使用 AWS 无服务器服务集成微服务

AWS 规范性指导

AWS 规范性指导: 使用 AWS 无服务器服务集成微服务

Table of Contents

使用 AWS 无服务器服务集成微服务

Tabby Ward、Abhishek Agawane 和 Matt Kahn，Amazon Web Services

2025 年 9 月（文档历史记录）

实现组织软件现代化的关键环节在于选择正确的架构模式，从而实现敏捷性并快速响应不断变化的业务需求。在某些应用中，单体架构是常见的选择。然而，对于许多组织而言，当使用案例与微服务带来的优势相契合时，将单体应用重构为微服务可能是一种有效的现代化策略。

微服务与单体架构并非互斥，许多成功的企业同时采用这两种模式，其中模块化单体架构服务于某些领域，微服务则处理其他领域。

当微服务成为架构的一部分时，可能会调用多个服务来获取一项业务交易的数据。实施这些集成需要精心设计，以应对数据一致性、延迟和运营复杂性等潜在挑战。如果微服务集成得当，可带来诸多优势，例如单独扩缩、开发速度提升以及潜在的成本优化。

该指南是内容系列的一部分，该系列涵盖了推荐的应用程序现代化方法 AWS。该系列还包括：

- 实现应用程序现代化的策略 AWS Cloud
- 分阶段实现应用程序现代化的方法 AWS Cloud
- 评估 AWS Cloud中应用程序现代化的准备情况
- 将单体分解为微服务

目标受众

本指南适用于确定微服务适用于其特定使用案例的应用程序所有者、业务所有者、架构师、技术负责人和项目经理。该指南介绍了微服务之间同步和异步通信的几种模式，即使用无服务器（ AWS 服务 例如使用无服务器） AWS Lambda 和 Amazon API Gateway 来实现自主性和可扩展性。

目标

通过使用本指南集成新的微服务，您可以高效地将组织的架构转变为微服务架构。这有助于通过高可扩展性、增强的弹性、持续交付和故障隔离，快速适应不断变化的业务需求。微服务架构还可助力加快创新速度，因为每个微服务都可以单独部署和测试。

微服务架构还有助于缩短产品或服务的上市时间，因为每项微服务都有一个独立的代码库，可以更轻松、更快地添加新功能并对其进行迭代。

安全性

您必须妥善保护微服务以保护服务和数据的完整性，同时确保安全措施不会对应用程序的性能造成负面影响。

在微服务环境中，必须考虑每项服务如何对来自外部客户端或其他微服务的请求进行身份验证和授权。还要考虑每项服务将如何安全地访问其他 AWS 服务。

AWS 服务 应通过范围狭窄 AWS Identity and Access Management (IAM) 角色授予访问权限。承担 IAM 角色会为微服务提供短期 IAM 凭证，形式包括访问密钥、访问密码和会话令牌。各种软件开发套件 (SDKs) 使用AWS 签名版本 4 (Sigv4) 对 AWS 服务 请求进行签名。

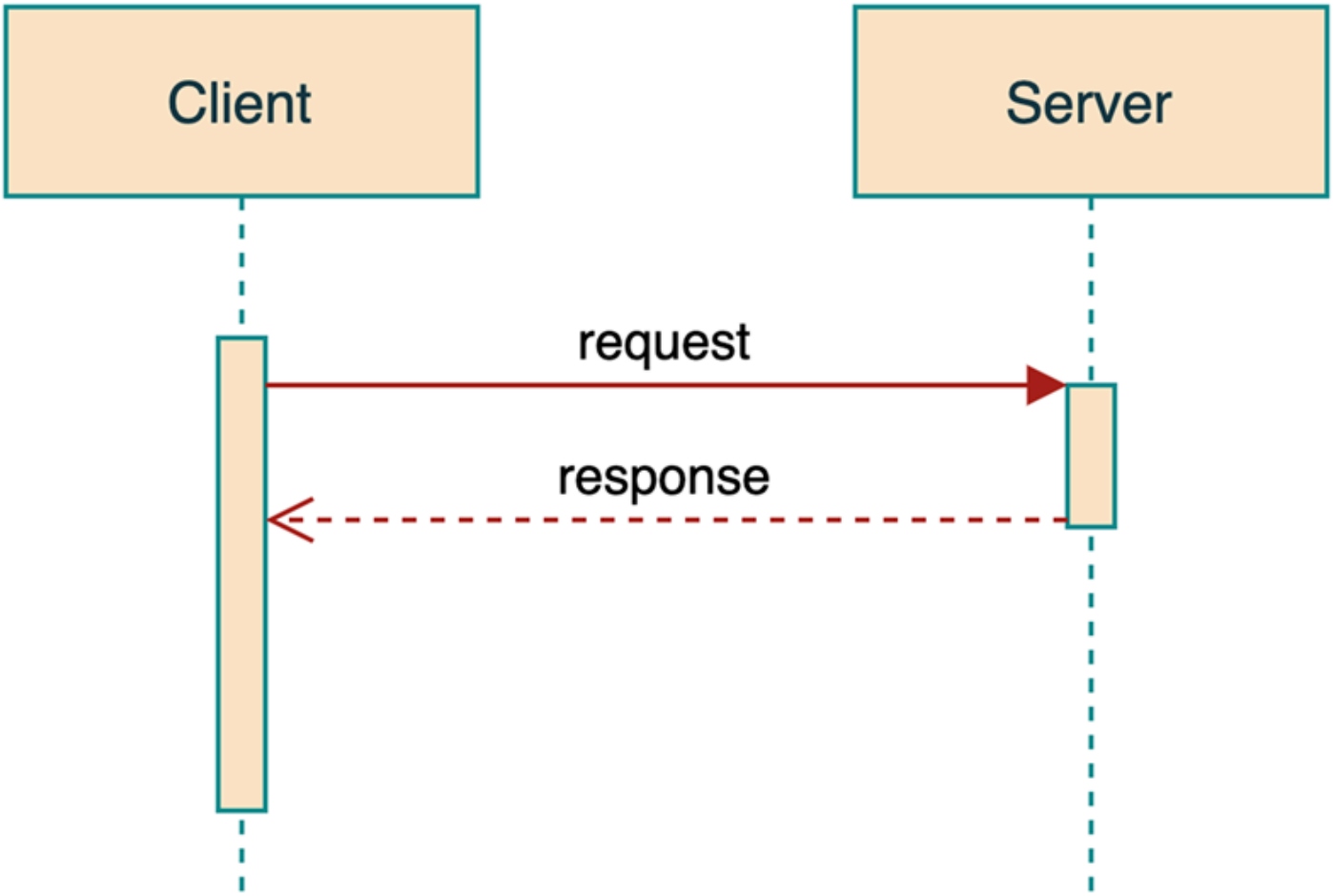
通信模式

在微服务架构中，通信主要以两种模式进行：同步和异步。在同步通信中，调用方需等待响应后再继续，类似于实时 HTTP REST API 调用。异步通信遵循基于消息的模式，其中调用方会继续处理而不等待响应，例如使用消息队列时。以下章节详细探讨了每种模式的实施、优点和使用案例。

主题

同步通信

在同步通信中，客户端会向服务发起请求，如下图所示。示例包括获取信息的请求（例如 HTTP GET 请求）或更改数据的请求（例如 HTTP PUT 请求）。无论哪种情况，客户端都需要等待服务器做出响应后再继续。同步调用对大多数开发人员而言并不陌生，其实施和故障排查都较为简单，在许多情况下更是被广泛接受的通信标准。



同步通信的优势包括：

- 流量控制可预测 – 确定性执行和清晰的请求-响应循环，相比异步通信更易于理解。
- 强一致性 – 即时确认数据更改和状态更新。
- 错误处理简单 – 直接传播错误和异常。
- 易于调试 – 直接跟踪和监控请求。
- 协议支持 – 诸如 HTTP、REST 等成熟协议，使得实施过程变得简单明了。

同步通信存在一些缺点：

- 紧密耦合 – 服务之间对彼此可用性的直接依赖关系。
- 网络影响 – 因持续保持连接而导致网络负载增加。
- 资源利用率 – 因维持连接状态而导致内存使用量增加。
- 级联故障 – 某项服务中的问题能够在系统中迅速蔓延。

异步通信

相反，在异步通信中，客户端向服务发出请求，但无法立即收到响应。这种情况下，客户端通常只会收到请求已被接受的确认信息。

同步通信的优势包括：

- 事件驱动型架构支持 – 天然契合事件溯源与命令查询责任分割（CQRS）模式。
- 资源管理更出色 – 服务能够根据其容量处理请求。
- 故障隔离经改进 – 服务解耦，可防止级联故障。
- 峰值负载处理 – 通过消息队列更好地处理流量峰值。

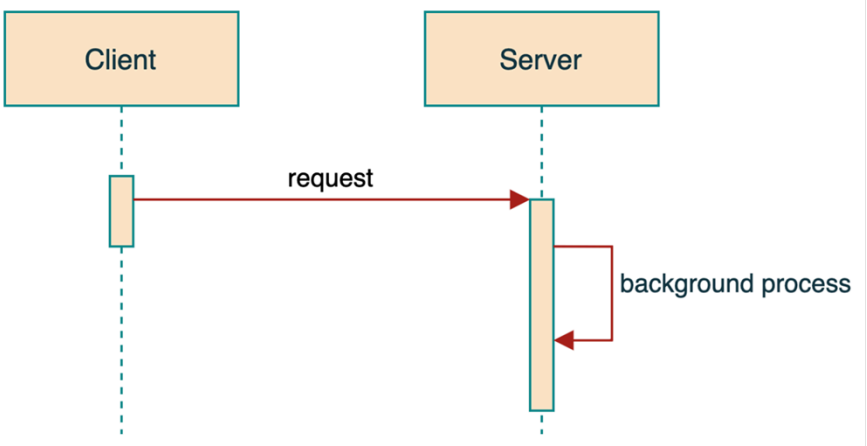
缺点包括复杂性。例如：

- 如果客户端需要获取异步操作的结果，则需要付出更多努力，才能实施获取或接收该结果的机制。
- 对异步操作进行故障排查可能更加困难，因为故障排查需要检查多个系统的日志。
- 测试异步操作可能更加困难，因为需要协调多项系统和服务，才能执行测试。

异步通信的方法包括即发即弃、声明检查、回调和双向通信。

即发即弃

在即发即弃模式中，客户端向服务器发出请求并同步收到一条确认消息，表明服务器已收到消息并将对其进行处理。然而，实际处理尚未进行，客户端无法知晓处理将在何时以何种方式完成。下图阐明了此模式。



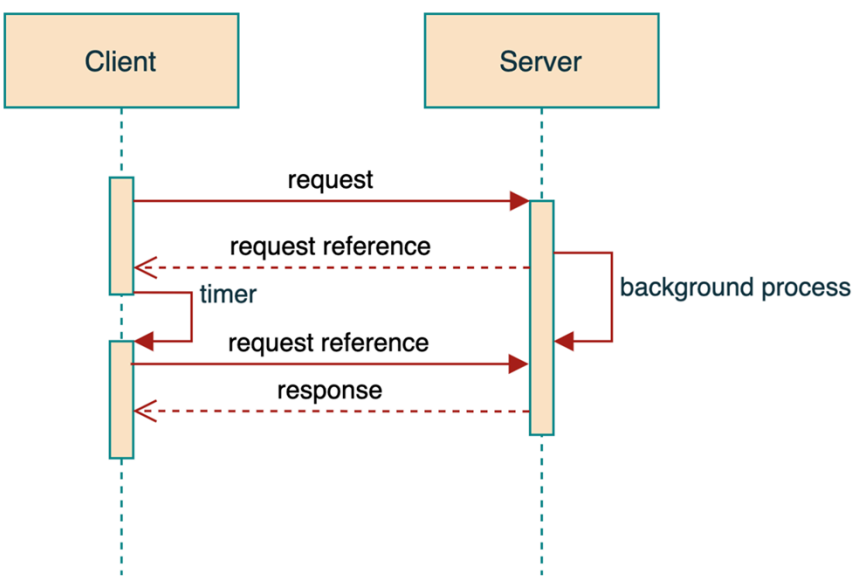
在此情况下，服务不应在对象被持久化保留之前发送确认。这种持久化可以作为数据库写入操作实施，也可以通过将项目放入队列来实施。

其他注意事项：

- 实施幂等性以处理重复消息。也就是说，每则消息只能被处理一次。
- 考虑针对处理失败的情况使用死信队列。
- 监控消息处理成功率。

声明检查

如果客户需要服务调用的结果，则可以构建服务，使其在收到请求时发出声明检查。下图阐明了此模式。声明检查充当服务在其确认中返回的标识符。客户端可以在后续阶段使用此标识符查询请求状态，并在请求完成时检索结果。



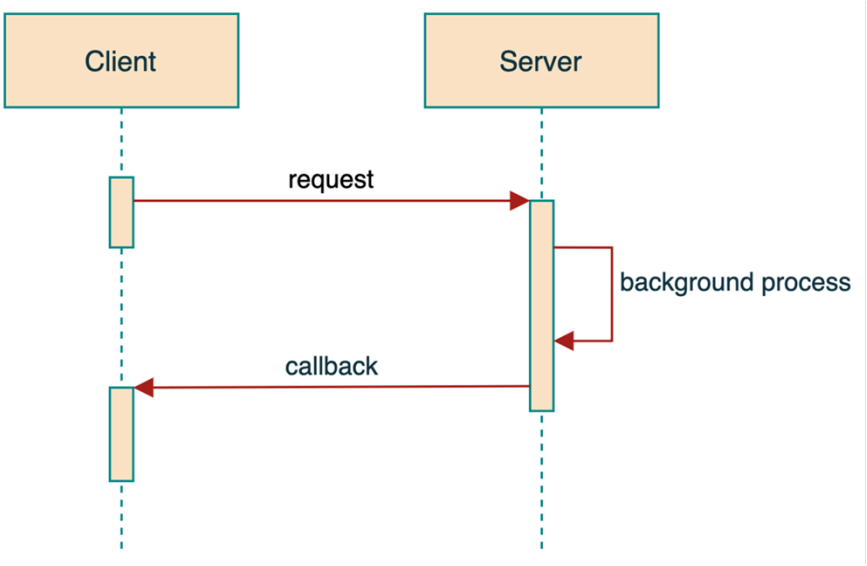
客户端必须实施一种机制来轮询结果。这可以是自动的（例如，可以每 n 分钟执行一次检查），也可以是手动的，即为了响应其他事件或用户操作而执行检查。实施声明检查模式的服务应明确声明检查的有效时间。

最佳实践：

- 为轮询实施指数回退。
- 为声明检查设置适当的生存时间（TTL）。
- 提供状态端点用于跟踪进度。

回调

在回调模式中，客户端向服务发起请求，并提供一个位置供服务在处理完成后联系。客户端不会等待结果，处理仍将继续。服务负责在处理完成后联系该位置并提供结果。常见的响应位置类型是 REST APIs 或队列。下图展示了此回调模式。

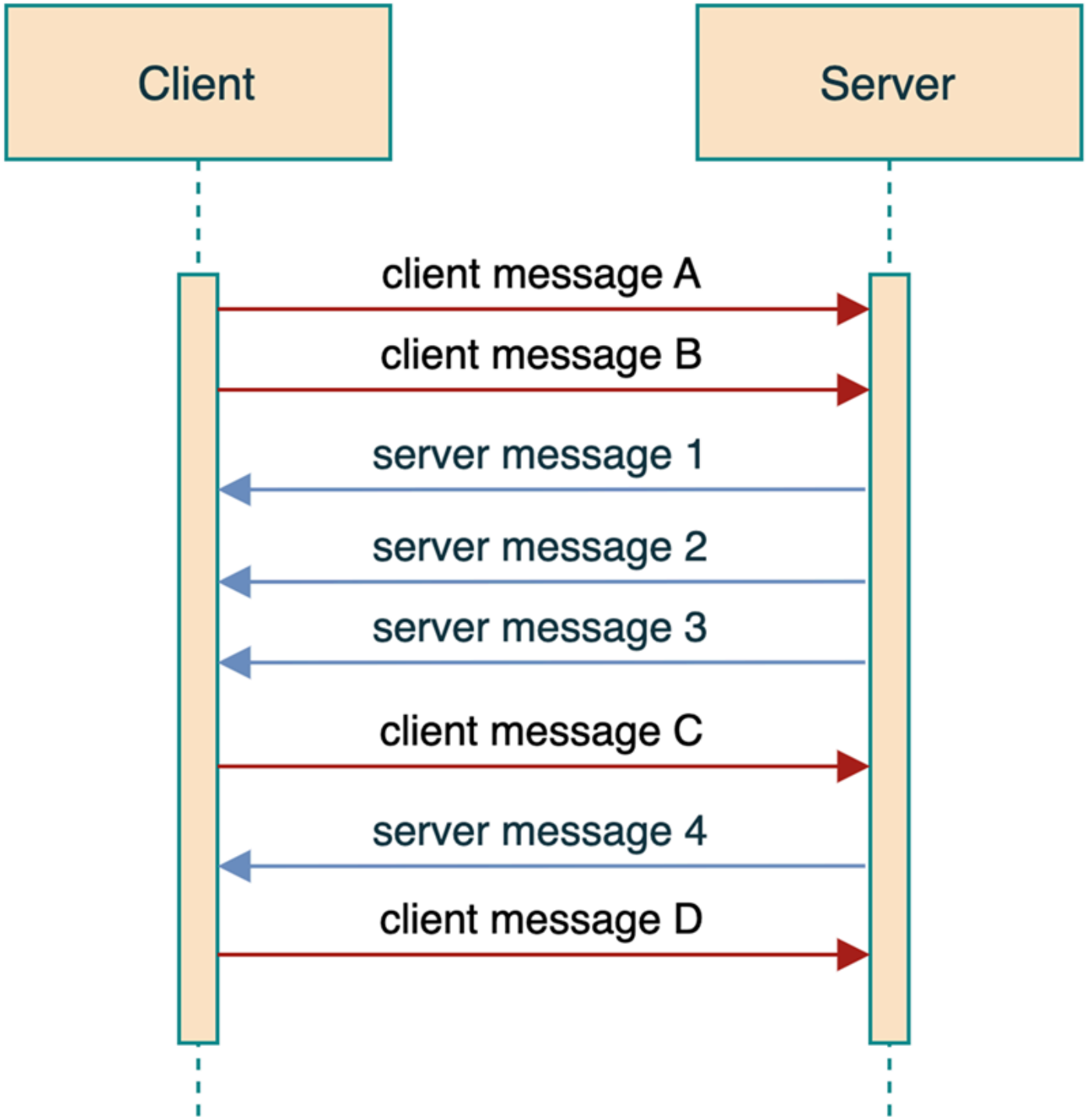


实施：

- 为失败的回调实施重试机制。
- 像保护其他服务一样保护回调位置。
- 处理回调超时。

双向通信

要实施双向通信，必须在客户端与服务之间建立有状态的连接，这样客户端与服务都可以发送和处理消息。此过程如下图所示。尽管通信是异步的，但服务必须能够支持每个客户端的开放连接。



实施的注意事项：

- 消息排序
  - 序列号

  - 分区策略
  - 消息排序
- 状态管理
  - 事件溯源模式
  - 状态协调
  - 一致性模型
- 错误处理
  - 死信队列
  - 重试策略
  - 断路器
  - 回退策略
- 监控和可观测性
  - 相关性 IDs
  - 消息跟踪
  - 性能指标
  - 系统运行状况指标

协调选项

对于调用单个服务或少量服务的客户端，同步和异步通信都表现良好。然而在实际应用环境中，这种通信机制往往很快变得复杂且难以扩展。完成一项工作可能需要多个微服务，而这些微服务之间可能存在相互依赖关系。通常，这些交互被建模为工作流。设计这些工作流有两种方法：编排和编配。

主题

编排

在这种方法中，单个编排工具负责调用每个微服务，确定是按顺序还是并行发出调用，在整个过程中操作各个服务的响应，并编译最终结果。编排工具可以组合使用同步和异步调用。

AWS Step Functions 和 Amazon Managed Workflows for Apache Airflow（Amazon MWAA）非常适合工作流编排工具。

当流程中存在逻辑分支，且需要在一个位置封装该逻辑时，编排机制是不错的选择。当您需要实施异步声明检查模式时，此方法同样非常有用。例如，Step Functions 中的标准工作流可以暂停工作流并等待来自其他服务的回调。使用编排工具还可以改善流程的监控和可观测性。

示例：Step Functions

您可以使用 Step Functions 来协调多个 Lambda 函数和其他函数 AWS 服务，为微服务集成构建复杂的工作流程。此选项对于涉及多个微服务、运行时间较长且包含多个步骤的流程尤为有用。

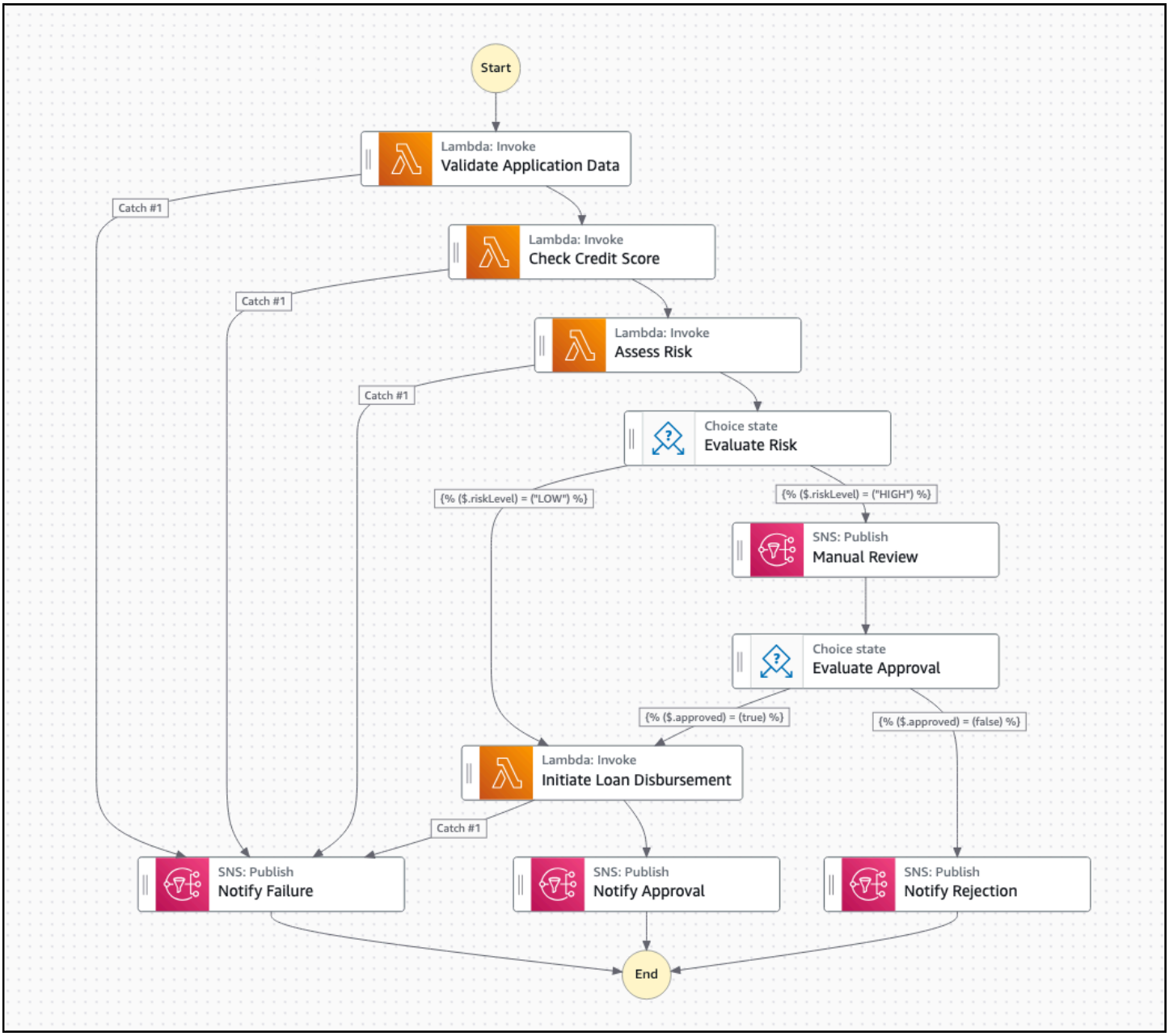
若为以下情况，应考虑使用 Step Functions：

- 微服务集成涉及复杂的多步骤流程。
- 您需要在长期运行的操作中保持状态。
- 您想在工作流层级实施错误处理和重试逻辑。
- 你需要同时协调同步和异步操作。

Step Functions 提供可视化编辑器用于设计复杂的工作流，从而简化状态机的创建与管理流程。它提供内置的错误处理机制，包括重试逻辑和错误状态管理，从而增强了应用程序的可靠性和稳健性。标准工

作流支持最长达一年的长期运行流程，适用于跨越较长时间段的工作流。此选项将编排逻辑和应用程序代码分开，因此可以显著降低代码的复杂性。这意味着开发人员只需专注于核心业务逻辑，而 Step Functions 则负责分布式组件的流量控制与协调。

例如，考虑金融服务应用程序中的贷款审批流程，如下图所示。此流程从提交贷款申请时开始。



在上图所示的状态机中，Step Functions 编排了以下步骤：

- 验证应用程序数据（Lambda 函数）
- 检查信用评分（调用外部 API 的 Lambda 函数）

- 评测风险（Lambda 函数）
- 如果风险很高，转至手动审查（人工审批任务）
- 如果获得批准，启动贷款发放（Lambda 函数）
- 向申请人发送通知（Amazon SNS）

可以采用这种方法来可靠地管理复杂且可能长期运行的流程，内置错误处理功能，并能同时包含自动化和手动步骤。

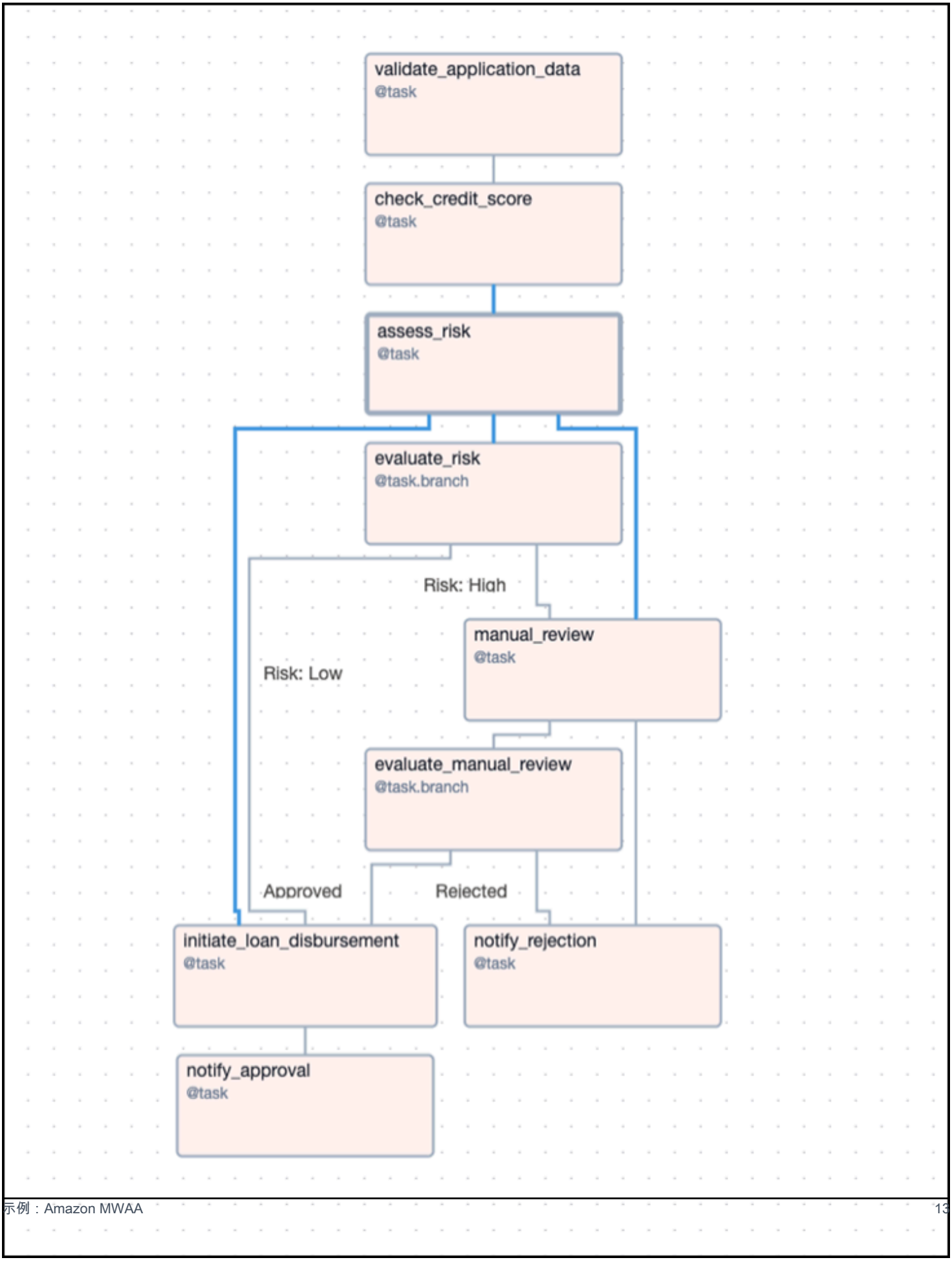
注意事项：

- 请精心设计状态机，以处理所有可能的状况。
- 尽可能并行执行步骤。
- 使用 Step Functions 中内置的错误处理与重试机制处理永久和临时故障。
- 根据您的使用案例，考虑使用标准或快速工作流。对于短周期或高吞吐量的流程，快速工作流可能是更优的选择。
- 监控执行指标以优化工作流。
- 使用嵌套的工作流在多个状态机之间封装和重复使用功能。
- 对于复杂的工作流，可以考虑使用 Amazon Bedrock 代理作为 Step Functions 的替代方案。

有关更多信息，请参阅 Step Functions 文档。

示例：Amazon MWAA

如果您的组织已经在使用 Apache Airflow，Amazon MWAA 将自然而然地用作工作流编排工具。在 Apache Airflow 中，你可以使用 Python 将工作流程构建为有向无环图 DAGs ()。Step Functions 部分所示状态机的 DAG 表示形式可能如下所示：


validate_application_data
@task
check_credit_score
@task
assess_risk
@task
evaluate_risk
@task.branch
Risk: High
Risk: Low
manual_review
@task
evaluate_manual_review
@task.branch
Approved
Rejected
initiate_loan_disbursement
@task
notify_rejection
@task
notify_approval
@task

有关使用的信息 DAGs，请参阅 Amazon MWAA 文档。

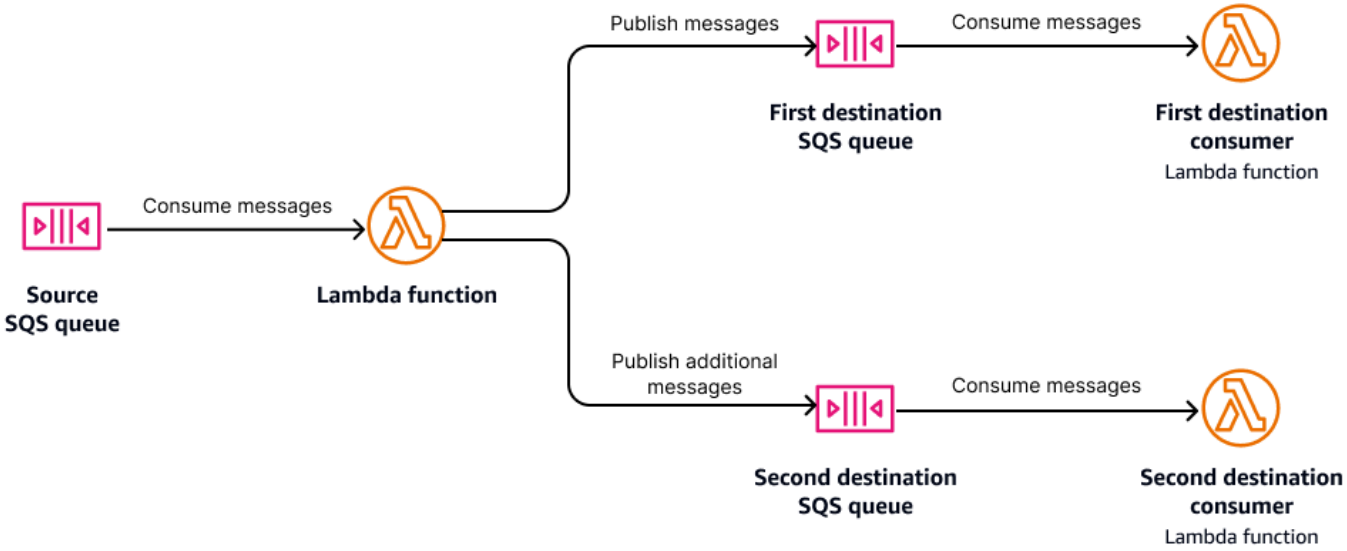
Step Functions 与 Amazon MWAA 的主要区别

- Step Functions 是一项完全托管式无服务器服务，无需预置基础设施，也无需安排维护时段。必须提前部署 Amazon MWAA，然后选择集群中的节点大小和数量。
- 在 Step Functions 中，您可以通过多种方式编写状态机，包括使用 Workflow Studio、直接以 JSON 格式编写，或者使用 AWS Cloud Development Kit (AWS CDK)。Apache A DAGs irflow 是用 Python 编写的。
- 使用 Step Functions 时，若无工作流在运行，不会产生任何费用。使用 Amazon MWAA，即使没有 DAGs 运行，您也会产生费用。

编配

在精心编配的系统中，各个组件接收任务，完成部分工作，并可能发出任务以完成后续工作。没有中央编排机制。编配机制使得服务能够轻松独立扩展，因为每个服务都相对独立地运行。当收到工作请求时，它便开始处理，无论服务的最大吞吐量为何。编配通常是事件驱动型架构（EDA）的核心部分。

下图中，Lambda 函数之间不存在协调关系。每个函数仅处理已订阅队列中的消息。每个函数负责自身的错误处理，并能控制并发性，例如当下游依赖项存在每秒请求数（RPS）限制时。



EDA 提供众多好处，例如服务的松耦合以及可扩展性。本指南未对 EDA 原则做详细介绍。有关更多信息，请参阅:

- AWS Well-Architected Framework — 无服务器应用程序视角

- 事件驱动型架构简介（无服务器平台）
- 过渡到事件驱动型架构（无服务器开发人员指南）

选择您的协调方法

在集成微服务时，编配和编排各有其用途。在单个微服务的边界内选择编配，这样就可以完全掌控依赖关系。跨微服务边界工作时，请选择编排。例如，参与分布式事务的多个微服务将受益于编排机制，以应对故障导致的回滚。处理其他微服务可能感兴趣的事件的微服务，将受益于编排和事件驱动型架构。

当单个事务涉及多个系统时，实现回滚的常见模式是 Saga 模式。

管理 APIs

适当的 API 管理使内部和外部消费者都能访问您的微服务。 AWS 提供了多种服务，您可以将这些服务一起使用来安全地公开您的微服务 APIs。这些服务使您能够加强安全性， APIs 并从一个中心位置实施监控和可观察性。如果您的用户 CloudFront与托管服务的地理位置相距甚远， APIs 您也可以使用 Amazon 来提高性能。 AWS 区域

Amazon API Gateway

Amazon API Gateway 是一项完全托管的服务，使开发人员能够以任何规模创建、发布、维护、监控和 WebSocket APIs 保护 REST。您可以使用 API Gateway 来实施本指南通信模式部分所述的多种模式。

REST 有两种主要类型 APIs：REST 和 HTTP。两种类型都支持 RESTful APIs，但提供的功能不同。要确定哪个最适合您的需求，请参阅 API Gateway 文档 APIs中的在 REST APIs 和 HTTP 之间进行选择。本指南的本部分重点介绍 API Gateway REST APIs。

使用 API Gateway 作为您的入口点 APIs 提供了一个实现常见问题（例如请求验证和安全）的地方。API Gateway REST APIs 提供请求验证，允许您使用 JSON 架构定义请求的格式。API Gateway 会根据您定义的架构验证传入的请求，并拒绝格式错误的请求。

身份验证和授权

API Gateway REST APIs 支持以下身份验证 (authN) 和授权 (authZ) 机制：

- IAM – 如果您使用 IAM，必须使用AWS 签名版本 4（SigV4）对 API 请求签名。
- Amazon Cognito – API Gateway 将验证所提供的持有者令牌是否由 Amazon Cognito 用户池签发。如果已经在使用第三方身份提供者（IdP），也可以将 Amazon Cognito 用户池配置为与之集成。您也可以使用 Amazon Cognito 用户池进行 machine-to-machine (M2M) 身份验证。
- AWS Lambda 授权方 — API Gateway 将调用您指定的 Lambda 函数来执行您想要的任何检查，以确定是否应授权请求。

有关更多信息，请参阅 API Gateway 文档 APIs中的控制和管理 REST 访问权限。

API 密钥和速率限制

您可以使用 API 密钥 APIs 和使用计划来控制允许谁给您打电话以及以什么费率给您打电话。API 密钥不应用于身份验证，但可与前文提到的架构结合使用。用户并不总是需要提供自己的 API 密钥，例

如，Lambda 授权方可以为用户返回 API 密钥。您可以在使用计划中指定吞吐量、突发限制和每月配额。有关更多信息，请参阅 API Gateway 文档 APIs中的 REST 使用计划和 API 密钥。

公共和私人 APIs

可通过互联网访问 APIs 的 API Gateway REST 支持两种终端节点类型：

- 边缘优化，这意味着呼叫者的请求会被路由到附近的接入 CloudFront 点 (POP)。这可以提升地理位置分散的客户端的性能表现。
- 区域性端点，这意味着请求会路由到特定 AWS 区域内的资源。当您的所有客户端都位于 API 部署区域附近时，这是不错的选择。

API Gateway REST APIs 还支持私有 API 终端节点，这些终端节点可通过接口 VPC 终端节点从虚拟私有云 (VPC) 进行访问。您还可以 APIs 通过在其他甚至其他接口中创建接口 VPC 终端节点来安全地共享私有 REST AWS 账户。 VPCs有关更多信息，请参阅 API Gateway 文档 APIs中的 REST 的 API 端点类型。

何时使用 API Gateway

API Gateway 是 RESTful 网络服务和实时 WebSocket连接的不错选择。 WebSocket APIs 在 API Gateway 中使用时，可以为连接和断开连接事件添加行为，例如将连接存储 IDs 在与客户端属性关联的外部数据存储中。还可以使用消息属性将请求路由到自定义行为。

REST 和 REST WebSocket APIs 都可以直接与许多计算资源集成， AWS 服务 而无需单独的计算资源，例如 Lambda 函数。这可以提高性能并降低成本。

REST 既 APIs 支持基于路径的路由，也支持基于标头的路由，你可以单独使用它们，也可以一起使用。一种常见的模式是提供 REST API 作为许多人的前门 APIs，实现前面讨论的共同关注点，然后像反向代理一样行事，并将授权请求路由到正确的 API 端点。

消息收发

如通信模式部分所述，您可以通过消息传送在服务之间实现同步或异步通信。AWS 提供多种无服务器服务，您应根据自身集成需求选择合适的服务。例如，如果需要按顺序传送消息，应选择 Amazon Simple Queue Service（Amazon SQS）或 Amazon Simple Notification Service（Amazon SNS）。这两项服务均支持先入先出（FIFO）传送模式，而 Amazon EventBridge 则不支持。

以下各节将更详细讨论这些服务。

主题

- Amazon SQS
- Amazon SNS
- Amazon EventBridge
- AWS AppSync Events 和 API Gateway

Amazon SQS

Amazon SQS 支持标准队列（不保证排序）和 FIFO 队列（可保证在给定消息组内排序）。

队列是编配微服务的常用方法，可为消息提供长达 14 天的持久存储空间。队列由生产者填充，由使用者清空。将 AWS Lambda 用作使用者时，可将 SQS 队列配置为事件源。这种情况下，Lambda 服务事件源映射（ESM）会为您轮询队列，并在消息可用时将其传送至您的 Lambda 函数。在 Amazon Elastic Container Service（Amazon ECS）、Amazon Elastic Compute Cloud（Amazon EC2）等其他类型的计算服务上运行的微服务必须实施自己的轮询机制，以便从队列提取可用的新消息。

适用于 Amazon SQS 的 Lambda ESM 还支持消息筛选，因此您可以根据消息正文的内容，仅处理队列中部分消息。

轮询

Amazon SQS 支持对消息进行短轮询和长轮询。短轮询会查询部分服务器以查找可用消息，并立即返回结果。但它可能不会返回所有可用消息。当您的应用程序需要尽快使用消息，或无法忍受较长时间的等待时，这很有用。

长轮询在经过可配置的时间间隔或接收可配置数量的消息后，才返回消息。这样可减少空轮询的数量，即没有消息返回的轮询次数，尤其对于接收消息较少的队列而言。减少空轮询的数量可降低 Amazon SQS 成本，因为此服务按每次请求收费，而每次轮询操作都算作一次请求。

指南

下列情况下，队列是一个不错的选择：

- 您希望解耦组件，且组件之间无需同步通信。
- 您正在具有不同可用性服务水平协议（SLA）或服务水平目标（SLO）的组件之间进行通信。
- 一组消息通常只有一个使用者。

下列情况下，可以考虑另外一种选择：

- 您需要同步通信。
- 您需要复杂的路由逻辑才能将消息发送给正确的使用者。

Amazon SNS

Amazon SNS 允许您创建标准主题和 FIFO 主题。主题用于实施发布/订阅（pub/sub）架构。Amazon SNS 支持多种订阅类型，包括电子邮件、短信（假设您已配置发起身份，例如免费电话号码或 10 位长代码）、HTTP（S）端点和 SQS 队列。最终用户对 SNS 主题的订阅（例如电子邮件和短信）必须由订阅用户确认。Amazon SNS 可助力服务广泛分发消息，这意味着单条消息可发送给潜在的大量订阅用户。SNS 标准主题的默认订阅限制为 1250 万。

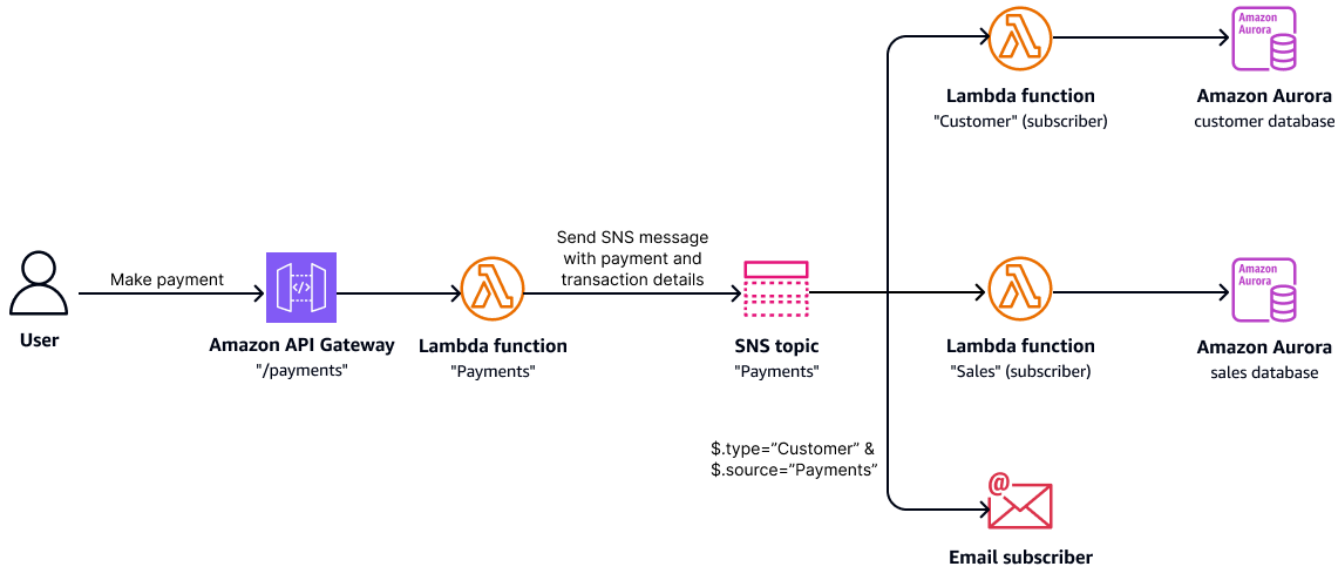
在微服务环境中，SNS 主题有助于将消息路由和传送逻辑与发布者解耦。这可以通过使用主题筛选器来实现。从概念上讲，主题筛选器与 Amazon EventBridge 规则有些相似，但它们是为每个订阅用户配置的，并非从一个集中位置提供。例如，假设您有：

- 订单服务，用于处理订单。
- 配送服务，用于处理订单配送。
- 忠诚度服务，为会员的订单奖励忠诚度积分。

当订单准备好发货时，会向主题发布一则消息。配送服务订阅了该主题，但没有应用筛选器，因其想了解所有订单。想象一下，您有一项忠诚度服务，负责在会员下单时向他们奖励积分。但是，并非所有订单都是会员下的。忠诚度服务将订阅该主题，但会实施订阅筛选器，以检查指示是会员订单还是访客订单的属性。

考虑一下系统收到最终用户发起付款请求时的情况，如下图所示。这种情况下，多个下游系统需要知道请求已发出，以便采取相应措施。当您使用 Amazon SNS 时，付款将发布到某个 SNS 主题，Lambda

函数会订阅该主题以更新客户和销售数据库。此外，电子邮件订阅功能（须客户确认）会通过订阅筛选器向客户发送确认邮件。



指南

此部分所述的 Amazon SNS 部分功能与事件总线（如 EventBridge）提供的功能存在重叠。考虑在以下情况下使用 Amazon SNS：

- 一个主题将拥有大量订阅用户。
- 您想使用 EventBridge 本身不支持的订阅类型（例如电子邮件或短信）。
- 订阅用户应该能够确定他们的订阅筛选器。
- 您需要按顺序发送消息给订阅用户（按消息组）。

若有许多主题，且需通过订阅和筛选器在微服务间路由消息，EventBridge 通常是更优选择。

Amazon EventBridge

Amazon EventBridge 是无服务器事件总线服务，通常是事件驱动型架构（EDA）的基础。您还可以用其在微服务之间异步路由和传送消息。生产者使用 EventBridge 将事件发布到总线。您可以配置基于事件内容的匹配规则，并选择一个或多个目标，用于接收符合该规则的事件。EventBridge 支持多种规则目标。事件总线可助力将生产者与使用者分开，并整合路由和传送逻辑。

在 EventBridge 中，您还可以创建定时规则，以便在特定时间执行操作。可以使用基于 cron 和基于速率的表达式来定义事件。

借助 EventBridge Pipes，您无需使用计算服务（如 AWS Lambda）即可将消息从源传输到目标。例如，假设您有一个接收消息的 SQS 队列，该队列应触发 AWS Step Functions 状态机。与其创建一个具有事件源映射的 Lambda 函数来从队列中使用消息，并使用 AWS SDK 编写代码来调用状态机，不如直接使用 EventBridge 管道来完成这些操作，无需编写任何自定义代码。

EventBridge 通常与其他消息传送服务一起使用，例如 Amazon SQS 和 Amazon SNS。例如，将事件传送到 SQS 队列时，接收服务能够灵活地在自身具备处理能力时使用消息，使用速率与事件生成速率无关。同样，您也可以将应向大量订阅用户分发的事件发布到 SNS 主题。

指南

请在下列情况下使用 EventBridge：

- 不需要在服务之间进行同步通信。
- 想要将消息路由逻辑与微服务解耦。微服务仅负责生成事件并将其发布到事件总线，感兴趣的服务会创建规则来匹配并分发这些事件。
- 您需要将消息从一项支持的服务传送到另一项支持的服务。

下列情况下，可以考虑其他服务：

- 需要对事件进行严格的排序。在这些情况下，考虑 Amazon SQS FIFO 队列或 Amazon SNS FIFO 主题。或者，考虑 Amazon Kinesis Data Streams、Amazon Managed Streaming for Apache Kafka（Amazon MSK）等事件流服务。

AWS AppSync Events 和 API Gateway

AWS AppSync Events 和 Amazon API Gateway 都可为您的微服务提供托管的 WebSocket 体验。

AWS AppSync Events 通过使用 WebSocket 提供简化的实时消息传送体验。AWS AppSyncEvents 支持单播和组播消息传送，支持将通道灵活分组到命名空间，并支持通配符。通过使用 AWS AppSync Events，微服务之间可以通过多种方式通信。例如，接收实时数据的服务可以将数据转换并发布到相应的通道，订阅用户将在该通道上实时接收数据。

API Gateway 还支持 WebSocket API。您可以定义与 AWS 服务（例如 AWS Lambda 和 Amazon DynamoDB）的集成，并配置映射到这些集成的路由选择表达式。API Gateway 提供专用路由，可用

于授权和管理 WebSocket 连接。您可以视需要将 WebSocket 连接信息存储在 DynamoDB 等数据存储库中。利用此信息，可通过 REST API 基于特定连接 ID，向特定 WebSocket 连接发布消息。

指南

请在下列情况下使用 AWS AppSync Events：

- 您有多个按命名空间分组的消息传送通道，并且想要通过使用通配符来发布和订阅通道组。
- 您的通信主要发生在不同系统之间，而非 AWS 服务之间。

请在下列情况下使用 API Gateway WebSocket API：

- 您想要客户端与 AWS 服务集成建立实时持久连接。
- 您想要自行管理 WebSocket 连接。例如，您可能想要允许其他系统在查询连接 ID 后向特定客户端发送消息。
- 您想要使用 API Gateway 功能，例如分阶段部署或代理集成，或者想要配置自己的子协议。

常见问题解答

如何组合使用不同的集成模式？

大多数情况下，您需要组合使用多种集成模式。例如，您可以使用 AWS Step Functions 通过声明检查模式，编排调用远程服务的流程。或者，您可能有一个精心设计的流程，先将消息放入队列，进而触发预先编排的服务。

使用微服务架构的主要好处是什么？

主要优势包括独立扩缩服务、改进故障隔离、通过团队并行工作提高开发速度，以及持续交付和部署（CI/CD）的能力。

如何在这些模式中实施错误处理？

您可以使用 AWS 服务中的内置机制实施错误处理。例如，可以使用重试逻辑配置 AWS Lambda 函数，而 Amazon SQS 支持死信队列来处理持续故障。此外，Step Functions 还提供工作流级别的错误处理和重试机制。

在异步通信中使用声明检查模式有哪些优势？

采用声明检查模式时，客户提交请求时会收到一个标识符。此标识符可以稍后用于检查状态和检索结果。这种模式提供了一种无需同步等待即可轮询结果的机制，可让客户受益。有关更多信息，请参阅本指南中之前的声明检查部分。

回调模式如何改善微服务中的异步通信？

回调模式允许客户端提供一个位置，供服务在处理完成后联系，从而改善了异步通信。这样，客户端无需等待响应，即可继续执行其他任务。有关更多信息，请参阅本指南中之前的回调部分。

我能否使用所描述的模式在微服务中实施双向通信？

您可以通过在客户端与服务之间创建有状态的连接来实施双向通信，这样它们就可以异步发送和处理消息。这要求服务为每个客户端提供一个开放连接。有关更多信息，请参阅本指南中之前的双向通信部分。

如何优化 Lambda 函数在异步通信模式中的使用？

您可以通过以下方式优化 Lambda 函数：确保 Lambda 函数具有幂等性来处理潜在的消息重复问题；使用 Amazon SQS 功能（例如消息组）进行排序；实施长轮询以降低成本。此外，您可以监控执行指标以确定优化机会。

使用 Amazon SNS 和 EventBridge 使用该 pub/sub 模式有什么主要区别？

Amazon SNS 会向所有订阅用户发送一则消息，其中可能包含部分订阅者不需要的数据。Amazon EventBridge 允许您拥有与单个事件匹配的多个规则，每个规则触发不同的下游服务或操作，从而实现更精细的控制。有关更多信息，请参阅本指南前EventBridge面的 Amazon SNS 和章节。

资源

AWS 服务 文档

- Amazon API Gateway
- AWS AppSync事件
- Amazon EventBridge ()
- Amazon MWAA
- Amazon SNS
- Amazon SQS
- AWS Step Functions

补充阅读

- 在 AWS Cloud中实现应用程序现代化的策略
- 在 AWS Cloud中实现应用程序现代化的分阶段方法
- 评估 AWS Cloud中应用程序现代化的准备情况
- 将单体分解为微服务
- 使用 AWS 消息收发服务实施企业集成模式：点对点渠道
- 发布/订阅消息：异步事件通知

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
主要更新	扩展、重组并更新了指南，以反映 AWS 服务更新。	2025 年 9 月 10 日
初次发布	—	2021 年 1 月 11 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：

- Refactor/re-architect — 充分利用云原生功能来提高敏捷性、性能和可扩展性，从而移动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 数据库迁移到亚马逊 Aurora PostgreSQL-Compatible 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 Cloud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：将您的客户关系管理 (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，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 Cloud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

A2A () Agent-to-Agent

一种支持任务委托和状态转移的代理到代理协作的状态协议。

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。

座席

一种能够使用工具自主推理、计划和采取行动来实现目标的人工智能系统。

特工行动

在生产环境中大规模构建、测试、部署和运行 AI 代理的操作实践。

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 (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。

blue/green 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，请参阅指南中的 “实施破碎玻璃程序” AWS Well-Architected 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。

公民开发者

使用无code/low代码平台创建 AI 应用程序但没有专业技术技能的企业用户。

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 Cloud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，创建登录区、定义 CCoE、建立运营模型）
- 迁移 - 迁移单个应用程序
- Re-invention — 优化产品和服务，在云端进行创新

Stephen Orban在 AWS Cloud 企业战略博客的博客文章《走向之旅 Cloud-First 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）

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《 AWS Well-Architected 框架》中安全支柱的组成部分。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

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。

深度防御

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，深度防御方法可能将多因素身份验证、网络分段和加密结合起来。

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《工作负载灾难恢复 AWS： AWS Well-Architected 框架中的云端恢复》。

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。埃里克·埃文斯（Eric Evans）在他的《Domain-Driven 设计：解决软件核心的复杂性》（波士顿： Addison-Wesley 专业版，2003年）一书中介绍了这个概念。有关如何使用带有 strangler fig 模式的域驱动设计的信息，请参阅使用容器和 Amazon API Gateway 逐步实现传统微软 ASP.NET (ASMX) 网络服务的现代化。

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。 Big-endian 系统首先存储最重要的字节。 Little-endian 系统首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 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

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。这种技术是情境学习的应用，模型可以从提示中嵌入的示例（镜头）中学习。 Few-shot 对于需要特定格式、推理或领域知识的任务，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，它已使用海量的通用和未标注数据集进行训练。FM 能够执行各种常规任务，例如理解语言、生成文本和图像以及使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。

FM 网关

一种集中式中介，用于控制和规范对基础模型的访问。也称为 LLM 网关。

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

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。

护栏 (AI)

用于过滤、验证和限制代理输入和输出的安全机制，有助于确保负责任和安全的 AI 行为。

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。

人机在圈 (HitL)

一种工作流程模式，其中代理执行在关键决策点暂停以供人工审查和批准。

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 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。

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，请参阅框架中的使用不可变基础架构部署最佳实践。 AWS Well-Architected

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）在2016年推出，指的是通过连接性、实时数据、自动化、分析和的进步实现制造流程的现代化。 AI/ML

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 LLM。

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。

MCP

参见模型上下文协议。

模型上下文协议（MCP）

一种用于代理与工具通信的无状态协议。

MCP 服务器

一种通过模型上下文协议公开一个或多个工具的服务。

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 AWS Well-Architected 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publish/subscribe模式的轻量级机器对机器 (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

Cross-functional 通过自动化、敏捷的方法简化工作负载迁移的团队。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 Cloud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 Cloud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 Cloud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 Cloud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，该 AWS Well-Architected 框架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机器对机器 (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 框架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

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

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，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，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 Cloud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 Cloud 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 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。

暗影人工智能

在组织内受管控渠道之外构建或使用的未经授权的 AI 应用程序。

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 模式

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 Cloud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 逐步实现传统微软 ASP.NET (ASMX) 网络服务的现代化。

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

Key-value 对充当用于组织 AWS 资源的元数据。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。

工具

代理可以调用以在外部系统中执行操作的函数或 API。

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，该服务可以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。

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。