的最低持续时间。除此之外，Lambda 函数可以访问超过 1 v 的电压 CPU，但在这种情况下，该函数无法使用额外的功率。出于同样的原因，在内存高达 1.8 GB 的情况下，成本是稳定的。内存越多，成本就会增加，因为这种工作负载没有额外的性能优势。显然，Graviton2 处理器为这种计算密集型功能提供了更好的性能和更低的成本。

要将您的函数配置为在 Graviton 中使用 ARM 基于的处理器，请执行以下操作：

1. 登录 AWS Management Console 并打开 [Lambda 控制台](#)。
2. 选择 Create function (创建函数)。
3. 对于函数名称，输入一个名称。
4. 对于“运行时间”，选择。NET6 (C#/) PowerShell。
5. 对于架构，请选择 arm64。
6. 根据需要进行任何其他配置，然后选择创建函数。

其他资源

- [Lambda 函数作为目标 \(文档\)](#) AWS
- [使用 AWS Compute Optimizer \(AWS 计算博客\) 优化 AWS Lambda 成本和性能](#)
- [优化 AWS Lambda 成本 — 第 1 部分 \(AWS 计算博客\)](#)

- [优化 AWS Lambda 成本 — 第 2 部分](#) (AWS 计算博客)
- [构建无服务器。NET AWS Lambda 使用中的应用程序。NET7](#) (AWS 计算博客)

考虑专门构建的数据库

概述

运行基于 Microsoft 的工作负载最昂贵的方面之一来自于商用数据库 (例如服务器) 的许可。SQL 企业通常将 SQL 服务器作为首选的数据库平台进行标准化，这已在组织的开发文化中根深蒂固。无论用例如何，开发人员通常都会选择 SQL 基于关系服务器的模型。原因包括：

- 该企业已经有可用的 SQL 服务器实例和/或许可证。
- 通过使用共享库和业务逻辑，团队已经习惯了 SQL 编程模型。ORMs
- 管理层不知道其他选择。
- 开发人员不知道其他选择。

专门构建的数据库可以适应您的用例的数据访问模式。这些数据库越来越多地被企业采用，因为它们采用了更现代的架构 (例如微服务) ，而且单个应用程序的范围也越来越小。

专门构建的数据库并不排除关系模型，也不需要一个“否”SQL (非关系) 模型。实际上，当根据工作负载的特定需求选择关系数据库时，关系数据库被认为是专门构建的。使用专门构建的数据库可以帮助团队降低与其相关的数据库成本。NET 应用程序，同时还可以获得标准的云优势，例如可扩展性、弹性以及减少无差别的繁重工作。

下表显示了提供的专用数据库。AWS

数据库	类型	特性
亚马逊 Aurora PostgreSQL 或 亚马逊 Aurora MySQL	关系型	数据具有固定结构的用例 关系数据库自然会通过 ACID 事务保持数据一致性
Amazon DynamoDB	键值对	没有使用哈希表数据结构存储数据的数据 SQL 库

数据库	类型	特性
		<p>非结构化数据的高性能存储和检索</p> <p>用例包括用户配置文件、会话状态和购物车数据</p>
Amazon ElastiCache	内存中	<p>高性能没有SQL数据库在内存中存储非结构化数据，访问时间为亚毫秒</p> <p>用于经常访问的临时数据（例如用户会话），并用作其他速度较慢的数据存储之前的缓存层</p> <p>包括对 ElastiCache (Redis) 和 (MemcachedOSS) 的支持 ElastiCache</p>
Amazon MemoryDB	内存中经久耐用	兼容 Redis 的专用数据库，具有持久存储空间
Amazon Timestream	时间序列	<p>专为按时间顺序进行高吞吐量数据摄取而设计的数据库</p> <p>用例包括物联网 (IoT) 应用程序和存储指标或遥测数据</p>
Amazon DocumentDB	文档	<p>没有规定结构或与其他数据的强制关系就无法存储数据的数据SQL库</p> <p>通常用于读取密集型工作负载，例如产品目录</p>

数据库	类型	特性
Amazon Neptune	图表	<p>没有一个SQL数据库可以同时保存数据和数据项之间连接的表示形式</p> <p>用例包括欺诈检测、推荐引擎和社交应用程序</p>
亚马逊 Quantum Ledger 数据库 (亚马逊QLDB)	分类账	<p>存储交易数据并为每笔交易提供加密验证，从而提供可审计、不可变的历史记录</p> <p>当需要单一可验证的事实来源时，通常由应用程序使用</p>
Amazon Keyspaces	宽列	<p>基于 Apache Cassandra 的高性能分布式数据库</p> <p>用例包括物联网应用程序、事件处理和游戏应用程序</p>

专用数据库采用的一个重要驱动因素可以归因于取消商业许可。但是，Dynamo [DB](#) (包括按需模式)、Aurora、[Amazon Neptune](#) 和 [Amazon Keyspaces](#) 等数据库的自动缩放功能使您能够为普通案例而非峰值使用量预置容量。专门构建的数据库 (例如 Timestream 或 Amazon QLDB) 是无服务器的 QLDB，无需任何预配置即可自动扩展以满足需求。

AWS SQL如果您想使用专门构建的、与开源兼容的关系数据库，但又无法或不愿对应用程序进行重大代码更改，则提供 [Babelfish for Aurora PostgreSQL](#)。在某些情况下，Babelfish 允许您使用现有的SQL服务器访问代码，几乎不做任何更改。

在为应用程序选择专门构建的关系数据库时，保留应用程序所需的相同 (或功能等同的) 功能非常重要。此建议适用于将专门构建的数据库作为应用程序的主数据存储。其他建议中介绍了特定的应用程序 (例如缓存)。

成本影响

采用专门构建的数据库.NET工作负载虽然不太可能直接影响计算消耗/成本，但可以直接影响所使用的数据库服务的成本。NET应用程序。实际上，与敏捷性、可扩展性、弹性和数据持久性等额外优势相比，节省成本可能是次要目标。

解释为应用程序选择专门构建的数据库以及重新设计数据策略以有效使用这些数据库的完整过程超出了本指南的范围。有关更多信息，请参阅 AWS 教程目录中的[专用数据库](#)。

下表显示了几个示例，说明用专门构建的数据库替换 SQL Server 会如何改变应用程序成本。请注意，这些只是粗略的估计。需要对实际工作负载进行基准测试和优化，才能计算出确切的生产成本。

这些是一些常用的专用数据库估算值，包括按需计算和中的 100 GB SSD 单实例数据库。us-east-1许可费用包括SQL服务器许可证和软件保障。

下表显示了商业数据库示例的估计成本。

数据库引擎	许可模式	实例类型/规格	AWS 计算 + 存储成本	许可证成本	每月总费用
SQLAmazon 上的服务器标准版 EC2	随附许可证	r6i.2xlarge (8 /64 GB) CPU RAM	1,345.36 美元	0.00	1,345.36 美元
SQL亚马逊上的服务器企业版 EC2	随附许可证	r6i.2xlarge (8 /64 GB) CPU RAM	2,834.56 美元	0.00	2,834.56 美元
SQLAmazon 上的服务器标准版 EC2	BYOL	r6i.2xlarge (8 /64 GB) CPU RAM	644.56 美元	456.00 美元	1,100.56 美元
SQL亚马逊上的服务器企业版 EC2	BYOL	r6i.2xlarge (8 /64 GB) CPU RAM	644.56 美元	1,750.00 美元	2,394.56 美元
SQLAmazon 上的服务器标准版 RDS		db.r6i.2xlarge (8 /64 GB) CPU RAM	2,318.30 美元	0.00	2,318.30 美元

数据库引擎	许可模式	实例类型/规格	AWS 计算 + 存储成本	许可证成本	每月总费用
SQL 亚马逊上的服务器企业版 RDS		db.r6i.2xlarge (8 /64 GB) CPU RAM	3,750.56 美元	0.00	3,750.56 美元

下表显示了专门构建的示例的估计成本。

数据库引擎	实例类型/规格	AWS 计算 + 存储成本	许可证成本	每月总费用
亚马逊 Aurora PostgreSQL	r6g.2xlarge (8 /64 GB) CPU RAM	855.87 美元	0.00	855.87 美元
DynamoDB	预配置基础 100 /400 WCU RCU	72.00 美元		72.00 美元
Amazon DocumentDB	db.r6i.2xlarge (8 /64 GB) CPU RAM	778.60 美元		778.60 美元
Amazon QLDB	80M Write 400M Read 2 TB 期刊 600 GB 索引	320.00 美元		320.00 美元

Important

该表基于购买前三年带软件保障的 SQL 服务器的估计许可成本。SQL 服务器标准版：4,100 美元，2 个核心包，3 年。SQL 服务器企业版：15,700 美元，2 个核心包，3 年。

我们建议您在采用专门构建的数据库之前考虑成本影响。例如，更新应用程序以使用专门构建的数据库的成本与应用程序和源数据库的复杂性有关。在规划此架构切换时，请务必考虑总拥有成本。这包括重构应用程序、提高员工使用新技术的技能，以及仔细规划每项工作负载的预期性能和消耗。然后，您可以确定投资是否值得节省成本。在大多数情况下，维护 end-of-support 产品会带来安全和合规风险，修复产品的成本值得付出努力和初始投资。

成本优化建议

对于。NET 访问 SQL 服务器的应用程序中，有专门构建的关系数据库的替代库。你可以在应用程序中实现这些库来取代类似的 SQL 服务器应用程序功能。

下表重点介绍了一些可以在许多常见场景中使用的库。

Library	数据库	替换的	框架兼容性
Npgsql 实体框架核心提供者	亚马逊 Aurora Postgre SQL	实体框架核心 SQL 服务器提供商	现代。NET
Npgsql 实体框架 6 提供者	亚马逊 Aurora Postgre SQL	实体框架 6.0 SQL 服务器提供商	。NET 框架
npgsql (。ADO NET 兼容的 Postgre SQL 库)	亚马逊 Aurora Postgre SQL	ADO.NET	。NET 框架/现代。 NET
我的 SQL 实体框架核心提供者	亚马逊 Aurora 我的 SQL	实体框架核心 SQL 服务器提供商	现代。NET
Pomelo。EntityFrameworkCore。MySQL	亚马逊 Aurora 我的 SQL	实体框架核心 SQL 服务器提供商	现代。NET

使用 [Babelfish 连接亚马逊 Aurora Postgre SQL](#) 不需要任何特殊编码即可连接。但是，所有代码在使用前都应经过彻底测试。

其他专门构建的数据库有可供访问的库。NET 兼容库，使您能够访问专门构建的数据库。示例包括：

- [使用亚马逊 DynamoDB SQL 没有数据库 \(文档 \)](#) AWS SDK for .NET
- [MongoDB C# 驱动程序 \(MongoDB 文档 \)](#)

- [。 NET \(时间流文档 \)](#)
- [的@@ 亚马逊QLDB驱动程序。 NET \(亚马逊QLDB文档 \)](#)
- [使用 Cassandra。 NET以编程方式访问亚马逊密钥空间的核心客户端驱动程序 \(Amazon Keyspaces 文档 \)](#)
- [使用。 NET连接到 Neptune 数据库实例 \(Neptune 文档 \)](#)

如果您迁移到专门构建的数据库，则可以使用以下工具 AWS 来帮助完成迁移过程：

- [AWS Schema Conversion Tool \(AWS SCT\)](#) 可以帮助您将SQL服务器架构转换为亚马逊 Aurora 和亚马逊 DynamoDB。
- [AWS Database Migration Service \(AWS DMS\)](#) 可以帮助您一次性或持续地将数据从SQL服务器迁移到 Aurora 或 DynamoDB。
- [Babelfish Compass](#) 可以帮助你检查SQL服务器数据库的兼容性，以便与 Babelfish for Aurora Postgre 一起使用。SQL

其他 资源

- [将SQL服务器迁移到亚马逊 Aurora Postgre 的指南 SQL \(AWS 数据库博客 \)](#)
- [。 NET现代化AWS 研讨会 \(工作室工作室 \)](#)
- [Babelfish APP 现代化沉浸日 \(AWS 工作室工作室 \)](#)
- [。 NET沉浸日 \(工作室工作AWS 室 \)](#)
- 使用 [Amazon Timestream 入门 NET](#)(GitHub)
- [的@@ 亚马逊QLDB驱动程序。 NET— 快速入门教程 \(Amazon QLDB 文档 \)](#)
- [专为现代设计的数据库。 NET AWS \(AWS 演示文稿 \) 上的应用程序](#)

后续步骤

阅读完本指南后，我们建议您采取以下后续步骤来实现 MACO：

1. 联系 MACO 专家。MACO 专家可以帮助回答您的问题并解决您的疑虑。如果您已经在与 AWS 客户团队合作，请联系该团队并向 MACO 专家寻求帮助。如果你没有客户团队，请联系 optimize-microsoft@amazon.com。
2. 应用建议。应用您在本指南以及与 MACO 专家交谈中学到的建议、最佳实践和策略。
3. 跟踪成本变化。标记您的工作负载并使用 AWS Cost Explorer 和之类的服务 AWS Budgets 进行详细的成本跟踪、监控和控制。

文档历史记录

下表介绍了本指南的一些重要更改。如果您希望收到有关未来更新的通知，可以订阅 [RSS 源](#)。

变更	说明	日期
SQL 服务器和容器更新	我们添加了“ 使用 Compute Optimizer 优化 SQL Server 大小 ”、“ SQL Server 工作负载的查看 Trusted Advisor 建议 ”和“ 使用 App2Container 重新构建 Windows 应用程序 ”部分。	2024年6月29日
SQL 服务器许可优化	我们添加了“ 使用 Compute Optimizer 优化 SQL Server 许可 ”部分。	2024 年 5 月 22 日
初次发布	—	2023 年 12 月 21 日

AWS 规范性指导词汇表

以下是 AWS 规范性指导提供的策略、指南和模式中的常用术语。若要推荐词条，请使用术语表末尾的提供反馈链接。

数字

7 R

将应用程序迁移到云中的 7 种常见迁移策略。这些策略以 Gartner 于 2011 年确定的 5 R 为基础，包括以下内容：

- **重构/重新架构** - 充分利用云原生功能来提高敏捷性、性能和可扩展性，以迁移应用程序并修改其架构。这通常涉及到移植操作系统和数据库。示例：将您的本地 Oracle 数据库迁移到兼容 Amazon Aurora PostgreSQL 的版本。
- **更换平台** - 将应用程序迁移到云中，并进行一定程度的优化，以利用云功能。示例：在中将您的本地 Oracle 数据库迁移到适用于 Oracle 的亚马逊关系数据库服务 (Amazon RDS) AWS Cloud。
- **重新购买** - 转换到其他产品，通常是从传统许可转向 SaaS 模式。示例：将您的客户关系管理 (CRM) 系统迁移到 Salesforce.com。
- **更换主机 (直接迁移)** - 将应用程序迁移到云中，无需进行任何更改即可利用云功能。示例：在中的 EC2 实例上将您的本地 Oracle 数据库迁移到 Oracle AWS Cloud。
- **重新定位 (虚拟机监控器级直接迁移)**：将基础设施迁移到云中，无需购买新硬件、重写应用程序或修改现有操作。您可以将服务器从本地平台迁移到同一平台的云服务。示例：将 Microsoft Hyper-V 应用程序迁移到 AWS。
- **保留 (重访)** - 将应用程序保留在源环境中。其中可能包括需要进行重大重构的应用程序，并且您希望将工作推迟到以后，以及您希望保留的遗留应用程序，因为迁移它们没有商业上的理由。
- **停用** - 停用或删除源环境中不再需要的应用程序。

A

ABAC

请参阅[基于属性的访问控制](#)。

抽象服务

参见[托管服务](#)。

酸

参见[原子性、一致性、隔离性、持久性](#)。

主动-主动迁移

一种数据库迁移方法，在这种方法中，源数据库和目标数据库保持同步（通过使用双向复制工具或双写操作），两个数据库都在迁移期间处理来自连接应用程序的事务。这种方法支持小批量、可控的迁移，而不需要一次性割接。与[主动-被动迁移](#)相比，它更灵活，但需要更多的工作。

主动-被动迁移

一种数据库迁移方法，在这种方法中，源数据库和目标数据库保持同步，但在将数据复制到目标数据库时，只有源数据库处理来自连接应用程序的事务。目标数据库在迁移期间不接受任何事务。

聚合函数

一个 SQL 函数，它对一组行进行操作并计算该组的单个返回值。聚合函数的示例包括SUM和MAX。

AI

参见[人工智能](#)。

AIOps

参见[人工智能操作](#)。

匿名化

永久删除数据集中个人信息的过程。匿名化可以帮助保护个人隐私。匿名化数据不再被视为个人数据。

反模式

一种用于解决反复出现的问题的常用解决方案，而在这类问题中，此解决方案适得其反、无效或不如替代方案有效。

应用程序控制

一种安全方法，仅允许使用经批准的应用程序，以帮助保护系统免受恶意软件的侵害。

应用程序组合

有关组织使用的每个应用程序的详细信息的集合，包括构建和维护该应用程序的成本及其业务价值。这些信息是[产品组合发现和分析过程](#)的关键，有助于识别需要进行迁移、现代化和优化的应用程序并确定其优先级。

人工智能 (AI)

计算机科学领域致力于使用计算技术执行通常与人类相关的认知功能，例如学习、解决问题和识别模式。有关更多信息，请参阅[什么是人工智能？](#)

人工智能运营 (AIOps)

使用机器学习技术解决运营问题、减少运营事故和人为干预以及提高服务质量的过程。有关如何在 AWS 迁移策略中使用 AIOps 的更多信息，请参阅[运营集成指南](#)。

非对称加密

一种加密算法，使用一对密钥，一个公钥用于加密，一个私钥用于解密。您可以共享公钥，因为它不用于解密，但对私钥的访问应受到严格限制。

原子性、一致性、隔离性、持久性 (ACID)

一组软件属性，即使在出现错误、电源故障或其他问题的情况下，也能保证数据库的数据有效性和操作可靠性。

基于属性的访问权限控制 (ABAC)

根据用户属性 (如部门、工作角色和团队名称) 创建精细访问权限的做法。有关更多信息，请参阅 AWS Identity and Access Management (IAM) 文档 [AWS 中的 AB AC](#)。

权威数据源

存储主要数据版本的位置，被认为是最可靠的信息源。您可以将数据从权威数据源复制到其他位置，以便处理或修改数据，例如对数据进行匿名化、编辑或假名化。

可用区

中的一个不同位置 AWS 区域，不受其他可用区域故障的影响，并向同一区域中的其他可用区提供低成本、低延迟的网络连接。

AWS 云采用框架 (AWS CAF)

该框架包含指导方针和最佳实践 AWS，可帮助组织制定高效且有效的计划，以成功迁移到云端。AWS CAF 将指导分为六个重点领域，称为视角：业务、人员、治理、平台、安全和运营。业务、人员和治理角度侧重于业务技能和流程；平台、安全和运营角度侧重于技术技能和流程。例如，人员角度针对的是负责人力资源 (HR)、人员配置职能和人员管理的利益相关者。从这个角度来看，AWS CAF 为人员发展、培训和沟通提供了指导，以帮助组织为成功采用云做好准备。有关更多信息，请参阅[AWS CAF 网站](#)和[AWS CAF 白皮书](#)。

AWS 工作负载资格框架 (AWS WQF)

一种评估数据库迁移工作负载、推荐迁移策略和提供工作估算的工具。AWS WQF 包含在 AWS Schema Conversion Tool (AWS SCT) 中。它用来分析数据库架构和代码对象、应用程序代码、依赖关系和性能特征，并提供评测报告。

B

坏机器人

旨在破坏个人或组织或对其造成伤害的[机器人](#)。

BCP

参见[业务连续性计划](#)。

行为图

一段时间内资源行为和交互的统一交互式视图。您可以使用 Amazon Detective 的行为图来检查失败的登录尝试、可疑的 API 调用和类似的操作。有关更多信息，请参阅 Detective 文档中的[行为图中的数据](#)。

大端序系统

一个先存储最高有效字节的系统。另请参见[字节顺序](#)。

二进制分类

一种预测二进制结果（两个可能的类别之一）的过程。例如，您的 ML 模型可能需要预测诸如“该电子邮件是否为垃圾邮件？”或“这个产品是书还是汽车？”之类的问题

bloom 筛选条件

一种概率性、内存高效的数据结构，用于测试元素是否为集合的成员。

蓝/绿部署

一种部署策略，您可以创建两个独立但完全相同的环境。在一个环境中运行当前的应用程序版本（蓝色），在另一个环境中运行新的应用程序版本（绿色）。此策略可帮助您在影响最小的情况下快速回滚。

自动程序

一种通过互联网运行自动任务并模拟人类活动或互动的软件应用程序。有些机器人是有用或有益的，例如在互联网上索引信息的网络爬虫。其他一些被称为恶意机器人的机器人旨在破坏个人或组织或对其造成伤害。

僵尸网络

被[恶意软件](#)感染并受单方（称为[机器人](#)牧民或机器人操作员）控制的机器人网络。僵尸网络是最著名的扩展机器人及其影响力的机制。

分支

代码存储库的一个包含区域。在存储库中创建的第一个分支是主分支。您可以从现有分支创建新分支，然后在新分支中开发功能或修复错误。为构建功能而创建的分支通常称为功能分支。当功能可以发布时，将功能分支合并回主分支。有关更多信息，请参阅[关于分支](#)（GitHub 文档）。

破碎的玻璃通道

在特殊情况下，通过批准的流程，用户 AWS 账户 可以快速访问他们通常没有访问权限的内容。有关更多信息，请参阅 [Well -Architected 指南中的“实施破碎玻璃程序”](#) 指示 AWS 器。

棕地策略

您环境中的现有基础设施。在为系统架构采用棕地策略时，您需要围绕当前系统和基础设施的限制来设计架构。如果您正在扩展现有基础设施，则可以将棕地策略和[全新](#)策略混合。

缓冲区缓存

存储最常访问的数据的内存区域。

业务能力

企业如何创造价值（例如，销售、客户服务或营销）。微服务架构和开发决策可以由业务能力驱动。有关更多信息，请参阅[在 AWS 上运行容器化微服务](#)白皮书中的[围绕业务能力进行组织](#)部分。

业务连续性计划（BCP）

一项计划，旨在应对大规模迁移等破坏性事件对运营的潜在影响，并使企业能够快速恢复运营。

C

CAF

参见[AWS 云采用框架](#)。

金丝雀部署

向最终用户缓慢而渐进地发布版本。当你有信心时，你可以部署新版本并全部替换当前版本。

CCoE

参见[云卓越中心](#)。

CDC

参见[变更数据捕获](#)。

更改数据捕获 (CDC)

跟踪数据来源 (如数据库表) 的更改并记录有关更改的元数据的过程。您可以将 CDC 用于各种目的，例如审计或复制目标系统中的更改以保持同步。

混沌工程

故意引入故障或破坏性事件来测试系统的弹性。您可以使用 [AWS Fault Injection Service \(AWS FIS\)](#) 来执行实验，对您的 AWS 工作负载施加压力并评估其响应。

CI/CD

查看[持续集成和持续交付](#)。

分类

一种有助于生成预测的分类流程。分类问题的 ML 模型预测离散值。离散值始终彼此不同。例如，一个模型可能需要评估图像中是否有汽车。

客户端加密

在目标 AWS 服务 收到数据之前，对数据进行本地加密。

云卓越中心 (CCoE)

一个多学科团队，负责推动整个组织的云采用工作，包括开发云最佳实践、调动资源、制定迁移时间表、领导组织完成大规模转型。有关更多信息，请参阅 AWS Cloud 企业战略博客上的 [CCoE 帖子](#)。

云计算

通常用于远程数据存储和 IoT 设备管理的云技术。云计算通常与[边缘计算](#)技术相关。

云运营模型

在 IT 组织中，一种用于构建、完善和优化一个或多个云环境的运营模型。有关更多信息，请参阅[构建您的云运营模型](#)。

云采用阶段

组织迁移到以下阶段时通常会经历四个阶段 AWS Cloud：

- 项目 - 出于概念验证和学习目的，开展一些与云相关的项目
- 基础 - 进行基础投资以扩大云采用率 (例如，创建登录区、定义 CCoE、建立运营模型)

- 迁移 - 迁移单个应用程序
- 重塑 - 优化产品和服务，在云中创新

Stephen Orban 在 AWS Cloud 企业战略博客的博客文章 [《云优先之旅和采用阶段》](#) 中定义了这些阶段。有关它们与 AWS 迁移策略的关系的信息，请参阅 [迁移准备指南](#)。

CMDB

参见 [配置管理数据库](#)。

代码存储库

通过版本控制过程存储和更新源代码和其他资产（如文档、示例和脚本）的位置。常见的云存储库包括 GitHub 或 AWS CodeCommit。每个版本的代码都称为一个分支。在微服务结构中，每个存储库都专门用于一个功能。单个 CI/CD 管道可以使用多个存储库。

冷缓存

一种空的、填充不足或包含过时或不相关数据的缓冲区缓存。这会影响性能，因为数据库实例必须从主内存或磁盘读取，这比从缓冲区缓存读取要慢。

冷数据

很少访问的数据，且通常是历史数据。查询此类数据时，通常可以接受慢速查询。将这些数据转移到性能较低且成本更低的存储层或类别可以降低成本。

计算机视觉 (CV)

[人工智能](#) 领域，使用机器学习来分析和提取数字图像和视频等视觉格式中的信息。例如，AWS Panorama 提供将 CV 添加到本地摄像机网络的设备，而 Amazon 则为 CV SageMaker 提供图像处理算法。

配置偏差

对于工作负载，配置会从预期状态发生变化。这可能会导致工作负载变得不合规，而且通常是渐进的，不是故意的。

配置管理数据库 (CMDB)

一种存储库，用于存储和管理有关数据库及其 IT 环境的信息，包括硬件和软件组件及其配置。您通常在迁移的产品组合发现和分析阶段使用来自 CMDB 的数据。

合规性包

一系列 AWS Config 规则和补救措施，您可以汇编这些规则和补救措施，以自定义合规性和安全性检查。您可以使用 YAML 模板将一致性包作为单个实体部署在 AWS 账户和区域或整个组织中。有关更多信息，请参阅 AWS Config 文档中的 [一致性包](#)。

持续集成和持续交付 (CI/CD)

自动执行软件发布过程的源代码、构建、测试、暂存和生产阶段的过程。CI/CD 通常被描述为管道。CI/CD 可以帮助您实现流程自动化、提高工作效率、改善代码质量并加快交付速度。有关更多信息，请参阅[持续交付的优势](#)。CD 也可以表示持续部署。有关更多信息，请参阅[持续交付与持续部署](#)。

CV

参见[计算机视觉](#)。

D

静态数据

网络中静止的数据，例如存储中的数据。

数据分类

根据网络中数据的关键性和敏感性对其进行识别和分类的过程。它是任何网络安全风险管理策略的关键组成部分，因为它可以帮助您确定对数据的适当保护和保留控制。数据分类是 Well-Architected AWS d Framework 中安全支柱的一个组成部分。有关详细信息，请参阅[数据分类](#)。

数据漂移

生产数据与用来训练机器学习模型的数据之间的有意义差异，或者输入数据随时间推移的有意义变化。数据漂移可能降低机器学习模型预测的整体质量、准确性和公平性。

传输中数据

在网络中主动移动的数据，例如在网络资源之间移动的数据。

数据网格

一种架构框架，可提供分布式、去中心化的数据所有权以及集中式管理和治理。

数据最少化

仅收集并处理绝对必要数据的原则。在中进行数据最小化 AWS Cloud 可以降低隐私风险、成本和分析碳足迹。

数据边界

AWS 环境中的一组预防性防护措施，可帮助确保只有可信身份才能访问来自预期网络的可信资源。有关更多信息，请参阅在[上构建数据边界](#)。AWS

数据预处理

将原始数据转换为 ML 模型易于解析的格式。预处理数据可能意味着删除某些列或行，并处理缺失、不一致或重复的值。

数据溯源

在数据的整个生命周期跟踪其来源和历史的过程，例如数据如何生成、传输和存储。

数据主体

正在收集和处理其数据的个人。

数据仓库

一种支持商业智能（例如分析）的数据管理系统。数据仓库通常包含大量历史数据，通常用于查询和分析。

数据库定义语言 (DDL)

在数据库中创建或修改表和对象结构的语句或命令。

数据库操作语言 (DML)

在数据库中修改（插入、更新和删除）信息的语句或命令。

DDL

参见[数据库定义语言](#)。

深度融合

组合多个深度学习模型进行预测。您可以使用深度融合来获得更准确的预测或估算预测中的不确定性。

深度学习

一个 ML 子字段使用多层神经网络来识别输入数据和感兴趣的目标变量之间的映射。

defense-in-depth

一种信息安全方法，经过深思熟虑，在整个计算机网络中分层实施一系列安全机制和控制措施，以保护网络及其中数据的机密性、完整性和可用性。当你采用这种策略时 AWS，你会在 AWS Organizations 结构的不同层面添加多个控件来帮助保护资源。例如，一种 defense-in-depth 方法可以结合多因素身份验证、网络分段和加密。

委托管理员

在中 AWS Organizations，兼容的服务可以注册 AWS 成员帐户来管理组织的帐户并管理该服务的权限。此帐户被称为该服务的委托管理员。有关更多信息和兼容服务列表，请参阅 AWS Organizations 文档中[使用 AWS Organizations 的服务](#)。

部署

使应用程序、新功能或代码修复在目标环境中可用的过程。部署涉及在代码库中实现更改，然后在应用程序的环境中构建和运行该代码库。

开发环境

参见[环境](#)。

侦测性控制

一种安全控制，在事件发生后进行检测、记录日志和发出警报。这些控制是第二道防线，提醒您注意绕过现有预防性控制的安全事件。有关更多信息，请参阅在 AWS 上实施安全控制中的[侦测性控制](#)。

开发价值流映射 (DVSM)

用于识别对软件开发生命周期中的速度和质量产生不利影响的限制因素并确定其优先级的流程。DVSM 扩展了最初为精益生产实践设计的价值流映射流程。其重点关注在软件开发过程中创造和转移价值所需的步骤和团队。

数字孪生

真实世界系统的虚拟再现，如建筑物、工厂、工业设备或生产线。数字孪生支持预测性维护、远程监控和生产优化。

维度表

在[星型架构](#)中，一种较小的表，其中包含事实表中定量数据的数据属性。维度表属性通常是文本字段或行为类似于文本的离散数字。这些属性通常用于查询约束、筛选和结果集标注。

灾难

阻止工作负载或系统在其主要部署位置实现其业务目标的事件。这些事件可能是自然灾害、技术故障或人为操作的结果，例如无意的配置错误或恶意软件攻击。

灾难恢复 (DR)

您用来最大限度地减少[灾难](#)造成的停机时间和数据丢失的策略和流程。有关更多信息，请参阅 Well-Architected Framework AWS work 中的[“工作负载灾难恢复：云端 AWS 恢复”](#)。

DML

参见[数据库操作语言](#)。

领域驱动设计

一种开发复杂软件系统的方法，通过将其组件连接到每个组件所服务的不断发展的领域或核心业务目标。Eric Evans 在其著作[领域驱动设计：软件核心复杂性应对之道](#)（Boston: Addison-Wesley Professional, 2003）中介绍了这一概念。有关如何将领域驱动设计与 strangler fig 模式结合使用的信息，请参阅[使用容器和 Amazon API Gateway 逐步将原有的 Microsoft ASP.NET \(ASMX \) Web 服务现代化](#)。

DR

参见[灾难恢复](#)。

漂移检测

跟踪与基准配置的偏差。例如，您可以使用 AWS CloudFormation 来[检测系统资源中的偏差](#)，也可以使用 AWS Control Tower 来[检测着陆区中可能影响监管要求合规性的变化](#)。

DVSM

参见[开发价值流映射](#)。

E

EDA

参见[探索性数据分析](#)。

边缘计算

该技术可提高位于 IoT 网络边缘的智能设备的计算能力。与[云计算](#)相比，边缘计算可以减少通信延迟并缩短响应时间。

加密

一种将人类可读的纯文本数据转换为密文的计算过程。

加密密钥

由加密算法生成的随机位的加密字符串。密钥的长度可能有所不同，而且每个密钥都设计为不可预测且唯一。

字节顺序

字节在计算机内存中的存储顺序。大端序系统先存储最高有效字节。小端序系统先存储最低有效字节。

端点

参见[服务端点](#)。

端点服务

一种可以在虚拟私有云 (VPC) 中托管，与其他用户共享的服务。您可以使用其他 AWS 账户 或 AWS Identity and Access Management (IAM) 委托人创建终端节点服务，AWS PrivateLink 并向其授予权限。这些账户或主体可通过创建接口 VPC 端点来私密地连接到您的端点服务。有关更多信息，请参阅 Amazon Virtual Private Cloud (Amazon VPC) 文档中的[创建端点服务](#)。

企业资源规划 (ERP)

一种自动化和管理企业关键业务流程 (例如会计、[MES](#) 和项目管理) 的系统。

信封加密

用另一个加密密钥对加密密钥进行加密的过程。有关更多信息，请参阅 AWS Key Management Service (AWS KMS) 文档中的[信封加密](#)。

environment

正在运行的应用程序的实例。以下是云计算中常见的环境类型：

- 开发环境 — 正在运行的应用程序的实例，只有负责维护应用程序的核心团队才能使用。开发环境用于测试更改，然后再将其提升到上层环境。这类环境有时称为测试环境。
- 下层环境 — 应用程序的所有开发环境，比如用于初始构建和测试的环境。
- 生产环境 — 最终用户可以访问的正在运行的应用程序的实例。在 CI/CD 管道中，生产环境是最后一个部署环境。
- 上层环境 — 除核心开发团队以外的用户可以访问的所有环境。这可能包括生产环境、预生产环境和用户验收测试环境。

epic

在敏捷方法学中，有助于组织工作和确定优先级的功能类别。epics 提供了对需求和实施任务的总体描述。例如，AWS CAF 安全史诗包括身份和访问管理、侦探控制、基础设施安全、数据保护和事件响应。有关 AWS 迁移策略中 epics 的更多信息，请参阅[计划实施指南](#)。

ERP

参见[企业资源规划](#)。

探索性数据分析 (EDA)

分析数据集以了解其主要特征的过程。您收集或汇总数据，并进行初步调查，以发现模式、检测异常并检查假定情况。EDA 通过计算汇总统计数据和创建数据可视化得以执行。

F

事实表

[星形架构](#)中的中心表。它存储有关业务运营的定量数据。通常，事实表包含两种类型的列：包含度量的列和包含维度表外键的列。

失败得很快

一种使用频繁和增量测试来缩短开发生命周期的理念。这是敏捷方法的关键部分。

故障隔离边界

在中 AWS Cloud，诸如可用区 AWS 区域、控制平面或数据平面之类的边界，它限制了故障的影响并有助于提高工作负载的弹性。有关更多信息，请参阅[AWS 故障隔离边界](#)。

功能分支

参见[分支](#)。

特征

您用来进行预测的输入数据。例如，在制造环境中，特征可能是定期从生产线捕获的图像。

特征重要性

特征对于模型预测的重要性。这通常表示为数值分数，可以通过各种技术进行计算，例如 Shapley 加法解释 (SHAP) 和积分梯度。有关更多信息，请参阅[机器学习模型的可解释性：AWS](#)。

功能转换

为 ML 流程优化数据，包括使用其他来源丰富数据、扩展值或从单个数据字段中提取多组信息。这使得 ML 模型能从数据中获益。例如，如果您将“2021-05-27 00:15:37”日期分解为“2021”、“五月”、“星期四”和“15”，则可以帮助学习与不同数据成分相关的算法学习精细模式。

FGAC

请参阅[精细的访问控制](#)。

精细访问控制 (FGAC)

使用多个条件允许或拒绝访问请求。

快闪迁移

一种数据库迁移方法，它使用连续的数据复制，通过[更改数据捕获](#)在尽可能短的时间内迁移数据，而不是使用分阶段的方法。目标是将停机时间降至最低。

G

地理封锁

请参阅[地理限制](#)。

地理限制 (地理阻止)

在 Amazon 中 CloudFront，一种阻止特定国家/地区的用户访问内容分发的选项。您可以使用允许列表或阻止列表来指定已批准和已禁止的国家/地区。有关更多信息，请参阅 CloudFront 文档[中的限制内容的地理分布](#)。

GitFlow 工作流程

一种方法，在这种方法中，下层和上层环境在源代码存储库中使用不同的分支。Gitflow 工作流程被认为是传统的，而[基于主干的工作流程](#)是现代的首选方法。

全新策略

在新环境中缺少现有基础设施。在对系统架构采用全新策略时，您可以选择所有新技术，而不受对现有基础设施 (也称为[棕地](#)) 兼容性的限制。如果您正在扩展现有基础设施，则可以将棕地策略和全新策略混合。

防护机制

一种高级规则，用于跨组织单位 (OU) 管理资源、策略和合规性。预防性防护机制会执行策略以确保符合合规性标准。它们是使用服务控制策略和 IAM 权限边界实现的。侦测性防护机制会检测策略违规和合规性问题，并生成警报以进行修复。它们通过使用 AWS Config、Amazon、AWS Security Hub GuardDuty AWS Trusted Advisor、Amazon Inspector 和自定义 AWS Lambda 支票来实现。

H

HA

参见[高可用性](#)。

异构数据库迁移

将源数据库迁移到使用不同数据库引擎的目标数据库（例如，从 Oracle 迁移到 Amazon Aurora）。异构迁移通常是重新架构工作的一部分，而转换架构可能是一项复杂的任务。[AWS 提供了 AWS SCT](#) 来帮助实现架构转换。

高可用性 (HA)

在遇到挑战或灾难时，工作负载无需干预即可连续运行的能力。HA 系统旨在自动进行故障转移、持续提供良好性能，并以最小的性能影响处理不同负载和故障。

历史数据库现代化

一种用于实现运营技术 (OT) 系统现代化和升级以更好满足制造业需求的方法。历史数据库是一种用于收集和存储工厂中各种来源数据的数据库。

同构数据库迁移

将源数据库迁移到共享同一数据库引擎的目标数据库（例如，从 Microsoft SQL Server 迁移到 Amazon RDS for SQL Server）。同构迁移通常是更换主机或更换平台工作的一部分。您可以使用本机数据库实用程序来迁移架构。

热数据

经常访问的数据，例如实时数据或近期的转化数据。这些数据通常需要高性能存储层或存储类别才能提供快速的查询响应。

修补程序

针对生产环境中关键问题的紧急修复。由于其紧迫性，修补程序通常是在典型的 DevOps 发布工作流程之外进行的。

hypercare 周期

割接之后，迁移团队立即管理和监控云中迁移的应用程序以解决任何问题的时间段。通常，这个周期持续 1-4 天。在 hypercare 周期结束时，迁移团队通常会将应用程序的责任移交给云运营团队。

|

IaC

参见[基础架构即代码](#)。

基于身份的策略

附加到一个或多个 IAM 委托人的策略，用于定义他们在 AWS Cloud 环境中的权限。

|

空闲应用程序

90 天内平均 CPU 和内存使用率在 5% 到 20% 之间的应用程序。在迁移项目中，通常会停用这些应用程序或将其保留在本地。

IloT

参见[工业物联网](#)。

不可变的基础架构

一种为生产工作负载部署新基础架构，而不是更新、修补或修改现有基础架构的模型。[不可变基础架构本质上比可变基础架构更一致、更可靠、更可预测](#)。有关更多信息，请参阅 Well-Architected Framework 中的[使用不可变基础架构 AWS 部署最佳实践](#)。

入站 (入口) VPC

在 AWS 多账户架构中，一种接受、检查和路由来自应用程序外部的网络连接的 VPC。[AWS 安全参考架构](#)建议使用入站、出站和检查 VPC 设置网络账户，保护应用程序与广泛的互联网之间的双向接口。

增量迁移

一种割接策略，在这种策略中，您可以将应用程序分成小部分进行迁移，而不是一次性完整割接。例如，您最初可能只将几个微服务或用户迁移到新系统。在确认一切正常后，您可以逐步迁移其他微服务或用户，直到停用遗留系统。这种策略降低了大规模迁移带来的风险。

工业 4.0

该术语由[克劳斯·施瓦布 \(Klaus Schwab \)](#)于2016年推出，指的是通过连接、实时数据、自动化、分析和人工智能/机器学习的进步实现制造流程的现代化。

基础设施

应用程序环境中包含的所有资源和资产。

基础设施即代码 (IaC)

通过一组配置文件预置和管理应用程序基础设施的过程。IaC 旨在帮助您集中管理基础设施、实现资源标准化和快速扩展，使新环境具有可重复性、可靠性和一致性。

工业物联网 (IloT)

在工业领域使用联网的传感器和设备，例如制造业、能源、汽车、医疗保健、生命科学和农业。有关更多信息，请参阅[制定工业物联网 \(IloT \) 数字化转型策略](#)。

检查 VPC

在 AWS 多账户架构中，一种集中式 VPC，用于管理 VPC（相同或不同 AWS 区域）、互联网和本地网络之间的网络流量检查。[AWS 安全参考架构](#)建议使用入站、出站和检查 VPC 设置网络账户，保护应用程序与广泛的互联网之间的双向接口。

物联网 (IoT)

由带有嵌入式传感器或处理器的连接物理对象组成的网络，这些传感器或处理器通过互联网或本地通信网络与其他设备和系统进行通信。有关更多信息，请参阅[什么是 IoT？](#)

可解释性

它是机器学习模型的一种特征，描述了人类可以理解模型的预测如何取决于其输入的程度。有关更多信息，请参阅[使用 AWS 实现机器学习模型的可解释性](#)。

IoT

参见[物联网](#)。

IT 信息库 (ITIL)

提供 IT 服务并使这些服务符合业务要求的一套最佳实践。ITIL 是 ITSM 的基础。

IT 服务管理 (ITSM)

为组织设计、实施、管理和支持 IT 服务的相关活动。有关将云运营与 ITSM 工具集成的信息，请参阅[运营集成指南](#)。

ITIL

请参阅[IT 信息库](#)。

ITSM

请参阅[IT 服务管理](#)。

L

基于标签的访问控制 (LBAC)

强制访问控制 (MAC) 的一种实施方式，其中明确为用户和数据本身分配了安全标签值。用户安全标签和数据安全标签之间的交集决定了用户可以看到哪些行和列。

登录区

landing zone 是一个架构精良的多账户 AWS 环境，具有可扩展性和安全性。这是一个起点，您的组织可以从这里放心地在安全和基础设施环境中快速启动和部署工作负载和应用程序。有关登录区的更多信息，请参阅[设置安全且可扩展的多账户 AWS 环境](#)。

大规模迁移

迁移 300 台或更多服务器。

LBAC

参见[基于标签的访问控制](#)。

最低权限

授予执行任务所需的最低权限的最佳安全实践。有关更多信息，请参阅 IAM 文档中的[应用最低权限许可](#)。

直接迁移

见 [7 R](#)。

小端序系统

一个先存储最低有效字节的系统。另请参见[字节顺序](#)。

下层环境

参见[环境](#)。

M

机器学习 (ML)

一种使用算法和技术进行模式识别和学习的人工智能。ML 对记录的数据 (例如物联网 (IoT) 数据) 进行分析和学习，以生成基于模式的统计模型。有关更多信息，请参阅[机器学习](#)。

主分支

参见[分支](#)。

恶意软件

旨在危害计算机安全或隐私的软件。恶意软件可能会破坏计算机系统、泄露敏感信息或获得未经授权的访问。恶意软件的示例包括病毒、蠕虫、勒索软件、特洛伊木马、间谍软件和键盘记录器。

托管服务

AWS 服务 它 AWS 运行基础设施层、操作系统和平台，您可以访问端点来存储和检索数据。亚马逊简单存储服务 (Amazon S3) Service 和 Amazon DynamoDB 就是托管服务的示例。这些服务也称为抽象服务。

制造执行系统 (MES)

一种软件系统，用于跟踪、监控、记录和控制车间将原材料转化为成品的生产过程。

MAP

参见[迁移加速计划](#)。

机制

一个完整的过程，在此过程中，您可以创建工具，推动工具的采用，然后检查结果以进行调整。机制是一种在运行过程中自我增强和改进的循环。有关更多信息，请参阅在 Well-Architect AWS ed 框架中[构建机制](#)。

成员账户

AWS 账户 除属于组织中的管理账户之外的所有账户 AWS Organizations。一个账户一次只能是一个组织的成员。

MES

参见[制造执行系统](#)。

消息队列遥测传输 (MQTT)

[一种基于发布/订阅模式的轻量级 machine-to-machine \(M2M\) 通信协议，适用于资源受限的物联网设备。](#)

微服务

一种小型独立服务，通过明确定义的 API 进行通信，通常由小型独立团队拥有。例如，保险系统可能包括映射到业务能力（如销售或营销）或子域（如购买、理赔或分析）的微服务。微服务的好处包括敏捷、灵活扩展、易于部署、可重复使用的代码和恢复能力。有关更多信息，请参阅[使用 AWS 无服务器服务集成微服务](#)。

微服务架构

一种使用独立组件构建应用程序的方法，这些组件将每个应用程序进程作为微服务运行。这些微服务使用轻量级 API 通过明确定义的接口进行通信。该架构中的每个微服务都可以更新、部署和扩展，以满足对应用程序特定功能的需求。有关更多信息，请参阅[在上实现微服务。AWS](#)

迁移加速计划 (MAP)

AWS 该计划提供咨询支持、培训和服务，以帮助组织为迁移到云奠定坚实的运营基础，并帮助抵消迁移的初始成本。MAP 提供了一种以系统的方式执行遗留迁移的迁移方法，以及一套用于自动执行和加速常见迁移场景的工具。

大规模迁移

将大部分应用程序组合分波迁移到云中的过程，在每一波中以更快的速度迁移更多应用程序。本阶段使用从早期阶段获得的最佳实践和经验教训，实施由团队、工具和流程组成的迁移工厂，通过自动化和敏捷交付简化工作负载的迁移。这是 [AWS 迁移策略](#) 的第三阶段。

迁移工厂

跨职能团队，通过自动化、敏捷的方法简化工作负载迁移。迁移工厂团队通常包括运营、业务分析师和所有者、迁移工程师、开发 DevOps 人员和冲刺专业人员。20% 到 50% 的企业应用程序组合由可通过工厂方法优化的重复模式组成。有关更多信息，请参阅本内容集中[有关迁移工厂的讨论](#)和[云迁移工厂](#)指南。

迁移元数据

有关完成迁移所需的应用程序和服务器器的信息。每种迁移模式都需要一套不同的迁移元数据。迁移元数据的示例包括目标子网、安全组和 AWS 账户。

迁移模式

一种可重复的迁移任务，详细列出了迁移策略、迁移目标以及所使用的迁移应用程序或服务。示例：使用 AWS 应用程序迁移服务重新托管向 Amazon EC2 的迁移。

迁移组合评测 (MPA)

一种在线工具，可提供信息，用于验证迁移到的业务案例。AWS Cloud MPA 提供了详细的组合评测（服务器规模调整、定价、TCO 比较、迁移成本分析）以及迁移计划（应用程序数据分析和数据收集、应用程序分组、迁移优先级排序和波次规划）。所有 AWS 顾问和 APN 合作伙伴顾问均可免费使用 [MPA 工具](#)（需要登录）。

迁移准备情况评测 (MRA)

使用 AWS CAF 深入了解组织的云就绪状态、确定优势和劣势以及制定行动计划以缩小已发现差距的过程。有关更多信息，请参阅[迁移准备指南](#)。MRA 是 [AWS 迁移策略](#) 的第一阶段。

迁移策略

用于将工作负载迁移到的方法 AWS Cloud。有关更多信息，请参阅此词汇表中的 [7 R](#) 条目和[动员组织以加快大规模迁移](#)。

ML

参见[机器学习](#)。

现代化

将过时的（原有的或单体）应用程序及其基础设施转变为云中敏捷、弹性和高度可用的系统，以降低成本、提高效率和利用创新。有关更多信息，请参阅[中的应用程序现代化策略](#)。AWS Cloud

现代化准备情况评估

一种评估方式，有助于确定组织应用程序的现代化准备情况；确定收益、风险和依赖关系；确定组织能够在多大程度上支持这些应用程序的未来状态。评估结果是目标架构的蓝图、详细说明现代化进程发展阶段和里程碑的路线图以及解决已发现差距的行动计划。有关更多信息，请参阅[中的评估应用程序的现代化准备情况](#) AWS Cloud。

单体应用程序（单体式）

作为具有紧密耦合进程的单个服务运行的应用程序。单体应用程序有几个缺点。如果某个应用程序功能的需求激增，则必须扩展整个架构。随着代码库的增长，添加或改进单体应用程序的功能也会变得更加复杂。若要解决这些问题，可以使用微服务架构。有关更多信息，请参阅[将单体分解为微服务](#)。

MPA

参见[迁移组合评估](#)。

MQTT

请参阅[消息队列遥测传输](#)。

多分类器

一种帮助为多个类别生成预测（预测两个以上结果之一）的过程。例如，ML 模型可能会询问“这个产品是书、汽车还是手机？”或“此客户最感兴趣什么类别的产品？”

可变基础架构

一种用于更新和修改现有生产工作负载基础架构的模型。为了提高一致性、可靠性和可预测性，Well-Architect AWS ed Framework 建议使用[不可变基础设施](#)作为最佳实践。

O

OAC

请参阅[源站访问控制](#)。

OAI

参见[源访问身份](#)。

OCM

参见[组织变更管理](#)。

离线迁移

一种迁移方法，在这种方法中，源工作负载会在迁移过程中停止运行。这种方法会延长停机时间，通常用于小型非关键工作负载。

OI

参见[运营集成](#)。

OLA

参见[运营层协议](#)。

在线迁移

一种迁移方法，在这种方法中，源工作负载无需离线即可复制到目标系统。在迁移过程中，连接工作负载的应用程序可以继续运行。这种方法的停机时间为零或最短，通常用于关键生产工作负载。

OPC-UA

参见[开放流程通信-统一架构](#)。

开放流程通信-统一架构 (OPC-UA)

一种用于工业自动化的 machine-to-machine (M2M) 通信协议。OPC-UA 提供了数据加密、身份验证和授权方案的互操作性标准。

运营级别协议 (OLA)

一项协议，阐明了 IT 职能部门承诺相互交付的内容，以支持服务水平协议 (SLA)。

运营准备情况审查 (ORR)

一份问题清单和相关的最佳实践，可帮助您理解、评估、预防或缩小事件和可能的故障的范围。有关更多信息，请参阅 Well-Architecte AWS d Frame [work 中的运营准备情况评估 \(ORR\)](#)。

操作技术 (OT)

与物理环境配合使用以控制工业运营、设备和基础设施的硬件和软件系统。在制造业中，OT 和信息技术 (IT) 系统的集成是[工业 4.0](#) 转型的重点。

运营整合 (OI)

在云中实现运营现代化的过程，包括就绪计划、自动化和集成。有关更多信息，请参阅[运营整合指南](#)。

组织跟踪

由 AWS CloudTrail 创建的跟踪记录组织 AWS 账户中所有人的所有事件 AWS Organizations。该跟踪是在每个 AWS 账户中创建的，属于组织的一部分，并跟踪每个账户的活动。有关更多信息，请参阅 CloudTrail 文档中的[为组织创建跟踪](#)。

组织变革管理 (OCM)

一个从人员、文化和领导力角度管理重大、颠覆性业务转型的框架。OCM 通过加快变革采用、解决过渡问题以及推动文化和组织变革，帮助组织为新系统和战略做好准备和过渡。在 AWS 迁移策略中，该框架被称为人员加速，因为云采用项目需要变更的速度。有关更多信息，请参阅[OCM 指南](#)。

来源访问控制 (OAC)

在中 CloudFront，一个增强的选项，用于限制访问以保护您的亚马逊简单存储服务 (Amazon S3) 内容。OAC 全部支持所有 S3 存储桶 AWS 区域、使用 AWS KMS (SSE-KMS) 进行服务器端加密，以及对 S3 存储桶的动态 PUT 和 DELETE 请求。

来源访问身份 (OAI)

在中 CloudFront，一个用于限制访问权限以保护您的 Amazon S3 内容的选项。当您使用 OAI 时，CloudFront 会创建一个 Amazon S3 可以对其进行身份验证的委托人。经过身份验证的委托人只能通过特定 CloudFront 分配访问 S3 存储桶中的内容。另请参阅[OAC](#)，其中提供了更精细和增强的访问控制。

或者

参见[运营准备情况审查](#)。

OT

参见[运营技术](#)。

出站 (出口) VPC

在 AWS 多账户架构中，一种处理从应用程序内部启动的网络连接的 VPC。[AWS 安全参考架构](#)建议使用入站、出站和检查 VPC 设置网络账户，保护应用程序与广泛的互联网之间的双向接口。

P

权限边界

附加到 IAM 主体的 IAM 管理策略，用于设置用户或角色可以拥有的最大权限。有关更多信息，请参阅 IAM 文档中的[权限边界](#)。

个人身份信息 (PII)

直接查看其他相关数据或与之配对时可用于合理推断个人身份的信息。PII 的示例包括姓名、地址和联系信息。

PII

查看[个人身份信息](#)。

playbook

一套预定义的步骤，用于捕获与迁移相关的工作，例如在云中交付核心运营功能。playbook 可以采用脚本、自动化运行手册的形式，也可以是操作现代化环境所需的流程或步骤的摘要。

PLC

参见[可编程逻辑控制器](#)。

PLM

参见[产品生命周期管理](#)。

策略

一个对象，可以在中定义权限（参见[基于身份的策略](#)）、指定访问条件（参见[基于资源的策略](#)）或定义组织中所有账户的最大权限 AWS Organizations（参见[服务控制策略](#)）。

多语言持久性

根据数据访问模式和其他要求，独立选择微服务的数据存储技术。如果您的微服务采用相同的数据存储技术，它们可能会遇到实现难题或性能不佳。如果微服务使用最适合其需求的数据存储，则可以更轻松地实现微服务，并获得更好的性能和可扩展性。有关更多信息，请参阅[在微服务中实现数据持久性](#)。

组合评测

一个发现、分析和确定应用程序组合优先级以规划迁移的过程。有关更多信息，请参阅[评估迁移准备情况](#)。

谓词

返回true或的查询条件false，通常位于子WHERE句中。

谓词下推

一种数据库查询优化技术，可在传输前筛选查询中的数据。这减少了必须从关系数据库检索和处理的数据量，并提高了查询性能。

预防性控制

一种安全控制，旨在防止事件发生。这些控制是第一道防线，帮助防止未经授权的访问或对网络的意外更改。有关更多信息，请参阅在 AWS 上实施安全控制中的[预防性控制](#)。

主体

中 AWS 可以执行操作和访问资源的实体。此实体通常是 IAM 角色的根用户或用户。AWS 账户有关更多信息，请参阅 IAM 文档中[角色术语和概念](#)中的主体。

隐私设计

一种贯穿整个工程化过程考虑隐私的系统工程方法。

私有托管区

私有托管区就是一个容器，其中包含的信息说明您希望 Amazon Route 53 如何响应一个或多个 VPC 中的某个域及其子域的 DNS 查询。有关更多信息，请参阅 Route 53 文档中的[私有托管区的使用](#)。

主动控制

一种[安全控制](#)措施，旨在防止部署不合规的资源。这些控件会在资源置备之前对其进行扫描。如果资源与控件不兼容，则不会对其进行配置。有关更多信息，请参阅 AWS Control Tower 文档中的[控制参考指南](#)，并参见在上实施安全[控制中的主动控制](#) AWS。

产品生命周期管理 (PLM)

在产品的整个生命周期中，从设计、开发和上市，到成长和成熟，再到衰落和移除，对产品进行数据和流程的管理。

生产环境

参见[环境](#)。

可编程逻辑控制器 (PLC)

在制造业中，一种高度可靠、适应性强的计算机，用于监控机器并实现制造过程自动化。

假名化

用占位符值替换数据集中个人标识符的过程。假名化可以帮助保护个人隐私。假名化数据仍被视为个人数据。

发布/订阅 (发布/订阅)

一种支持微服务间异步通信的模式，以提高可扩展性和响应能力。例如，在基于微服务的 [MES](#) 中，微服务可以将事件消息发布到其他微服务可以订阅的频道。系统可以在不更改发布服务的情况下添加新的微服务。

Q

查询计划

一系列步骤，例如指令，用于访问 SQL 关系数据库系统中的数据。

查询计划回归

当数据库服务优化程序选择的最佳计划不如数据库环境发生特定变化之前时。这可能是由统计数据、约束、环境设置、查询参数绑定更改和数据库引擎更新造成的。

R

RACI 矩阵

参见 [“负责任、负责、咨询、知情” \(RACI \)](#)。

勒索软件

一种恶意软件，旨在阻止对计算机系统或数据的访问，直到付款为止。

RASCI 矩阵

参见 [“负责任、负责、咨询、知情” \(RACI \)](#)。

RCAC

请参阅 [行和列访问控制](#)。

只读副本

用于只读目的的数据库副本。您可以将查询路由到只读副本，以减轻主数据库的负载。

重新架构师

见 [7 R](#)。

恢复点目标 (RPO)

自上一个数据恢复点以来可接受的最长时间。这决定了从上一个恢复点到服务中断之间可接受的数据丢失情况。

恢复时间目标 (RTO)

服务中断和服务恢复之间可接受的最大延迟。

重构

见 [7 R](#)。

区域

地理区域内的 AWS 资源集合。每一个 AWS 区域 都相互隔离，彼此独立，以提供容错、稳定性和弹性。有关更多信息，请参阅[指定 AWS 区域 您的账户可以使用的账户](#)。

回归

一种预测数值的 ML 技术。例如，要解决“这套房子的售价是多少？”的问题 ML 模型可以使用线性回归模型，根据房屋的已知事实（如建筑面积）来预测房屋的销售价格。

重新托管

见 [7 R](#)。

版本

在部署过程中，推动生产环境变更的行为。

搬迁

见 [7 R](#)。

更换平台

见 [7 R](#)。

回购

见 [7 R](#)。

故障恢复能力

应用程序抵御中断或从中断中恢复的能力。在中规划弹性时，[高可用性](#)和[灾难恢复](#)是常见的考虑因素。AWS Cloud有关更多信息，请参阅[AWS Cloud 弹性](#)。

基于资源的策略

一种附加到资源的策略，例如 AmazonS3 存储桶、端点或加密密钥。此类策略指定了允许哪些主体访问、支持的操作以及必须满足的任何其他条件。

责任、问责、咨询和知情 (RACI) 矩阵

定义参与迁移活动和云运营的所有各方的角色和责任的矩阵。矩阵名称源自矩阵中定义的责任类型：负责 (R)、问责 (A)、咨询 (C) 和知情 (I)。支持 (S) 类型是可选的。如果包括支持，则该矩阵称为 RASCI 矩阵，如果将其排除在外，则称为 RACI 矩阵。

响应性控制

一种安全控制，旨在推动对不良事件或偏离安全基线的情况进行修复。有关更多信息，请参阅在 AWS 上实施安全控制中的[响应性控制](#)。

保留

见 [7 R](#)。

退休

见 [7 R](#)。

旋转

定期更新[密钥](#)以使攻击者更难访问凭据的过程。

行列访问控制 (RCAC)

使用已定义访问规则的基本、灵活的 SQL 表达式。RCAC 由行权限和列掩码组成。

RPO

参见[恢复点目标](#)。

RTO

参见[恢复时间目标](#)。

运行手册

执行特定任务所需的一套手动或自动程序。它们通常是为了简化重复性操作或高错误率的程序而设计的。

S

SAML 2.0

许多身份提供商 (IdPs) 使用的开放标准。此功能支持联合单点登录 (SSO)，因此用户无需在 IAM 中为组织中的所有人创建用户即可登录 AWS Management Console 或调用 AWS API 操作。有关基于 SAML 2.0 的联合身份验证的更多信息，请参阅 IAM 文档中的[关于基于 SAML 2.0 的联合身份验证](#)。

SCADA

参见[监督控制和数据采集](#)。

SCP

参见[服务控制政策](#)。

secret

在中 AWS Secrets Manager，您以加密形式存储的机密或受限信息，例如密码或用户凭证。它由密钥值及其元数据组成。密钥值可以是二进制、单个字符串或多个字符串。有关更多信息，请参阅 [Secrets Manager 密钥中有什么？](#) 在 Secrets Manager 文档中。

安全控制

一种技术或管理防护机制，可防止、检测或降低威胁行为体利用安全漏洞的能力。安全控制主要有四种类型：[预防性](#)、[侦测](#)、[响应式](#)和[主动式](#)。

安全加固

缩小攻击面，使其更能抵御攻击的过程。这可能包括删除不再需要的资源、实施授予最低权限的最佳安全实践或停用配置文件中不必要的功能等操作。

安全信息和事件管理 (SIEM) 系统

结合了安全信息管理 (SIM) 和安全事件管理 (SEM) 系统的工具和服务。SIEM 系统会收集、监控和分析来自服务器、网络、设备和其他来源的数据，以检测威胁和安全漏洞，并生成警报。

安全响应自动化

一种预定义和编程的操作，旨在自动响应或修复安全事件。这些自动化可作为[侦探或响应式](#)安全控制措施，帮助您实施 AWS 安全最佳实践。自动响应操作的示例包括修改 VPC 安全组、修补 Amazon EC2 实例或轮换证书。

服务器端加密

在目的地对数据进行加密，由接收方 AWS 服务 进行加密。

服务控制策略 (SCP)

一种策略，用于集中控制 AWS Organizations 的组织中所有账户的权限。SCP 为管理员可以委托给用户或角色的操作定义了防护机制或设定了限制。您可以将 SCP 用作允许列表或拒绝列表，指定允许或禁止哪些服务或操作。有关更多信息，请参阅 AWS Organizations 文档中的[服务控制策略](#)。

服务端点

的入口点的 URL AWS 服务。您可以使用端点，通过编程方式连接到目标服务。有关更多信息，请参阅 AWS 一般参考 中的 [AWS 服务 端点](#)。

服务水平协议 (SLA)

一份协议，阐明了 IT 团队承诺向客户交付的内容，比如服务正常运行时间和性能。

服务级别指示器 (SLI)

对服务性能方面的衡量，例如其错误率、可用性或吞吐量。

服务级别目标 (SLO)

代表服务运行状况的目标指标，由服务[级别指标](#)衡量。

责任共担模式

描述您在云安全与合规方面共同承担 AWS 的责任的模型。AWS 负责云的安全，而您则负责云中的安全。有关更多信息，请参阅[责任共担模式](#)。

暹粒

参见[安全信息和事件管理系统](#)。

单点故障 (SPOF)

应用程序的单个关键组件出现故障，可能会中断系统。

SLA

参见[服务级别协议](#)。

SLI

参见[服务级别指标](#)。

SLO

参见[服务级别目标](#)。

split-and-seed 模型

一种扩展和加速现代化项目的模式。随着新功能和产品发布的定义，核心团队会拆分以创建新的产品团队。这有助于扩展组织的能力和服务，提高开发人员的工作效率，支持快速创新。有关更多信息，请参阅[中的分阶段实现应用程序现代化的方法](#)。 [AWS Cloud](#)

恶作剧

参见[单点故障](#)。

星型架构

一种数据库组织结构，它使用一个大型事实表来存储交易数据或测量数据，并使用一个或多个较小的维度表来存储数据属性。此结构专为在[数据仓库](#)中使用或用于商业智能目的而设计。

strangler fig 模式

一种通过逐步重写和替换系统功能直至可以停用原有的系统来实现单体系统现代化的方法。这种模式用无花果藤作为类比，这种藤蔓成长为一棵树，最终战胜并取代了宿主。该模式是由 [Martin Fowler](#) 提出的，作为重写单体系统时管理风险的一种方法。有关如何应用此模式的示例，请参阅[使用容器和 Amazon API Gateway 逐步将原有的 Microsoft ASP.NET \(ASMX \) Web 服务现代化](#)。

子网

您的 VPC 内的一个 IP 地址范围。子网必须位于单个可用区中。

监控和数据采集 (SCADA)

在制造业中，一种使用硬件和软件来监控有形资产和生产操作的系统。

对称加密

一种加密算法，它使用相同的密钥来加密和解密数据。

综合测试

以模拟用户交互的方式测试系统，以检测潜在问题或监控性能。你可以使用 [Amazon S CloudWatch ynthetic](#) 来创建这些测试。

T

标签

键值对，充当用于组织资源的元数据。AWS 标签可帮助您管理、识别、组织、搜索和筛选资源。有关更多信息，请参阅[标记您的 AWS 资源](#)。

目标变量

您在监督式 ML 中尝试预测的值。这也被称为结果变量。例如，在制造环境中，目标变量可能是产品缺陷。

任务列表

一种通过运行手册用于跟踪进度的工具。任务列表包含运行手册的概述和要完成的常规任务列表。对于每项常规任务，它包括预计所需时间、所有者和进度。

测试环境

参见[环境](#)。

训练

为您的 ML 模型提供学习数据。训练数据必须包含正确答案。学习算法在训练数据中查找将输入数据属性映射到目标（您希望预测的答案）的模式。然后输出捕获这些模式的 ML 模型。然后，您可以使用 ML 模型对不知道目标的新数据进行预测。

中转网关

中转网关是网络中转中心，您可用它来互连 VPC 和本地网络。有关更多信息，请参阅 AWS Transit Gateway 文档中的[什么是公交网关](#)。

基于中继的工作流程

一种方法，开发人员在功能分支中本地构建和测试功能，然后将这些更改合并到主分支中。然后，按顺序将主分支构建到开发、预生产和生产环境。

可信访问权限

向您指定的服务授予权限，该服务可代表您在其账户中执行任务。AWS Organizations 当需要服务相关的角色时，受信任的服务会在每个账户中创建一个角色，为您执行管理任务。有关更多信息，请参阅 AWS Organizations 文档中的[AWS Organizations 与其他 AWS 服务一起使用](#)。

优化

更改训练过程的各个方面，以提高 ML 模型的准确性。例如，您可以通过生成标签集、添加标签，并在不同的设置下多次重复这些步骤来优化模型，从而训练 ML 模型。

双披萨团队

一个小 DevOps 团队，你可以用两个披萨来喂食。双披萨团队的规模可确保在软件开发过程中充分协作。

U

不确定性

这一概念指的是不精确、不完整或未知的信息，这些信息可能会破坏预测式 ML 模型的可靠性。不确定性有两种类型：认知不确定性是由有限的、不完整的数据造成的，而偶然不确定性是由数据中固有的噪声和随机性导致的。有关更多信息，请参阅[量化深度学习系统中的不确定性指南](#)。

无差别任务

也称为繁重工作，即创建和运行应用程序所必需的工作，但不能为最终用户提供直接价值或竞争优势。无差别任务的示例包括采购、维护和容量规划。

上层环境

参见[环境](#)。

V

vacuum 操作

一种数据库维护操作，包括在增量更新后进行清理，以回收存储空间并提高性能。

版本控制

跟踪更改的过程和工具，例如存储库中源代码的更改。

VPC 对等连接

两个 VPC 之间的连接，允许您使用私有 IP 地址路由流量。有关更多信息，请参阅 Amazon VPC 文档中的[什么是 VPC 对等连接](#)。

漏洞

损害系统安全的软件缺陷或硬件缺陷。

W

热缓存

一种包含经常访问的当前相关数据的缓冲区缓存。数据库实例可以从缓冲区缓存读取，这比从主内存或磁盘读取要快。

暖数据

不常访问的数据。查询此类数据时，通常可以接受中速查询。

窗口函数

一个 SQL 函数，用于对一组以某种方式与当前记录相关的行进行计算。窗口函数对于处理任务很有用，例如计算移动平均线或根据当前行的相对位置访问行的值。

工作负载

一系列资源和代码，它们可以提供商业价值，如面向客户的应用程序或后端过程。

工作流

迁移项目中负责一组特定任务的职能小组。每个工作流都是独立的，但支持项目中的其他工作流。例如，组合工作流负责确定应用程序的优先级、波次规划和收集迁移元数据。组合工作流将这些资产交付给迁移工作流，然后迁移服务器和应用程序。

蠕虫

参见 [一次写入，多读](#)。

WQF

请参阅 [AWS 工作负载资格框架](#)。

一次写入，多次读取 (WORM)

一种存储模型，它可以一次写入数据并防止数据被删除或修改。授权用户可以根据需要多次读取数据，但他们无法对其进行更改。这种数据存储基础架构被认为是 [不可变的](#)。

Z

零日漏洞利用

一种利用未修补 [漏洞](#) 的攻击，通常是恶意软件。

零日漏洞

生产系统中不可避免的缺陷或漏洞。威胁主体可能利用这种类型的漏洞攻击系统。开发人员经常因攻击而意识到该漏洞。

僵尸应用程序

平均 CPU 和内存使用率低于 5% 的应用程序。在迁移项目中，通常会停用这些应用程序。

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。