为亚马逊 Neptune 应用 AWS Well-Architected 框架

AWS 规范性指导

AWS 规范性指导: 为亚马逊 Neptune 应用 AWS Well-Architected 框架

Table of Contents

为亚马逊 Neptune 应用 AWS Well-Architected 框架

Amazon Web Services（贡献者）

2026 年 1 月（文件历史记录）

您可以使用 Amazon Neptune 在 Amazon Web Services（AWS）上构建基于图形的解决方案。本指南为在规划 Neptune 部署时应用 AWS Well-Architected Framework 原则提供了规范指引。

Wel AWS l-Architected Framework 可帮助您为各种应用程序和工作负载构建安全、高性能、弹性和高效的基础架构。它还为您提供了评估架构和实施可扩展设计的一致方法。

Well AWS -Architected 框架围绕以下六大支柱构建：

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

本指南提供了来自 Well-Architect AWS ed Framework 设计支柱和最佳实践的信息，以及部署 Neptune 时需要记住的注意事项。 AWS

目标受众

本指南适用于设计和实施使用 AWS上的图表的解决方案的数据工程师、解决方案架构师和数据分析师。

目标

本指南可以帮助您和您的组织执行以下操作：

- 根据您的使用案例和查询模式，选择支持的部署选项和查询语言。
- 遵循 Wel AWS l-Architected 的设计模式，这将有助于提高灵活性和安全性。
- 设计查询以获得最佳性能并节省成本。

- 了解如何在生产环境中管理 Neptune 集群时提高运营效率。

卓越运营支柱

Well-Architect AWS ed Framework 的卓越运营支柱侧重于运行和监控系统，以及不断改进流程和程序。其中包括有效地支持开发和运行工作负载的能力，获取对运营的洞察，以及不断改进支持流程和程序以实现业务价值。您可以通过自我修复工作负载降低运营复杂性，这些工作负载无需人工干预即可检测和修复大多数问题。您可以通过遵循本节中描述的最佳实践来努力实现这一目标。当您的工作负载偏离预期行为时 APIs，使用 Amazon Neptune 指标和机制来正确响应。

本次对卓越运营支柱的讨论侧重于以下关键领域：

- 基础设施即代码（IaC）
- 变更管理
- 韧性策略
- 事件管理
- 合规性审计报告
- 日志记录和监控

使用 IaC 方法自动部署

使用 IaC 在 Neptune 上自动部署的最佳实践包括：

- 尽可能应用基础设施即代码 (IaC) 来部署 Neptune 集群。为了实现一致的环境配置，请使用AWS CloudFormation模板或 HashiCorp Terraform 为集群创建所有必需的资源。AWS Cloud Development Kit (AWS CDK)
- 尽可能自动执行 Neptune 操作程序，例如调整实例大小、添加或删除只读副本，或者对全局表进行手动失效转移。
- 将连接字符串存储在客户端外部。使用提取、转换和加载 (ETL) 流程来促进 blue/green 部署策略、灾难恢复 (DR) 以及向新集群的近乎零的停机迁移。连接字符串可以存储在 AWS Secrets Manager、Amazon DynamoDB 或任何可以动态更改它们的位置。
- 使用标签向 Neptune 资源添加元数据并基于标签跟踪使用情况。有关更多信息，请参阅标记 Amazon Neptune 资源。

频繁进行可逆的小规模更改

以下建议侧重于小的、可逆的更改，以最大限度地降低复杂性并降低工作负载中断的可能性：

- 将 IaC 模板和脚本存储在源代码控制服务中，例如 GitHub 或 GitLab。

> **Important**
>
> 不要在源代码管理中存储 AWS 凭据。

- 要求 IaC 部署才能使用持续集成和持续交付（CI/CD）服务，例如 AWS CodePipeline 或 AWS CodeBuild。这些服务会在包含临时 Neptune 集群的非生产环境中编译、测试和部署代码，然后再影响您的生产 Amazon Neptune 集群。
- 在将基础设施和应用程序查询部署到生产环境之前，先在较低的环境中对其进行测试。这将最大限度地减少中断的可能性，并有助于确保它们在您的工作负载和扩展下表现良好。

预测故障

自我修复的基础设施通过预测故障并尝试在没有干预的情况下解决任何问题来体现卓越运营。以下建议可以帮助您使用 Neptune 实现该成熟度：

- 创建使用 Amazon CloudWatch 指标监控数据库实例的 CPU 和内存使用情况并了解使用模式的监控计划。为应用程序日志中的关键指标和 Neptune 客户端响应创建 CloudWatch仪表板和警报。有关 CPU 利用率高或低指标的更多信息，请参阅 Neptune CloudWatch 文档中的使用在 Neptune 中监控数据库实例性能。

  如果您经常在查询中遇到 out-of-memory异常，可以考虑减少查询遍历的节点总数，或者尝试使用该X2系列中的实例，该实例的比例更高 RAM-to-CPU。
- 设置通知以监控 Neptune 集群的运行状况。例如，`BufferCacheHitRatio` 应始终处于高位（大于 99.9%），而 `MainRequestQueuePendingRequests` 应始终保持较低水平（理想情况下为 0，但取决于您的要求和延迟容忍度）。
- 考虑使用只读副本在 Neptune 内实现高可用性。您应该在与写入器实例不同的可用区中至少有两个只读副本，以确保在失效转移事件期间，实例始终可用于处理读取查询。
- 根据利用率指标自动扩展只读副本。有关更多信息，请参阅自动扩缩 Amazon Neptune 数据库集群中的副本数量。
- 测试您的数据库实例的故障转移，以了解对于您的使用案例而言，该过程需要多长时间。
- 如果您的应用程序需要在完全 AWS 区域 中断后存活下来，请考虑将全局数据库作为灾难恢复计划的一部分。

从所有操作故障中吸取教训

自我修复基础设施是一项长期的工作，当出现罕见问题或响应效果不如预期时，它会不断迭代发展。采取以下实践有助于集中精力实现该目标：

- 从所有故障中吸取教训，推动改进。
- 在团队和组织内部分享经验教训。如果组织内有多个团队使用 Neptune，请创建一个公共聊天室或用户群组，以便分享经验和最佳实践。

使用日志记录功能来监控未经授权或异常的活动

要观察异常性能和活动模式，请将日志存储在 Ama CloudWatch zon 日志中。考虑下面的最佳实践：

- 启用慢速查询日志记录。定期查看日志并诊断某些查询缓慢的原因。使用适用于 Gremlin、SPARQL 或 openCypher 的 Neptune explain 和 profile 端点，来深入了解这些查询缓慢的原因。
- 启用 Neptune 审核日志，并定期查看日志中是否存在未经授权的访问或异常情况。
- 如果您使用的是慢查询日志记录或审核日志记录，请启用发布到 CloudWatch 日志。这将帮助您避免实例上的磁盘空间不足。Neptune 实例的日志存储容量有限，超过日志空间时会覆盖较旧的日志文件。 CloudWatch 日志支持长期保留日志。 CloudWatch 日志中增强的监控功能将提高您查询日志和诊断问题的能力。
- 为了便于使用更好的审计日志分析工具，您可以将 Neptune 数据库集群配置为在 Logs 中将审计日志数据发布到日志组中 CloudWatch 。借助 CloudWatch 日志，您可以对日志数据进行实时分析，用于 CloudWatch 创建警报和查看指标，并使用 CloudWatch 日志将日志记录存储在高度耐用的存储中。有关更多信息，请参阅将 Nep tune 日志发布到 Ama CloudWatch zon 日志。
- Neptune 支持使用 AWS CloudTrail记录控制面板操作。有关更多信息，请参阅使用记录亚马逊 Neptune API 调用。 AWS CloudTrail

安全支柱

云安全 AWS 是重中之重。作为 AWS 客户，您可以受益于专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构。

安全是双方共同承担 AWS 的责任。责任共担模型将其描述为云的安全性和云中的安全性：

- 云安全 — AWS 负责保护在云 AWS 服务 中运行的基础架构 AWS 云。 AWS 还为您提供可以安全使用的服务。作为AWS 合规计划的一部分，第三方审计师定期测试和验证 AWS 安全的有效性。要了解适用于 Amazon Neptune 的合规性计划，请参阅合规性计划范围内的AWS 服务。
- 云端安全 — 您的责任由您 AWS 服务 使用的内容决定。您还需要对其他因素负责，包括您的数据的敏感性、您公司的要求以及适用的法律法规。有关数据隐私的更多信息，请参阅数据隐私常见问题。有关欧洲数据保护的信息，请参阅 AWS Shared Responsibility Model and GDPR 博客文章。

安全支柱可帮助您了解在使用 Neptune 时如何应用分担责任模型。以下主题说明如何配置 Neptune 以实现您的安全性和合规性目标。您还将学习如何使用其他 AWS 服务 方法来监控和保护您的 Neptune 资源。

安全支柱包括以下关键重点领域：

- 数据安全性
- 网络安全
- 身份验证和授权

实现数据安全

数据泄露和违规行为会使您的客户处于危险之中，并可能对您的公司造成严重的负面影响。以下最佳实践有助于保护您的客户数据免遭无意和恶意泄露：

- 集群名称、标签、参数组、 AWS Identity and Access Management (IAM) 角色和其他元数据不应包含机密或敏感信息，因为这些数据可能会出现在账单或诊断日志中。
- URIs 或者指向作为数据存储在 Neptune 中的外部服务器的链接不应包含用于验证请求的凭据信息。
- Neptune 加密的实例通过防止对基础存储进行未经授权的访问来帮助保护您的数据，提供了额外的一层数据保护。您可以使用 Neptune 加密来增强部署在云中的应用程序的数据保护。您还可以使用 Neptune 加密来满足静态数据的合规性要求。

要为新的 Neptune 数据库实例启用加密，请在 Neptune 控制台的 “启用加密” 部分中选择 “是”（默认选中），或者在中设置属性。AWS::Neptune::DBCluster::StorageEncrypted CloudFormation如果启用了加密，Neptune 将默认使用 Amazon Relational Database Service（Amazon RDS）AWS 托管式密钥，或者您可以创建客户自主管理型密钥。有关创建 Neptune 数据库实例的信息，请参阅创建新的 Neptune 数据库集群。有关更多详细信息，请参阅静态加密 Neptune 资源。您的自动快照和手动快照使用的加密方式与您为 Neptune 集群选择的加密方式相同。

- 使用 SPARQL 和 OpenCypher 语言时，练习正确的输入验证和参数化技术，以防止 SQL 注入及其他形式的攻击。避免构造使用字符串连接和用户提供的输入的查询。使用参数化查询或预准备语句将输入参数安全地传递到图形数据库。有关更多信息，请参阅 openCypher 参数化查询示例和 SPARQL 注入防御。
- 对于 Gremlin 语言，请使用 Gremlin 语言变体，而不是直接传递基于字符串的 Gremlin 脚本，以避免潜在的注入问题。

保护网络安全

只能 AWS上在虚拟私有云（VPC）中创建 Amazon Neptune 数据库集群。在 Neptune 1.4.6.0 之前，Neptune 数据库集群的终端节点只能在该 VPC 内访问。从 Neptune 1.4.6.0 及更高版本开始，可以将 Neptune 实例配置为可通过互联网公开访问。最佳做法是仅在非生产环境中使用此功能，以简化开发人员对 Neptune 的访问（尽管始终需要启用 IAM 身份验证才能实现公共可访问性）。如果您启用了公共可访问性，请考虑将数据库端口的入站安全组规则设置为仅限已知的 IP 地址流量。在生产环境或包含敏感数据的集群中，通过不允许公众访问和限制对 Neptune 数据库集群所在的 VPC 的访问来保护您的 Neptune 数据。有关更多信息，请参阅连接到您的 Amazon Neptune 图形。

为了保护传输中数据，Neptune 使用安全协议和密码，通过 HTTPS 强制执行与任何实例或集群端点的 SSL 连接。Neptune 为 Neptune 数据库实例提供 SSL 证书。Neptune SSL 证书仅支持集群端点、读取器端点和实例端点主机名。

如果您使用的是负载平衡器或代理服务器（例如 HAProxy），则必须使用 SSL 终止并在代理服务器上拥有自己的 SSL 证书。SSL 传递不起作用，因为提供的 SSL 证书与代理服务器主机名不匹配。有关使用 SSL 连接到 Neptune 端点的更多信息，请参阅使用 HTTP REST 端点连接到 Neptune 数据库实例。

实现身份验证和授权

要控制谁可以对 Neptune 数据库集群和数据库实例执行 Neptune 管理操作，请启用 IAM 数据库身份验证并使用 IAM 证书。使用 IAM 凭证连接到 AWS 时，您的 IAM 角色必须具有授予执行 Neptune 管

理操作所需的权限的 IAM 策略。请确保遵循最低权限原则，仅授予完成任务所需的权限。有关更多信息，请参阅使用不同类型的 IAM 策略控制对 Neptune 的访问权限和使用临时凭证进行 IAM 身份验证。

要控制谁可以连接到 Neptune 集群并查询数据，您可以使用 IAM 对 Neptune 数据库实例或数据库集群进行身份验证。如果您在 Neptune 数据库集群中启用 IAM 身份验证，则必须先对访问该数据库集群的任何人进行身份验证。有关启用 IAM 身份验证的步骤的更多信息，请参阅在 Neptune 中启用 IAM 数据库身份验证。

启用 IAM 数据库身份验证后，每个请求都必须使用 AWS 签名版本 4 进行签名。要了解如何向启用了 IAM 身份验证的所有 Neptune 端点发送已签名的请求，请参阅使用 AWS 签名版本 4 进行连接和签名。许多库和工具（例如 awscurl）已支持 AWS 签名版本 4。

为了与其他人互动 AWS 服务，Amazon Neptune 使用 IAM 服务相关角色。服务相关角色是一种独特类型的 IAM 角色，它与 Neptune 直接相关。服务相关角色由 Neptune 预定义，并包含服务代表您调用其他 AWS 服务 所需的所有权限。有关更多信息，请参阅为 Neptune 使用服务相关角色。

可靠性支柱

可靠性支柱包括工作负载在预期时正确、一致地执行其预期功能的能力。这包括在其全部生命周期内运行和测试工作负载的能力。

可靠的工作负载始于前期的软件和基础设施设计决策。您的架构选择将影响您在所有 Well-Architecte AWS d 支柱上的工作负载行为。针对可靠性，您必须遵循特定的模式。

可靠性支柱重点关注以下关键领域：

- 工作负载架构，包括服务配额和部署模式
- 变更管理
- 故障管理

了解 Neptune 服务配额

除中国外，Neptune 集群的最大容量可以增长到 128 TiB (TiB)，中国 AWS 区域 除外，其配额为 64 T GovCloud iB。

128TiB 配额足以在图表中存储大约 2000-4000 亿个对象。在带标签的属性图（LPG）中，对象是节点、边缘，或者是节点或边缘上的属性。在资源描述框架（RDF）图中，对象是四元组。

对于任何 Neptune Serverless 集群，您可以设置海王星容量单位的最小和最大数量 ()。NCUs每个 NCU 由 2GiB 的内存及关联的 vCPU 和网络组成。这些最小和最大 NCU 值适用于集群中的任何无服务器实例。您可以设置的最大 NCU 值为 128.0 NCUs，最低的最小值为 1.0。 NCUs通过观察 Amazon CloudWatch 指标`ServerlessDatabaseCapacity`来优化最适合您的应用程序的 NCU 范围，捕捉您经常遇到的范围，并将该范围内的不良行为或成本关联起来。`NCUUtilization`在许多工作负载中，1.0 NCU 的起点太低，在一段时间不活动后会导致行为不可靠。如果您发现工作负载的扩展速度不够快，请增加最小值， NCUs 以便在扩展时为最初的激增提供足够的处理能力。

每个 AWS 账户 区域对您可以创建的数据库资源数量都有配额。这些资源包括数据库实例和数据库集群。在您达到某一资源的限制时，再进行创建该资源的调用就会失败并引发异常。有些配额是软配额，可以通过请求增加。有关 Amazon Neptune 和 Amazon RDS、Amazon Aurora 和 Amazon DocumentDB（兼容 MongoDB）之间共享的配额列表，以及请求增加配额的链接（如果有），请参阅 Amazon RDS 中的配额。

了解 Neptune 部署模式

在 Neptune 数据库集群中，有一个主数据库实例和最多 15 个 Neptune 副本。主数据库实例支持读取和写入操作，并执行针对集群卷的所有数据修改。Neptune 副本连接到同一存储卷作为主数据库实例并仅支持读取操作。Neptune 副本可以从主数据库实例分载读取工作负载。

要实现高可用性，请使用只读副本。在不同的可用区中提供一个或多个只读副本实例可以提高可用性，因为只读副本充当主实例的失效转移目标。如果主实例失败，Neptune 将只读副本实例提升为主实例。当这种情况发生时，在提升的实例重启时会出现短暂的中断（通常少于 30 秒），在此期间，对主实例的读写请求将失败，同时引发异常。要获得最高的可靠性，请考虑在不同的可用区中使用两个只读副本。如果可用区 1 中的主实例离线，则可用区 2 中的实例会升级为主实例，但是在这种情况发生时它无法处理查询。因此，在过渡期间，需要可用区 3 中的实例来处理读取查询。

如果您使用的是 Neptune 无服务器，则所有可用区中的读取器和写入器实例将根据数据库负载相互独立地纵向扩展和缩减。您可以将读取器实例的提升层设置为 0 或 1，这样它就可以随写入器实例的容量一起纵向扩展和缩减。这样，它随时可以接管当前工作负载。

如果您的应用程序遍布全球或需要多区域故障转移，请考虑使用 Neptun e 全局数据库。Amazon Neptune 全球数据库跨越多个数据库 AWS 区域，可实现低延迟的全局读取，并在极少数中断影响整个数据库的情况下提供快速恢复。 AWS 区域 Neptune 全局数据库由一个区域中的主数据库集群和位于不同区域的多达五个辅助数据库集群组成。

管理和扩展 Neptune 集群

您可以使用 Neptune 自动扩缩来自动调整数据库集群中 Neptune 副本的数量，以满足您的连接和工作负载要求。通过自动扩缩，您的 Neptune 数据库集群可以应对工作负载的突然增加。工作负载减少时，自动扩缩会删除不必要的副本，这样您就不会为未使用的容量付费。请注意，启动新实例可能需要长达 15 分钟的时间，因此仅靠自动扩缩不足以应对需求的快速变化。

您只能对已经有一个主写入器实例和至少一个只读副本实例的 Neptune 数据库集群使用自动扩缩（请参阅 Amazon Neptune 数据库集群和实例）。此外，集群中的所有只读副本实例都必须处于可用状态。如果任何只读副本处于除可用状态以外的状态，则在集群中的每个只读副本都可用之前，Neptune 自动扩缩不会执行任何操作。

如果您遇到需求的快速变化，请考虑使用无服务器实例。无服务器实例可以在短时间内垂直扩展，而自动扩缩可以在较长的时间段内水平扩展。此配置提供了最佳的可扩展性，因为无服务器实例可以垂直扩展，而自动扩缩则实例化新的只读副本以处理超出单个无服务器实例最大容量的工作负载。有关 Amazon Neptune Serverless 容量扩展的更多信息，请参阅 Neptune 无服务器数据库集群中的容量扩展。

如果您的扩展需求在可预测的时间发生变化，则可以计划更改最小实例数、最大实例数和阈值，以更好地应对这些不断变化的需求。请记住至少提前 15 分钟安排横向扩展事件，以便这些实例在需要时可以上线。

通过使用数据库参数组中的参数，在 Amazon Neptune 中管理数据库配置。参数组就像是引擎配置值的容器，这些值可应用于一个或多个数据库实例。修改参数组中的集群参数时，了解静态参数和动态参数之间的区别，以及如何和何时进行应用。使用状态端点查看当前应用的配置。

管理备份和失效转移事件

Neptune 自动备份您的集群卷，并在备份保留期内保留备份的数据。Neptune 备份是连续且递增的，因此，您可以快速还原到备份保留期内的任何时间点。在创建或修改数据库集群时，可指定 1-35 天备份保留期。

要将备份保留期延长，您还可以为集群卷中的数据创建快照。存储快照会产生 Neptune 的标准存储费用。

在创建数据库集群的 Amazon Neptune 快照时，Neptune 创建集群的存储卷快照，同时备份集群的所有数据，而不仅仅是各个实例。您随后可通过从该数据库集群快照还原来创建新的数据库集群。恢复数据库集群时，您需要提供用于恢复的数据库集群快照的名称，然后提供恢复所创建的新数据库集群的名称。

测试您的系统如何响应失效转移事件。使用 Neptune API 强制执行失效转移事件。如果需要模拟数据库实例故障以进行测试，或是在失效转移进行之后将操作恢复到原始可用区，通过失效转移重启十分有用。有关更多信息，请参阅配置和管理多可用区部署。当您重启数据库写入器集群时，它会失效转移到备用副本。重启 Neptune 副本不会初始化失效转移。

设计客户端时要注重可靠性。测试它们在失效转移事件期间的行为。使用指数回退逻辑在客户端中实现重试逻辑。可以在 Amazon Neptune AWS Lambda 函数示例下的文档中找到实现此逻辑的代码示例。

如果您有一组适用于多个数据库引擎的常见备份要求，请考虑使用 AWS Backup。

性能效率支柱

Well-Architect AWS ed Framework 的性能效率支柱侧重于如何在摄取或查询数据时优化性能。性能优化是一个循序渐进的持续过程，包括：

- 确认业务需求
- 衡量工作负载性能
- 识别性能不佳的组件
- 优化组件以满足您的业务需求

性能效率支柱提供了特定于使用案例的准则，可以帮助识别要使用的正确图表数据模型和查询语言。还包括将数据摄取到 Amazon Neptune 以及从 Amazon Neptune 使用数据时应遵循的最佳实践。

性能效率支柱侧重于以下关键领域：

- 图形建模
- 查询优化
- 优化集群规模
- 写入优化

了解图形建模

了解标签属性图（LPG）和资源描述框架（RDF）模型之间的区别。在大多数情况下，这是一个偏好问题。但是，在一些使用案例中，一种模型比另一种模型更适合。如果您需要了解连接图表中两个节点的路径，请选择 LPG。如果要跨 Neptune 集群或其他图三元组存储联合数据，请选择 RDF。

如果您正在构建软件即服务（SaaS）应用程序或需要多租户的应用程序，请考虑在数据模型中纳入租户的逻辑分离，而不是为每个集群设置一个租户。要实现该类型的设计，您可以使用 SPARQL 命名图形和标注策略，例如在标签前添加客户标识符或添加代表租户标识符的属性键值对。确保您的客户端层注入这些值以保持逻辑分离。有关多租户建议的更多信息，请参阅运行 Amazon ISVs Neptune 数据库的多租户指南。

查询的性能取决于在处理查询时需要评估的图形对象（节点、边缘、属性）的数量。因此，图形模型可能会对应用程序的性能产生重大影响。尽可能使用精细标签，并仅存储实现路径确定或筛选所需的属性。要获得更高的性能，请考虑预计算图表的各个部分，例如创建汇总节点或连接公共路径的更直接的边缘。

尽量避免在具有相同标签的边缘数异常多的节点之间导航。此类节点通常有数千个边缘（其中大多数节点的边缘数仅在几十条左右）。结果是计算和数据复杂度大大增加。在某些查询模式中，这些节点可能不会出现问题，但我们建议您以不同的方式对数据进行建模以避免这种情况，尤其是在您需要通过节点作为中间步骤进行导航时。您可以使用慢速查询日志来帮助识别跨越这些节点的查询。您可能会观察到比平均查询模式更高的延迟和数据访问指标，尤其是在使用调试模式时。

如果您的用例支持，请 IDs 为节点和边使用确定性节点，而不是使用 Neptune 为其分配随机 GUID 值。 IDs通过 ID 访问节点是最有效的方法。

优化查询

在 LPG 模型上，openCypher 和 Gremlin 语言可以互换使用。如果性能是首要考虑因素，请考虑交替使用这两种语言，因为对于特定的查询模式，其中一种语言的性能可能优于另一种语言。

Neptune 正在转换为其替代查询引擎（DFE）。openCypher 仅在 DFE 上运行，但可以选择使用查询注释将 Gremlin 和 SPARQL 查询设置为在 DFE 上运行。考虑在激活 DFE 的情况下测试您的查询，并比较不使用 DFE 时的查询模式性能。

Neptune 针对事务型查询进行了优化，这些查询从单个节点或一组节点开始，然后从那里扇出，而不是评估整个图表的分析查询。对于您的分析查询工作负载，请使用 Neptune Analytics。Neptune Analytics 是需要快速迭代数据、分析和算法处理的研究、探索性或数据科学工作负载的理想选择。它还可以对图形数据执行矢量搜索，并可以直接从 Neptune 数据库实例加载数据。如果 Neptune Analytics 不能满足你的需求，你也可以考虑使用适用于 Pandas 的AWS SDK，或者将 nept une-export 与或 Amazon EMR 结合使用。AWS Glue

要识别模型和查询中的效率低下和瓶颈，请使用每种查询语言的`profile`和`explain` APIs 来获取查询计划和查询指标的详细解释。有关更多信息，请参阅 Gremlin profile、openCypher explain 和 SPARQL explain。

了解您的查询模式。如果图形中的不同边缘数太大，默认 Neptune 访问策略可能会效率低下。以下查询可能会变得非常低效：

- 未提供边缘标签时跨边缘向后导航的查询。
- 内部使用相同模式的子句（例如 Gremlin 中的 `.both()`），或者任何语言中用于删除节点的子句（这需要在不知道标签的情况下删除传入的边缘）。
- 在不指定属性标签的情况下访问属性值的查询。这些查询可能会变得非常低效。如果您的使用模式符合这种情况，请考虑启用 OSGP 索引（对象、主题、图表、谓词）。

使用慢速查询日志记录来识别慢速查询。查询速度慢可能是由于未优化的查询计划或不必要的大量索引查找造成的，这可能会增加成本。 I/O 适用于 Gremlin、SPARQL 或 openCypher 的 Neptune explain 和 profile 端点可以帮助您了解为什么这些查询速度很慢。原因可能包括：

- 与图中的平均节点相比，边缘数异常多的节点（例如，数千条和十条）可能会增加计算复杂性，从而延长延迟并增加资源消耗。确定这些节点的建模是否正确，或者是否可以改进访问模式，以减少必须遍历的边缘数。
- 未优化的查询将包含警告，提示特定步骤未经过优化。重写这些查询以使用优化步骤可能会提高性能。
- 冗余筛选条件可能会导致不必要的索引查找。同样，冗余模式可能会导致重复的索引查找，而这可以通过改进查询来优化（请参阅配置文件输出中的 `Index Operations - Duplication ratio`）。
- 某些语言（例如 Gremlin）没有强类型的数值，而是使用类型提升。例如，如果值为 55，则 Neptune 会查找与 55 等效的整数、长数、浮点数和其他数值类型的值。这会导致其他操作。如果您事先知道自己的类型匹配，则可以使用查询提示来避免这种情况。
- 您的图形模型会极大地影响性能。考虑通过使用更精细的标签或预计算多跳线性路径的快捷方式，来减少需要评估的对象数量。

如果仅靠查询优化无法满足性能要求，请考虑将各种缓存技术与 Neptune 配合使用来满足这些要求。

每个版本的 Neptune 性能都在不断提高。查看发行说明，了解每个版本的改进细节。考虑计划定期更新 Neptune 数据库集群，以帮助实现最佳性能。较新的版本也支持较新的实例。考虑升级到 1.4.5.0 或更高版本，以便能够使用这些实例。`r8g`有关这如何提高工作负载性能的更多信息，请参阅使用 Amazon N eptune v1.4.5 的 G AWS raviton4 r8g 实例的写入查询性价比提高 4.7 倍。

优化集群规模

根据您的并发和吞吐量要求调整集群规模。集群中每个实例可以处理的并发查询数量等于该实例上虚拟 CPUs (vCPUs) 数量的两倍。在所有 Worker 都被占用时到达的额外查询将放入服务器端队列。当工作线程可用时，将在 first-in-first-out (FIFO) 基础上处理这些查询。A `MainRequestQueuePendingRequests` mazon CloudWatch 指标显示每个实例的当前队列深度。如果此值经常大于零，请考虑选择一个具有更多 v 的实例CPUs。 如果队列深度超过 8,192，Neptune 将返回`ThrottlingException`错误。

每个实例的大约 65% 的 RAM 是为缓冲区缓存预留的。缓冲区缓存保存数据的工作数据集（不是整个图表；只是正在查询的数据）。要确定从缓冲区缓存而不是存储中提取的数据的百分比，请监控该

CloudWatch 指标`BufferCacheHitRatio`。如果该指标经常降至99.9%以下，请考虑尝试使用具有更多内存的实例，以确定它能否降低延迟和 I/O 成本。

只读副本的大小不必与您的写入器实例的大小相同。但是，繁重的写入工作负载可能会导致较小的副本落后并重启，因为它们无法跟上复制的速度。因此，我们建议创建等于或大于写入器实例的副本。

对只读副本使用自动扩缩时，请记住，将新的只读副本上线可能需要长达 15 分钟的时间。当客户端流量快速但可预测地增加时，考虑使用计划扩展来设置更高的只读副本的最小数量，以考虑该初始化时间。

无服务器实例支持多种不同的使用案例和工作负载。在以下情况下，考虑使用无服务器而不是预调配实例：

- 您一天中的工作负载经常波动。
- 您创建了一个新应用程序，但不确定工作负载的大小。
- 您正在进行开发和测试。

值得注意的是，按每 GB RAM 的美元计算，无服务器实例比同等预调配实例更昂贵。每个无服务器实例由 2GB 的 RAM 以及关联的 vCPU 和网络组成。在您的选项之间进行成本分析，以避免意外账单。通常，只有当您的工作负载每天只有几小时非常繁重，而其余时间几乎为零，或者当您的工作负载在一天中波动很大时，使用无服务器才能节省成本。

利用 Amazon Neptune 定价计算器，根据 queries-per-second (QPS) 要求等因素，帮助评估集群的正确配置。

优化写入

要优化写入情况，请考虑以下事项：

- Neptune 批量加载程序是初始加载数据库或附加到现有数据的最佳方式。Neptune 加载程序不是事务性的，也无法删除数据，因此如果您有这些要求，请不要使用它。
- 可以使用支持的查询语言进行事务更新。要优化写入 I/O 操作，请在每次提交时以 50-100 个对象为单位批量写入数据。对象是 LPG 中节点或边缘上的节点、边缘或属性，或者是 RDF 中的三元组存储或四元组。
- 每个连接的所有事务性 Neptune 写入操作都是单线程的。向 Neptune 发送大量数据时，考虑建立多个并行连接，每个连接都写入数据。当您选择 Neptune 预配置的实例时，实例大小与数字 v 相关联。CPUs Neptune 为实例上的每个 vCPU 创建两个数据库线程，因此在测试最佳并行化CPUs 时，从 v 数的两倍开始。 无服务器实例以大约每 4 NCUs 个 1 的速率缩放 v CPUs 的数量。

> **Note**
>
> 这不适用于批量加载 API，仅适用于直接连接。

- 即使任何时候只有一个连接ConcurrentModificationExceptions在写入数据，也要做好所有写入过程的规划和高效处理。设计客户端时，要确保在发生 `ConcurrentModificationExceptions` 时也能可靠运行。
- 如果您想删除所有数据，请考虑使用快速重置 API，而不是发出并发删除查询。与前者相比，后者将花费更长的时间并产生可观 I/O 的成本。
- 如果您要删除大部分数据，请考虑使用 neptune-export 将数据加载到新集群来导出要保留的数据。然后删除原始集群。

成本优化支柱

Well-Architecte AWS d Framework 的成本优化支柱侧重于避免不必要的成本。以下建议可以帮助您满足 Amazon Neptune 的成本优化设计原则和架构最佳实践。

成本优化支柱侧重于以下关键领域：

- 了解一段时间内的支出并控制资金分配
- 选择类型和数量正确的资源
- 在不超支的情况下进行扩展以满足业务需求

了解使用模式和所需的服务

如果您的数据模型具有清晰的图形结构，且您的查询需要探索关系并遍历多个跃点，则 Neptune 非常适合您的工作负载。图形数据库不适合以下模式：

- 主要是单跳查询（考虑您的数据是否可以更好地表示为对象的属性）
- JSON 或 BLOB 数据存储为属性
- 对数据集进行聚合的查询，例如计算大量节点的数值属性之和

考虑将多个专用数据库一起用于特定的访问模式是否可以满足您的所有需求。例如：

- 对于需要不那么频繁的复杂图形导航以及对单个节点属性进行高度并发检索的 API，最好使用一个或多个 Neptune、DynamoDB 或 Amazon DocumentDB 来呈现。
- 关系数据库可以与 Neptune 共存以维持您现有的功能，但只能将 Neptune 用于在关系数据库中性能不佳和扩展性不佳的多跳遍历。

了解与 Neptune 交互和补充 Neptune 的服务关联的成本，包括：

- Amazon Simple Storage Service（Amazon S3）存储成本，用于将数据文件批量加载到 Neptune 中
- 用于插入或更新插入查询、读取查询和 Neptune 流处理的 Lambda 函数
- 基于 Neptune 的 API 层用于与 Amazon API Gateway 中的客户端应用程序交互（而不是直接连接到数据库）或 AWS AppSync
- AWS Glue 用于在 Neptune 之间传输数据和从海王星传输数据的作业

- Amazon Kinesis 或 Amazon Managed Streaming for Apache Kafka（Amazon MSK）实例接收流数据，以近乎实时地将数据摄取到 Neptune。
- AWS Database Migration Service 用于将关系数据迁移到 Neptune
- Jupyter 笔记本和深度图库机器学习模型的 Amazon SageMaker Runtime 成本

选择资源时要注意成本

Neptune 定价基于每小时实例成本（或无服务器消耗的 Neptune 计算单元）、数据 I/O 和存储使用量。平均而言，实例占总成本的 85%，因此优化规模可能会带来显著的成本影响。调整实例大小的最佳方法是在各种实例上测试应用程序性能并比较以下因素：

- 该`MainRequestQueuePendingRequests` CloudWatch 指标是否一直保持在接近零的低水平？
- 该`BufferCacheHitRatio` CloudWatch 指标是否在大多数时间都保持在或高于 99.9%？
- 实例成本和相关数据 I/O 成本的成本和性能曲线是什么？ 如果实例容量过小，需要频繁地将缓冲区缓存与存储交换，则数据读取成本可能会显著增加。在这些情况下，`BufferCacheHitRatio` 将会频繁下降。

实例成本随相同实例系列内的大小线性扩展。`db.r6i.2xlarge` 实例的每小时成本是 `db.r6i.xlarge` 实例的两倍，资源分配也是后者的两倍。`db.r6i.24xlarge` 实例的每小时成本是 `db.r6i.xlarge` 实例每小时成本的 24 倍。

估算您必须支持的并发查询数量。您可以拥有 0 到 15 个只读副本来处理只读查询。如果您的要求因一天、一周或一个月的时间而异，则可以使用多个较小的实例按计划进行扩展。实例上的每个 vCPU 都提供两个用于处理并发查询的线程。三个 `db.r6i.xlarge` 只读副本（每个副本有 4 个 vCPU）可以处理 24 个并发查询。

如果您的流量改为以每秒查询数（QPS）来衡量，则必须进行实验以确定查询的平均延迟。Neptune 集群每秒可以支持的查询数等于 `vCPU × 2 × (1 second/average query latency)`。例如，如果您有 4 个 vCPU，查询延迟为 100 毫秒（0.1 秒），则 `QPS = 4 × 2 × (1s/0.1s) = 80 queries per second`。

对于持续、稳定和可预测的工作负载，预调配实例比无服务器实例更便宜。当您的工作负载每天只需几小时即可达到非常高的使用率（例如，`db.r6i.4xlarge`），而一天中的其余时间几乎没有流量（例如，1 个 Neptune 计算单位）时，无服务器可提供优化成本的机会。使用一个可纵向扩展几小时然后再缩减的无服务器实例，比全天使用预调配 `db.r6i.4xlarge` 实例更便宜。

考虑升级到 Neptune 1.4.5.0 或更高版本，并利用r8g实例以比老一代实例（例如或）更低的成本实现更好的读取和写入吞吐量。r7g r6g有关更多信息，请参阅使用 Amazon Neptune v1.4.5 的 G AWS raviton4 r8g 实例的写入查询性价比提高 4.7 倍（博客文章）。AWS

默认情况下，Neptune 集群是使用标准存储创建的（如果您使用控制台创建，则默认选择I/O-optimized storage). With I/O-optimized storage, you pay a slightly higher cost for storage and instances, but there are no I/O costs. This leads to more predictable recurring costs, but if your I/O usage is generally low, it may be more cost efficient to utilize standard storage. If you intend to load a lot of data initially, you can optimize cost by choosing I/O经过优化的存储，执行初始数据加载，然后切换到标准存储。存储类型仅影响计费模式，在 Neptune 数据库集群或实例配置中没有技术差异。您可以每 30 天更改一次存储类型。30 天后，查看您的 Neptune 详细费用，然后使用 Nept une 定价页面来计算使用优化后的费用是否会更高。I/O-optimized storage. If they would have been, continue to use standard storage, otherwise switch back to I/O

为您的工作负载选择最佳 Neptune 实例配置

如果您在 2025 年 7 月 15 日 AWS 账户 之前创建，则可以使用AWS 免费套餐进行 Neptune 的入门级实验。750 小时的 db.t3.medium 和 db.t4g.medium 实例免费使用时间足以让您很好地了解小规模下的 Neptune。在免费试用期结束后，您的集群仍将保留，但从那时起您将需要支付使用费用。

db.t3.medium和db.t4g.medium实例适用于不使用 OpenCypher、Graph Explorer 或各种生成式 AI 集成的低成本开发环境。这些实例的 RAM-to-vCPU比例 (2:1) 小于R系列实例 (8:1) 或X系列实例 (16:1)。这降低了比率，从而阻止了使用支持 OpenCypher 性能的 DFE 引擎统计信息、GenAI 集成（向 LLM 通报图形架构）和 Graph Explorer。使用T系列实例时，性能配置文件可能会有很大差异，特别是对于前面提到的工作负载。OutOfMemoryExceptions当查询在图表的很大一部分中导航时，这些实例还会增加出现的次数。要确定后一种情况是否可能受到影响，请检查BufferCacheHitRatio CloudWatch 指标。

我们强烈建议不要对T系列实例进行任何性能或负载测试，因为您可能会遇到不一致的结果，而这些结果并不代表生产环境。

当您的工作负载相当稳定且可预测时，预调配实例可为您提供最佳成本和性能组合。根据所需的请求并发性和查询复杂性选择实例大小。更高的并发性需要更多的 v CPUs。 查询复杂度越高，需要更多的 RAM。使用该MainRequestQueuePendingRequests CloudWatch 指标来确定前者的影响（大于零表示并发请求数多于可以处理的数量）。使用该BufferCacheHitRatio CloudWatch 指标来确定后者的影响。如果 RAM 使用率经常低于 99.9%，则表明 RAM 不足以容纳正在评估的图形的工作部分，从而导致更频繁地交换缓存。如果 R 系列实例提供了足够的并发性，但内存不足，请考虑尝试使用该X系列实例。

Neptune 文档中描述了无服务器实例的理想使用案例。如果您不确定预配置还是无服务器最适合您，并且成本是您的主要考虑因素，请在无服务器中测试您的工作负载以确定 NCUs 使用的数量，并将预配置 () 与无服务器 (`N hours × hourly provisioned cost`) 的成本进行比较。`sum of NCUs × hourly cost per NCU`如果您不确定同等大小的预调配实例，则一个 NCU 相当于大约 2GB 的 RAM 以及关联的 vCPU 和网络。如果您的预配置实例来自该`r6i`系列，则该比率为每 8 GB RAM 1 个 vCPU，或者 NCUs 4 个，再加上关联的网络。Amazon Neptune 定价计算器还提供比较，以帮助您确定最佳的成本配置。

在对主实例和副本实例使用无服务器时，请记住，升级层 0 和 1 中的只读副本将根据写 NCUs 入器实例进行扩展，以便在发生故障转移事件时可以正确扩展它们。根据您的哪个实例（写入器或读取器）接收的流量最多，为这些实例设置 NCU 限制。

在不需要集群全天候运行的环境中，可以考虑编写脚本，以便在不使用 Neptune 实例时将其关闭，并在需要使用之前重新启动它们。Neptune 实例将每 7 天自动重启一次，以确保应用所需的维护更新。如果您打算长时间关闭实例，请使用每周脚本再次将其关闭。

适当调整数据存储和传输的大小

更高效的查询（例如，需要触及图中较少节点、边和属性的查询）需要更少的 I/O 传输，并且由于需要的缓冲区缓存较少，因此有可能使用较小的实例。使用您的查询语言的 profile 或 explain 端点来优化查询，并考虑优化图表模型以提高查询性能。

Neptune 对大型字符串使用词典编码，该词典针对性能而非效率进行了优化。如果你的属性字符串很大 BLOBs、JSON 或者经常更改，可以考虑将它们存储在 Neptune 之外的亚马逊 S3、亚马逊 DynamoDB 或 Amazon DocumentDB 中，并且只在 Neptune 节点中存储引用。

在某些情况下，选择更大的实例大小可能更便宜。如果由于较低而导致 I/O 成本非常高`BufferCacheHitRatio`，则较大的缓冲区缓存可能会显著降低该成本。这是因为所有数据都将存放在缓存中，而不是经常从存储中交换并产生 I/O 传输速率。

Neptune 使用克隆 copy-on-write。在克隆以将图表拆分为多个分片时，不删除克隆集群上不需要的数据可能更有效，因为这将涉及创建新的数据页面，从而导致存储成本增加。克隆事件发生前未发生变化的数据将存在于两个集群之间共享的单个数据页面中，且仅对该单个副本收费。

请勿启用 OSGP 索引或使用 R5d 实例，除非您已测试确认它们对您的工作负载有重大影响。两者都是为极少发生的情况而设计的，它们可能会增加您的成本，而收益微乎其微或根本没有收益。

可持续发展支柱

可持续发展支柱侧重于最大限度地减少运行云工作负载对环境的影响。关键主题包括可持续发展责任共担模式、了解影响以及最大限度地使用以尽量减少所需资源和减轻下游影响。

可持续性支柱包含以下关键重点领域：

- 您的影响
- 可持续性目标
- 最大程度提高使用率
- 预测和采用更高效的新硬件和软件产品
- 使用托管服务
- 降低下游影响

本指南重点关注您的影响。有关其他可持续发展设计原则的更多信息，请参阅 Well-Architecte AWS d Framew ork。

您的选择和要求会对环境产生影响。如果您可以选择碳排放强度较低的 AWS 区域 ，且如果您的要求反映实际工作负载需求，而不仅仅是最大限度地提高正常运行时间和持久性，那么工作负载的可持续性就会提高。接下来的章节将讨论最佳实践和深思熟虑的注意事项，如果在您的工作负载设计和持续运营中采用这些实践和注意事项，将会对环境产生积极的影响。

AWS 区域 选择

AWS 区域 有些位于亚马逊可再生能源项目附近，或者位于电网公布的碳强度低于其他项目的地方。考虑可能适合您的工作负载的区域的可持续性影响，并将您的列表与 Neptune 可用的区域进行交叉引用。

基于用户行为模式的消费

根据用户的流量和行为调整资源消耗，有助于 AWS 最大限度地减少服务对环境的影响。设计解决方案时，考虑以下最佳实践：

- 监控CPUUtilization、MainRequestQueuePendingRequests、和之类的 Amazon CloudWatch 指标，TotalRequestsPerSec以确定您的需求何时为最高和最低，并确保在这段时间内您的集群资源大小合适。

- 在不使用非生产环境的时间段内，自动停止这些环境。有关更多信息，请参阅博客文章 Automate the stopping and starting of Amazon Neptune environment resources using resource tags。
- 如果您的流量模式变化频繁且不可预测，请考虑使用可随需求纵向扩展和缩减的 Neptune Serverless 实例，而不是使用为峰值流量预调配的实例。
- 除了业务连续性目标外，还可以考虑使您的服务水平协议与可持续性目标保持一致。放宽多区域灾难恢复、高可用性或长期备份保留等要求（尤其是对于非生产环境或非任务关键型工作负载）可以减少实现这些目标所需的资源量。

优化软件开发和架构模式

为防止浪费，请优化模型和查询，并共享计算资源，以便使用 Neptune 实例和集群中的所有可用资源。具体最佳实践包括：

- 让开发者共享 Neptune 实例和 Jupyter Notebook 应用程序实例，而不是各自创建自己的实例。通过使用多租户分区策略，在单个 Neptune 集群中为每位开发者提供自己的逻辑分区，并在单个 Jupyter 实例上为每位开发者创建单独的笔记本文件夹。
- 实现可最大限度利用资源并最大限度减少空闲时间的模式，例如使用并行线程加载数据，并将记录一起批处理到更大的事务中。
- 优化您的查询和图形模型，以最大限度地减少计算结果所需的资源。
- 对于 Gremlin 查询结果，请使用结果缓存功能最大限度地减少重新计算分页或经常重复的查询所花费的资源。
- 确保您的 Neptune 环境保持最新状态。最新版本的 Neptune 支持效率更高的最新亚马逊 EC2 实例，例如 Graviton。它们还改进了查询优化并修复了一些错误，从而减少了计算查询所需的资源量。

资源

参考

- AWS Well-Architected
- AWS WellArchited Framework 文档
- Neptune 最新更新
- 最佳实践：充分利用 Neptune
- 亚马逊 Neptune 定价计算器

博客文章

- 使用 Apache Gremlin 自动测试亚马逊 Neptune 数据访问权限 TinkerPop
- 使用资源标签自动停止和启动 Amazon Neptune 环境资源
- Amazon Neptune 数据面板操作的精细访问控制
- 使用 Amazon Neptune v1.4.5 的 AWS Graviton4 r8g 实例的写入查询性价比提高 4.7 倍
- Orca Security 如何优化其亚马逊 Neptune 数据库性能
- 使用 Amazon Neptune 公共终端节点更快地构建图形应用程序
- 全新 Amazon Neptune 引擎版本为 OpenCypher 查询性能提供高达 9 倍的速度和 10 倍的吞吐量

免费的“ AWS 技能大师”课程

- Amazon Neptune 入门
- 在 Amazon Neptune 上构建应用程序
- Amazon Neptune 的数据建模

贡献者

本指南的贡献者包括：

- Brian O'Keefe，Neptune 解决方案首席架构师 AWS
- Abhishek Mishra，Neptune 解决方案高级架构师， AWS
- Ganesh Sawhney，团队负责人——战略合作伙伴成功解决方案架构师， AWS
- Michael Havey，Neptune 解决方案高级架构师， AWS
- 凯文·菲利普斯，Neptune 解决方案架构师， AWS
- Melissa Kwok，Neptune 解决方案架构师， AWS
- Sakti Mishra，首席解决方案架构师 AWS
- Javed Ali，高级解决方案架构师， AWS

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
Neptune 版本更新	我们更新了文档，加入了有关 Amazon Neptune 1.4.6.0 及更高版本的信息。	2026年1月2日
初次发布	—	2023 年 9 月 27 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：将本地 Oracle 数据库迁移到 Amazon Aurora PostgreSQL 兼容版。
- 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：将本地 Oracle 数据库迁移到 AWS 云中的 Amazon Relational Database Service（Amazon RDS）for Oracle。
- 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：将客户关系管理（CRM）系统迁移到 Salesforce.com。
- 重新托管（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：将本地 Oracle 数据库迁移到 AWS 云中 EC2 实例上的 Oracle。
- 重新放置（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。您将服务器从本地平台迁移到同一平台的云服务中。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

请参阅托管服务。

ACID

请参阅原子性、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。它比主动-被动迁移更灵活，但工作量更大。

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

一种 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

请参阅人工智能。

AIOps

请参阅人工智能运营。

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？

人工智能操作 (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 AIOps AWS 迁移策略中使用的更多信息，请参阅操作集成指南。

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 (I AM) 文档 AWS中的 AB AC。

权威数据来源

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来源复制到其他位置，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

恶意机器人

一种旨在扰乱或伤害个人或组织的机器人。

BCP

请参阅业务连续性计划。

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。另请参阅字节顺序。

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应用程序版本（蓝色），在另一个环境中运行新应用程序版本（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交互的软件应用程序。有些机器人是有用或有益的，例如在互联网上索引信息的 Web 爬网程序。还有一些被称为恶意机器人的机器人，其目的是扰乱或伤害个人或组织。

僵尸网络

被恶意软件感染并受单方（称为僵尸网络控制者或僵尸网络操作者）控制的僵尸网络。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）。

紧急（break-glass）访问

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 AWS Well-Architected Guidance 中的 Implement break-glass procedures 指示器。

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

请参阅 AWS 云采用框架。

金丝雀部署

缓慢而渐进地向最终用户发布版本。当您确信无误后，即可部署新版本，并完全替换当前版本。

CCoE

请参阅云卓越中心。

CDC

请参阅更改数据捕获。

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韧性。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

请参阅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 云 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。云计算通常连接到边缘计算技术。

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 AWS 云中时通常会经历四个阶段：

- 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，创建着陆区、定义 CCo E、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 云 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

请参阅配置管理数据库。

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 Bitbucket Cloud。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 管线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）

一种 AI 领域，它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，Amazon SageMaker AI 为 CV 提供了图像处理算法。

配置偏移

对于工作负载而言，一种偏离预期状态的配置更改。这可能会导致工作负载变得不合规，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，以自定义您的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、提高生产力、提高代码质量和更快地交付。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

请参阅计算机视觉。

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-Architecte 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 云 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

请参阅数据库定义语言。

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 子字段使用多层人工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。

委派管理员

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。此账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

请参阅环境。

侦测性控制

一种安全控制，在事件发生后进行检测、记录日志和发出提醒。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

星型架构中的一种较小的表，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）

您用来最大程度地减少由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 Fr ame AWS work 中的 “工作负载灾难恢复：云端 AWS 恢复”。

DML

请参阅数据库操作语言。

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

请参阅灾难恢复。

偏差检测

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

请参阅开发价值流映射。

E

EDA

请参阅探索性数据分析。

EDI

请参阅电子数据交换。

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。与云计算比较时，边缘计算可以减少通信延迟并缩短响应时间。

电子数据交换（EDI）

组织之间业务文件的自动交换。有关更多信息，请参阅什么是电子数据交换。

加密

一种将人类可读的纯文本数据转换为加密文字的计算流程。

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

请参阅服务端点。

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。

环境

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

请参阅企业资源规划。

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

星型架构中的中心表。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。

快速失效机制

一种使用频繁且增量式的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 云，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

请参阅分支。

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，请参阅使用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。

少样本提示

在要求 LLM 执行类似任务之前，先向其提供少量示例，以演示任务和预期输出。此技术是上下文内学习的一种应用，其中模型可以从提示中嵌入的示例（样本）中学习。对于需要特定格式、推理或领域知识的任务，少样本提示可能非常有效。另请参阅零样本提示。

FGAC

请参阅精细访问控制。

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，通过更改数据捕获使用连续数据复制，在极短的时间内迁移数据，而非使用分阶段方法。目标是将停机时间降至最低。

FM

请参阅基础模型。

基础模型（FM）

一个大型深度学习神经网络，一直在广义和未标记数据的大量数据集上进行训练。 FMs 能够执行各种各样的一般任务，例如理解语言、生成文本和图像以及用自然语言进行对话。有关更多信息，请参阅什么是基础模型。

G

生成式人工智能

AI 模型的一个子集，这些模型已经过大量数据训练，可以使用简单的文本提示来创建新的内容和构件，例如图像、视频、文本和音频。有关更多信息，请参阅什么是生成式人工智能。

地理阻止

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工作流程，而基于中继的工作流程则是现代的、首选的方法。

黄金映像

系统或软件的快照，用作部署该系统或软件的新实例的模板。例如，在制造业中，黄金映像可用于在多个设备上预调配软件，并有助于提高设备制造操作的速度、可扩展性和生产效率。

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

帮助管理各组织单位的资源、策略和合规性的高级规则 (OUs)。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。侦测性护栏会检测策略违规和合规性问题，并生成提醒以进行修复。它们通过使用 AWS Config、、Amazon、 AWS Security Hub CSPM GuardDuty AWS Trusted Advisor、Amazon Inspector 和自定义 AWS Lambda 支票来实现。

H

HA

请参阅高可用性。

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。

保留数据

从用于训练机器学习模型的数据集中保留的一部分标注的历史数据。通过将模型预测与保留数据进行比较，您可以使用保留数据来评估模型性能。

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。

我

IaC

请参阅基础设施即代码。

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 云 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。

IIoT

请参阅工业物联网。

不可变基础设施

一种模型，可为生产工作负载部署新的基础设施，而不是更新、修补或修改现有基础设施。不可变基础设施本质上比可变基础设施更一致、更可靠、更可预测。有关更多信息，请参阅 AWS Well-Architected Framework 中的使用不可变基础设施进行部署最佳实践。

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 安全参考架构建议设置您的网络帐户，包括入站、出站和检查， VPCs 以保护您的应用程序与更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 Klaus Schwab 在 2016 年提出，指的是通过连接、实时数据、自动化、分析和 AI/ML 的进步来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调配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 (IIoT)

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 (IIoT) 数字化转型战略。

检查 VPC

在 AWS 多账户架构中，一种集中式 VPC，用于管理对 VPCs（相同或不同 AWS 区域）、互联网和本地网络之间的网络流量的检查。AWS 安全参考架构建议设置您的网络帐户，包括入站、出站和检查， VPCs 以保护您的应用程序与更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，请参阅使用机器学习模型的可解释性 AWS。

物联网

请参阅物联网。

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。

大语言模型（LLM）

一种基于大量数据进行预训练的深度学习 AI 模型。LLM 可以执行多项任务，例如回答问题、总结文档、将文本翻译成其他语言以及完成句子。有关更多信息，请参阅什么是 LLMs。

大规模迁移

迁移 300 台或更多服务器。

LBAC

请参阅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

请参阅 7 R。

小端序系统

一个先存储最低有效字节的系统。另请参阅字节顺序。

LLM

请参阅大型语言模型。

下层环境

请参阅环境。

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

请参阅分支。

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权限。恶意软件的示例包括病毒、蠕虫、勒索软件、木马、间谍软件和键盘记录器。

托管式服务

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。Amazon Simple Storage Service（Amazon S3）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、记录和控制在车间将原材料转化为成品的生产过程。

MAP

请参阅迁移加速计划。

机制

一个完整的过程，您可以在其中创建工具，推动工具的采用，然后检查结果以进行调整。机制是一种在运作过程中自我强化和改善的循环。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

请参阅制造执行系统。

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

一种小型的独立服务，通过明确的定义进行通信 APIs ，通常由小型的独立团队拥有。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通过定义明确的接口进行通信。 APIs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、开发人员和冲刺 DevOps 领域的专业人员。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，提供了用于验证迁移到 AWS 云的业务案例的信息。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

将工作负载迁移到 AWS 云的方法。有关更多信息，请参见术语表中的 7 R 词条，以及动员您的组织以加快大规模迁移。

ML

请参阅机器学习。

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，请参阅在 AWS 云中实现应用程序现代化的策略。

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，请参阅在 AWS 云中评估应用程序的现代化准备情况。

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

请参阅迁移组合评测。

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？”

可变基础设施

一种用于更新和修改生产工作负载的现有基础设施的模型。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

请参阅来源访问控制。

OAI

请参阅来源访问身份。

OCM

请参阅组织变革管理。

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

请参阅运营集成。

OLA

请参阅运营级别协议。

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

请参阅开放流程通信 – 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 提供了一个包含数据加密、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）

一份问题核对清单和关联的最佳实践，可帮助您了解、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 AWS Well-Architected Framework 中的运营准备情况审查（ORR）。

运营技术（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 转型的关键重点。

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 此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。

ORR

请参阅运营准备情况审查。

OT

请参阅运营技术。

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 安全参考架构建议设置您的网络帐户，包括入站、出站和检查， VPCs 以保护您的应用程序与更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

请参阅个人身份信息。

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

请参阅可编程逻辑控制器。

PLM

请参阅产品生命周期管理。

policy

一个对象，可以定义权限（请参阅基于身份的策略）、指定访问条件（请参阅基于资源的策略）或定义 AWS Organizations 的组织中所有账户的最大权限（请参阅服务控制策略）。

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。

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 `false` 的查询条件，通常位于 `WHERE` 子句中。

谓词下推

一种数据库查询优化技术，可在传输之前筛选查询中的数据。这将减少从关系数据库检索和处理的数据量，并提高查询性能。

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

一种在整个开发过程中都考虑隐私的系统工程方法。

私有托管区

一个容器，其中包含有关您希望 Amazon Route 53 如何响应针对一个或多个 VPCs域名及其子域名的 DNS 查询的信息。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，旨在防止部署不合规资源。这些控制会在资源预置之前对其进行扫描。如果资源与控制不兼容，则不会预置它。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）

对产品在其整个生命周期内的数据和流程的管理，从设计、开发和发布，到增长和成熟，再到衰退和淘汰。

生产环境

请参阅环境。

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。

提示串接

使用一个 LLM 提示的输出作为下一个提示的输入，以生成更好的响应。该技术用于将复杂的任务分解为子任务，或者迭代地完善或扩展初步响应。它有助于提高模型响应的准确性和相关性，并允许获得更精细的个性化结果。

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。

publish/subscribe (pub/sub)

一种支持微服务间异步通信的模式，可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

一系列用于访问 SQL 关系数据库系统中的数据的步骤，类似于指令。

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

请参阅责任、问责、咨询和知情（RACI）。

RAG

请参阅检索增强生成。

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

请参阅责任、问责、咨询和知情（RACI）。

RCAC

请参阅行列访问控制。

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

请参阅 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

请参阅 7 R。

Region

地理区域内的 AWS 资源集合。每一个 AWS 区域 都相互隔离，相互独立，以提供容错、稳定性和弹性。有关更多信息，请参阅指定您的账户可以使用的 AWS 区域。

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

请参阅 7 R。

版本

在部署过程中，推动生产环境变更的行为。

重新放置

请参阅 7 R。

更换平台

请参阅 7 R。

重新购买

请参阅 7 R。

韧性

应用程序抵御中断或从中断中恢复的能力。在 AWS 云中规划韧性时，高可用性和灾难恢复是常见的考虑因素。有关更多信息，请参阅 AWS 云 韧性。

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

请参阅 7 R。

停用

请参阅 7 R。

检索增强生成（RAG）

一种生成式人工智能技术，其中 LLM 在生成响应之前引用其训练数据来源之外的权威数据来源。例如，RAG 模型可以对组织的知识库或自定义数据执行语义搜索。有关更多信息，请参阅什么是 RAG。

轮换

定期更新密钥以使攻击者更难访问凭证的过程。

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

请参阅恢复点目标。

RTO

请参阅恢复时间目标。

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 管理控制台 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

请参阅监督控制和数据采集。

SCP

请参阅服务控制策略。

机密密钥

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 文档中的什么是 Amazon Secrets Manager 密钥？。

安全设计

一种在整个开发过程中都考虑安全的系统工程方法。

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。安全控制有以下四种类型：预防性、检测性、响应性和主动性。

安全固化

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

一种预定义的程序化操作，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 实例或轮换凭证。

服务器端加密

由接收数据的人在目的地对数据 AWS 服务 进行加密。

服务控制策略（SCP）

一种策略，用于集中控制组织中所有账户的权限 AWS Organizations。 SCPs 定义防护措施或限制管理员可以委托给用户或角色的操作。您可以使用 SCPs 允许列表或拒绝列表来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。

服务水平指示器（SLI）

对服务性能方面的衡量，例如错误率、可用性或吞吐量。

服务水平目标（SLO）

代表服务运行状况的目标指标，由服务水平指示器衡量。

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。

SIEM

请参阅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

请参阅服务水平协议。

SLI

请参阅服务水平指示器。

SLO

请参阅服务水平目标。

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，请参阅在 AWS 云中实现应用程序现代化的分阶段方法。

SPOF

请参阅单点故障。

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事务数据或测量数据，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。

监督控制和数据采集（SCADA）

在制造业中，一种使用硬件和软件来监控实物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 ynthetics 来创建这些测试。

系统提示

一种为 LLM 提供上下文、说明或准则以指导其行为的技术。系统提示有助于设置上下文并制定与用户交互的规则。

T

标签

键值对，用作组织资源的元数据。 AWS 标签有助于您管理、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

请参阅环境。

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

一个网络传输中心，可用于将您的网络 VPCs 和本地网络互连。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

请参阅环境。

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

两者之间的连接 VPCs ，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

一种对与当前记录有某种关联的一组行执行计算的 SQL 函数。窗口函数对于处理任务很有用，例如计算移动平均值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。

WORM

请参阅一次写入多次读取。

WQF

请参阅 AWS 工作负载资格鉴定框架。

一次写入多次读取（WORM）

一种存储模型，可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，但无法对其进行更改。此数据存储基础设施被认为不可变。

Z

零日漏洞利用

一种利用零日漏洞的攻击，通常为恶意软件。

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。

零样本提示

为 LLM 提供执行任务的说明，但没有可以帮助指导的示例（样本）。LLM 必须使用预先训练的知识来处理任务。零样本提示的有效性取决于任务的复杂性和提示的质量。另请参阅少样本提示。

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。