在上操作代理 AI AWS

AWS 规范性指导

AWS 规范性指导: 在上操作代理 AI AWS

Table of Contents

在 AI 上运行代理 AI AWS

Aaron Sempf、Brad Ryan、Bhargs Srivathsan 和 Akhil Bhaskar，Amazon Web Services

2025 年 8 月（文档历史记录）

Agentic AI 不是一项功能，而是一种新的运营模式。投资于纪律严明的架构、信任框架和与业务一致的部署模式的组织将引领下一代自适应的智能企业。

A@@ gentic AI 代表了自主软件代理和生成式 AI 的融合。它将代理的决策和目标导向行为与大型语言模型的语言理解和生成能力融为一体（ ）。LLMs这些代理可以在动态的企业环境中进行推理、行动、适应和协作。为了发挥这种潜力，企业必须将思维方式从模型部署转向代理基础架构。

本指南提供了一种组织策略，可将代理人工智能从孤立的实验转变为企业级的创造价值的基础架构。它可以帮助您在工作流程中嵌入具有治理、可扩展性和业务协调性的智能代理。

主要重点领域和建议

本指南重点介绍操作代理人工智能时的以下基础领域。为每个重点领域提供了组织和业务建议：

- 重点领域 1：明确代理意图和范围— 使代理与业务优先事项和认知瓶颈保持一致。将代理视为数字队友，而不仅仅是工具。
- 重点领域 2：为可组合性和协作性而设计— 采用具有模块化架构、语义协议和通过仲裁代理进行动态委派的多代理系统。
- 重点领域 3：多租户和控制架构师— 利用共享代理服务、集中式治理和基于角色的访问权限构建可扩展的租户感知基础架构。
- 重点领域 4：通过身份、护栏和可观察性建立信任— 强制执行可追溯性、运行时间控制和可解释性，以赢得利益相关者的信任。
- 重点领域 5：管理生命周期— 建立持续集成和持续部署 (CI/CD) 管道、即时版本控制、遥测和持续再训练，以支持代理的 AI 性能和效率。
- 重点领域6：使代理模式与商业模式保持一致— 通过基于使用量的模型、内部投资回报率指标和商业产品将代理能力货币化。

您可以使用本指南中的建议，让您的企业为大规模代理人工智能做好准备。它概述了组织必须如何围绕代理人工智能进行重组，包括 DevOps 为代理 (AgentOps) 团队构建、可互操作的系统以及扩大采用率的变更管理策略。它强调决策至上的思维以及与Well-Ar AWS chitected框架的一致性。

目标受众

本指南适用于企业架构师、 AI/ML 工程主管和数字化转型策略师，他们负责设计和扩展代理系统，将人工智能嵌入核心业务工作流程，以及在生产环境中操作 LLMs 和自主代理。要理解本指南中的概念和建议，您应该熟悉现代云原生架构和分布式系统、大型语言模型、基础模型功能以及 AI 治理的原则和平台工程。 DevOps

目标

通过实施本指南中的建议，您的组织可以实现以下业务成果：

- 通过自主的、以目标为导向的代理加速决策和工作流程执行，从而减少人为瓶颈和认知负担。
- 通过可重复使用的多租户代理平台，跨业务部门部署可扩展、经济高效的智能功能。
- 增强人工智能系统的弹性、信任度和治理，从而能够自信地在受监管、关键任务或面向客户的环境中采用。

关于此内容系列

本指南是关于代理人工智能的系列文章的一部分。 AWS要了解更多信息并查看本系列中的其他指南，请参阅 AWS 规范性指导网站上的 Agentic AI。

代理人工智能的战略基础

代理系统并不是什么新鲜事物。软件代理，包括机器人流程自动化 (RPA) 和决策引擎，已经存在了数十年。但是它们既简单又具有确定性，旨在遵循预定义的规则和符号逻辑来执行重复的、低变化的任务。随着生成式人工智能的兴起，游戏规则发生了变化。大型语言模型 (LLMs) 现在可以解释复杂的输入，动态生成响应，并快速合成知识。现在，你可以在没有脆弱或硬编码逻辑的情况下扩大代理规模。现在，代理可以推理、做出决策、调用工具、适应上下文，以及跨工作流程与其他代理进行协调。他们可以自主地朝着目标运作，保持记忆力，反思结果。

但是，仅有原始能力是不够的。没有整合的智能会产生新颖性，而不是影响力。为了从强大的力量中释放价值 LLMs，企业必须从孤立的实验转向工程生态系统。代理必须被视为生产级服务，在与任何企业系统相同的纪律下运行。这包括治理、可观察性、安全身份模型和生命周期管理。它们还必须带来真正的业务成果，而不是投机潜力。这些系统的架构应具有明确的决策和容错界限。整合自动恢复机制、实时性能监控和可扩展的资源管理非常重要。这可以帮助您处理代理交互的动态、非确定性，同时在整个企业工作流程中保持一致的服务级别。

在基础层面上，企业必须重新思考如何将智能嵌入运营结构。代理的设计必须能够与核心系统集成，遵守企业政策，并提供可衡量的价值。他们需要跨部门、跨领域和用户环境进行大规模运营。运营代理 AI 最终与使用有关；这是部署执行孤立任务的 AI 和部署可演变业务模式的代理之间的区别。

Agentic AI 代表了一种新的运营理念，它要求我们从根本上改变对待系统、流程和人员的方式，以在整个组织中扩展情报。代理成为增强人类能力的战略资产。通过将代理人工智能集成到运营中，组织可以解锁可推动业务价值、增强人类能力和优化复杂工作流程的见解。

代理人工智能的战略重点领域

要从早期的原型转变为生产级和价值创造系统，团队需要一种将架构、流程和产品思维融为一体的连贯策略。

许多组织仍然以工具优先或以模型为中心的思维方式来对待人工智能。生成式人工智能扩大了实验范围，但通常与业务战略或可衡量的结果没有明确的一致性。如果没有明确的战略角色，代理人就有可能成为耗尽资源而不是提供可扩展价值的新实验。要确立代理人工智能的战略作用，组织必须从业务优先事项入手。确定认知过载、决策瓶颈或工作流程分散的领域，在这些领域，自主性可以缓解压力。使用特定领域的问题陈述来确定代理责任。将代理视为能够推理、委派和适应的数字队友，而不是工具。

决策科学是一门将数据科学、分析和行为建模相结合以改善决策的学科。应在代理架构流程的早期将其集成，以使设计与业务结果保持一致。通过识别决策模式、模拟权衡和量化价值影响，决策科学可以帮助您确定代理自主权可以在哪些方面提供最高价值。决策科学可以加快决策、减少错误并实现实时适应。这种以数据为依据的基础为代理设计奠定了可衡量的见解，并且可以与规则引擎、分析平台和预测模型等现有企业技术更紧密地集成。

为了帮助确定代理的战略角色，本节介绍了构成代理人工智能运营支柱的基本重点领域。从负责构思和设计代理的技术负责人、架构师或产品负责人的角度来看，每项工作都映射出要完成的核心工作。这些重点领域不是连续步骤。在整个系统生命周期中，每一个都值得重新审视，以培养弹性、可扩展和可盈利的代理生态系统。

本节包含以下重点领域：

- 重点领域 1：明确代理意图和范围
- 重点领域 2：为可组合性和协作性而设计
- 重点领域 3：多租户和控制架构师
- 重点领域 4：通过身份、护栏和可观察性建立信任
- 重点领域 5：管理生命周期
- 重点领域6：使代理模式与商业模式保持一致

重点领域 1：明确代理意图和范围

Job to done：“帮我确保每个代理都能以明确的界限解决一个真正的问题，而不仅仅是一个很酷的演示。”

Agentic AI 不仅仅是能力建设。这是关于以正确的方式解决正确的问题，以获得正确的结果。首先要完全清楚代理人工智能解决方案的意图。

Strategy

很多时候，组织从模型可以做什么（例如打电话 APIs、回答问题或生成摘要）开始，然后围绕它改造用例。这会导致范围蔓延、集成不佳以及代理在技术上令人印象深刻但操作上毫无用处。取而代之的是，首先通过以下具体问题来定义代理的角色：

- 代理人对什么具体结果负责？
- 它代表谁行事？
- 谁受益？
- 代理的自主权在哪里开始和结束？
- 失败时会发生什么？

范围明确的代理人要有明确的工作、明确的职责和可衡量的成功标准。不要将代理视为助手或聊天机器人。相反，给它起一个职称。可以将其视为客户成功代理、产品退货处理员或合规监视器。

在吸引利益相关者或客户时，要强调代理人工智能系统的可扩展性和适应性。这些代理人随着业务的发展而发展，通过学习和反馈不断改进。为了减少阻力并加快采用率，请重点介绍代理工具的设计是如何在考虑员工同理心的情况下设计的。它们提供透明度、控制和可选的替代机制，以建立信任。代理人不是取代人员，而是增强了人的能力和决策能力，帮助员工了解最新情况，专注于高价值的任务。

成功实施的关键是使代理人工智能与具体、高影响力的业务成果保持一致。鼓励团队和合作伙伴从解决明显痛点的有针对性的试点项目开始。快速获胜可以产生可衡量的投资回报率 (ROI)，建立内部支持，并为更广泛的采用创造动力。

为了指导采用和成熟，组织可以按照演化模型来构思代理设计。代理自主权、复杂性和业务影响力逐渐增加。以下是该模型的各个阶段：

- 观察者代理从噪声中获得见解。一个例子是市场情绪代理人，它通过数字渠道跟踪品牌认知度。
- 助理代理支持人类决策。一个例子是交易咨询代理，它为销售团队综合竞争对手的数据和市场状况。
- 自主代理在定义的边界内独立行动。一个例子是资源分配代理，它可以根据需求动态调整云基础架构。
- Orchestrator 代理协调多代理工作流程。一个例子是供应链优化代理，它管理库存、物流和预测代理之间的交互。

- 创新者代理人创造了新的战略可能性。一个例子是商业模式创新机构，它分析市场趋势并推荐新的收入来源。

围绕这些战略成果和成熟度来构思代理可以提高注意力，加快采用速度，并建立利益相关者的信心。

为了支持这一重点领域的协调一致 AWS 服务，例如 Amazon Quick，可以直观显示与代理驱动结果相关的关键绩效指标 (KPIs)。您可以使用 Amazon CloudWatch 近乎实时地监控代理行为、性能指标和系统运行状况。使用操作反馈来调整代理互动和资源使用情况。 AWS CloudTrail可以在早期实验和完善阶段提供对代理活动和整合模式的可见性。

定义意图和范围的商业价值

代理人工智能的采用代表了组织实现数字化转型和卓越运营的方式的关键转变。这不仅仅是自动化。它旨在实现智能自主权，从而加快决策和价值实现。

主要业务驱动因素包括以下几点：

- 竞争优势 — 早期采用者通过更快的洞察、更好的服务和自适应运营获得战略优势。
- 增强客户体验 — 客服人员提供实时、个性化、全天候的支持，从而提高满意度和忠诚度。
- 运营效率 — Agentic AI 通过自动执行复杂、重复的决策任务，显著减轻了人类的认知负担。这使员工能够将注意力集中在价值更高的活动上，并可以降低成本。

各行各业的真实用例包括以下内容：

- 金融服务 — 人工智能代理可以提供个性化的财务建议并检测欺诈行为。
- 医疗保健 — 分诊和治疗计划药物可以提高临床通量。
- 零售 — 代理商可以充当智能购物助手或实时优化库存。
- 制造 — 代理商可以进行预测性维护或协调供应链。

重点领域 2：为可组合性和协作性而设计

Job to done：“让我像构建服务一样构建代理 — 模块化且可测试，这样它们就可以根据需要进行组合和编排。”

许多人工智能工作都是从单一的、以模型为中心的飞行员开始的。它们很有用，但很难跨领域扩展或适应复杂的问题。当这些药剂设计为互操作时，可以对化合物进行估值。在技术领域，可组合性是将模块化组件组合在一起以创建能够适应变化的灵活、可扩展的解决方案的行为。如果没有可组合性，智能就

会被锁定在特定的工作流程中。此外，代理协作还会带来编排、状态管理和协议协商的复杂性，而传统的自动化团队可能无法处理这些问题。

Strategy

拥抱多代理模式。像组织部门这样的模型代理：模块化、专业化和可互操作。定义清晰的接口、共享的上下文格式和标准通信协议，例如模型上下文协议 (MCP) 或 Agent2Agent (A2A)。采用多代理编排模式，例如 swarm、graph 或分层协调。根据任务结构和信任级别，这些模式可以帮助代理以并行、顺序或共识驱动的工作流程动态发现能力并相互请求服务。

要促进可扩展且受管控的协作，请使用仲裁代理。这种代理是一个中立的机构，可以根据已知的能力和后备策略促进任务分配。虽然不是集中式控制器，但仲裁代理在信任和合规性方面起着至关重要的作用。它确保敏感或受监管的任务仅路由给符合身份和策略要求的代理。它充当受策略约束的工作流程的看门人。它强制隔离并支持可解释的委托。至关重要的是，仲裁代理不是瓶颈；它与以横向方式运作的自我协调代理共存。 peer-to-peer这些代理直接委派子任务、共享上下文和解析依赖关系。

这种混合模型既支持确定性分配（通过仲裁代理），也支持紧急协作。它将结构与灵活性融为一体。在此架构中，代理可以分为以下特殊角色：

- 决策代理人，例如政策执行者、资源分配者和风险评估者
- 知识代理，例如上下文聚合器、模式识别器和异常检测器
- 执行代理，例如任务执行者、质量控制员和集成经理

为了有效协调，多代理系统必须支持用于状态管理、故障恢复和冲突解决的强大交互协议。即使代理人独立运作，这也促进了稳定性和问责制。

制定明确的扩展规则，例如基于负载的代理实例化、上下文感知资源分配以及自动能力发现和注册。这些措施有助于系统根据需求或复杂性动态增长。

将代理设计为分布式消息传递基板中的 ready-to-use模块。例如，您可以将 A mazon EventBridge 与 A2A 或 MCP 配合使用，而不是孤立的服务。采用版本控制、 CI/CD 管道和代理模板来支持系统稳定性，同时加快内部采用和生命周期演进。鼓励代码重用和标准化，以减少集成摩擦并促进具有弹性的生态系统。

协作是一种力量倍增器。它可以在多代理环境中解锁规模、专业化和弹性。为了支持这种动态协作，组织应设计一个用于代理协调的轻量级控制平面。该控制平面包括以下内容：

- 功能注册表，定义每个代理可以做什么，并支持版本化元数据以供同行发现
- 任务仲裁逻辑，使用仲裁员或主管代理根据上下文、可用性和策略来路由任务

- 生命周期和状态跟踪，可实现实时决策上下文和安全交接

控制平面可确保多代理系统保持可扩展性、策略一致性和容错性，而不会集中权限或减慢运营速度。

但是，多代理环境也带来了操作难题。维护座席交互的背景信息、管理共享状态和协调操作会增加复杂性和成本。如果您在代理间通信期间使用消耗代币 LLMs 的代币，则成本可能会增加。必须将这些成本与大规模智能自动驾驶的复合业务收益进行权衡。

为了应对这些挑战，可以考虑将关键问题抽象出来的代理平台，例如：

- 标准化的通信协议和语义格式
- 内置编排逻辑和动态路由
- 代理之间共享上下文和内存管理
- 故障期间的回退处理和优雅降级

对于采用多代理策略的团队，最好的方法是从小处着手，然后进行规模化设计。从解决实际问题的有针对性的单一代理解决方案开始。然后，逐步将这些代理组合成一个合作系统，在这个系统中，每个人都可以根据共同的目标和全系统的背景进行发现、协调和委派。

重要的是，强大的错误处理和优雅的降级必须是主要的设计原则。当代理不可用或出现故障时，多代理系统应能够继续部分工作流程或启动备份逻辑。这无需刚性耦合即可提高可靠性。

AWS 服务 提供强大的功能来大规模支持此架构。Amazon EventBridge 和 EventBridge P ipes 为多代理消息传递提供了结构化、事件驱动的支柱。为了管理模块化行为，AWS AppConfig允许在代理实例之间进行安全、动态的配置切换。要支持共享上下文和内存管理，请使用 Amazon Dynamo DB 实现轻量级、租户感知状态持久化以及跨代理的快速上下文检索。您可以使用亚马逊简单存储服务 (Amazon S3) S ervice 来存储结构化提示历史记录、共享项目或代理生成的输出。对于需要状态协调的更复杂的工作流程，AWS Step Functions可以使用检查点和错误恢复逻辑来协调长时间运行的流程。这些服务共同帮助您创建可组合、弹性且语义连接的多代理系统，这些系统可根据企业需求进行扩展。

多代理系统的商业价值

虽然许多组织从单代理解决方案开始其人工智能之旅，但代理人工智能的全部潜力是通过可扩展的多代理系统来释放的。这些系统是解决复杂的分布式问题和创建可随业务需求而演变的强大、灵活的人工智能生态系统的关键。

多代理系统的核心业务优势包括：

- 可扩展性 — 任务和工作负载可以分布在专门的代理上，以提高容量和性能。

- 灵活性-可以在最小的干扰下添加、替换或修改代理，从而在动态环境中实现灵活性。
- 弹性 — 借助冗余角色和智能故障转移，即使单个代理出现故障，也能保持系统稳定性。
- 专业化 — 专门构建的代理以更高的效率和精确度执行任务。
- 成本效益 — 可重复使用的代理组件可加快开发速度并降低新功能部署的成本。

虽然多代理系统需要更多的前期规划，但它们可以提供长期的敏捷性、速度和创新能力。投资于灵活的代理协作架构的企业有能力快速部署新的 AI 功能，适应不断变化的需求，并在日益由代理驱动的竞争格局中处于领先地位。

重点领域 3：多租户和控制架构师

Job to done：“帮助我在不失去控制、责任或知名度的前提下，在多个客户之间扩展代理使用率。”

早期的原型可以单独证明价值，但大多数企业需要同时支持多个客户、部门或工作流程。这意味着每个代理都必须在明确定义的政策、数据和身份界限内运作。如果没有多租户，运营就会变得脆弱且成本高昂，治理就会变得错综复杂。

Strategy

遵循软件即服务 (SaaS) 架构的原则。例如，为租户隔离、策略实施和资源控制而设计。使用租户感知内存、配置和身份架构代理和编排平台。要强制执行边界，请使用标记、基于角色的访问控制 (RBAC) 以及身份和访问管理范围界定。

采用统一的可观测性层，在该层中，代理遥测按租户上下文汇总。实施集中式策略引擎和基于配置的功能切换，以强制执行动态行为规则。

将代理部署作为服务构建。使内部团队或客户能够以可扩展、可管理的方式使用代理功能 APIs。AWS 为这些模式提供了坚实的基础。您可以使用 Amazon Cognito 管理用户和租户身份，AWS Organizations使用服务控制策略 (SCPs) 进行跨账户管理，AWS Resource Access Manager (AWS RAM) 用于安全共享功能。此外，AWS AppConfig还可以按租户或环境动态管理代理行为。这些服务有助于强制执行边界和政策，同时支持共享基础架构。

这种从静态部署到动态配置的过渡将代理人工智能转变为企业级平台。

多租户代理平台的商业价值

多租户不仅仅是一种架构上的便利，更是一种业务加速器。随着智能代理在部门和团队中激增，组织必须在不重复基础架构或分散治理的情况下支持增长。

多租户系统的主要业务优势包括：

- 可扩展性 — 多租户代理平台允许内部团队、业务部门或客户更快地加入人工智能功能，而无需定制环境。
- 成本效益 — 共享基础架构最大限度地减少了冗余部署，整合了运营成本，并简化了跨环境的维护。
- 治理和降低风险 — 集中式策略控制、身份模型和可观察性可帮助代理在所有租户中更安全、更合规地运营。
- 服务可重用性 — 为了促进重复使用和减少重复，可以将具有租户意识的代理作为内部服务提供，例如用于充实、合规或汇总。

多租户系统的示例用例包括以下内容：

- 跨子公司部署的合规代理通过租户特定的配置来调整其逻辑以适应当地法规。这样就无需为每个区域建立单独的代理。
- 内部工作流程自动化代理为具有不同数据边界和权限的多个部门提供服务。它可以保持隔离，同时加快任务完成速度。

通过将代理设计为 multi-tenant-aware服务，组织可以避免孤立的人工智能计划的开销。相反，他们培育了一个统一的情报平台。该架构可实现可扩展的部署、运营一致性和更高的投资回报率。它还可以更轻松地在整个企业中扩大人工智能的采用。

重点领域 4：通过身份、护栏和可观察性建立信任

Job to done：“让我有信心特工会安全和可预测地采取行动，尤其是在没人监视的时候。”

自主代理挑战传统的控制模型。如果管理不当，他们独立推理和行动的能力就会带来风险。如果没有明确的所有权、可审计性或政策限制，他们可能会偏离预期行为。建立组织信任需要的不仅仅是技术可靠性。它要求可解释性、问责制和一致性。

Strategy

构建身份优先的控制系统，作为可信自治的支柱。每个代理都必须使用可验证的身份、限定范围的权限和可追踪的执行历史记录进行操作。代理应嵌入到零信任框架中，该框架包括租户绑定、上下文访问继承以及通过护栏和策略引擎执行运行时。这使您可以根据组织规则和风险状况审计、撤消或限制代理操作。

通过智能护栏在运行时嵌入信任执行。这包括基于行为模式或工作量条件的费率控制和限制，在自动缩放的同时强制执行资源边界，以及用于评估风险的决策评分。构建触发器，以便在超过阈值时参与human-in-the-loop工作流程。

每个代理还必须透明且易于解释。通过日志、轨迹和推理摘要嵌入结构化遥测，以揭示决策逻辑。Support 决策跟踪和影响追踪。这可以帮助您将代理操作与关键指标或结果联系起来。实施偏差检测机制，监控与预期行为或策略的偏差。

引入能够持续观察代理行为和系统模式的反射剂。他们应该实时举报异常或不一致之处。这些代理有助于治理反馈循环，从而启动能力的重新验证、调整或停用。

建立管理委员会，负责审查代理政策、批准能力变更并监督事件响应协议。必须获得、衡量和不断加强信任。

AWS 为实施此信任框架提供了坚实的基础：

- AWS Identity and Access Management (IAM) 强制执行基于角色的执行和权限限制
- Amazon CloudWatch 并AWS X-Ray支持全面的可见性和可追溯性。
- Amazon GuardDuty 并AWS Config检测安全异常或政策偏差。

这些服务共同实现了大规模的身份强制执行、运行时安全和基于信任的治理。它们可以帮助使自主系统既强大又可靠。

可信自治的商业价值

随着代理变得更加自主，信任成为企业采用、治理和运营绩效的关键驱动力。建立身份、可观察性和防护的基础可以帮助组织在不牺牲治理或控制的情况下将代理人工智能扩展到敏感领域。

主要业务驱动因素包括以下几点：

- 治理保障 — 强大的身份模型、审计跟踪和权限界限可降低合规风险并支持监管一致。
- 操作连续性 — 运行时护栏和异常检测有助于防止意外行为，并支持从边缘故障中自我恢复。
- 利益相关者信心 — 决策可解释性和遥测可建立内部利益相关者、风险经理和外部审计师的信任。
- 事件弹性 — 嵌入式可观测性可加快根本原因分析和问题出现时的响应时间。

示例使用案例包括：

- 在金融服务领域，欺诈检测代理必须公开其推理，使用可追踪的身份记录每项操作，并在严格限定的 IAM 角色下运营。

- 在医疗保健领域，自主分诊代理必须执行运行时安全检查，在达到阈值时升级为人工审查，并提供完整的临床监督日志。

通过将信任机制嵌入代理生命周期，组织可以允许其系统在问责的情况下自主运行。该基金会降低了风险，并使代理人能够以透明和诚信的方式代表企业行事。

最终，可信自主性让用户和领导层都有信心在核心运营中扩展智能代理，从而加快采用速度。

重点领域 5：管理生命周期

Job to done：“确保我的队伍能够随着时间的推移改进特工，而不会出现混乱或英雄气概。”

与仅由代码塑造的传统应用程序不同，代理行为也受提示、内存、工具和训练上下文的影响。这些因素会随着时间的推移而变化。漂移会削弱可靠性，抬高成本，使调试几乎不可能。如果没有生命周期控制，代理就会停止提供价值，开始积累风险。

Strategy

DevOps 为代理 (AgentOps) 建立一种实践。整合专为代理量身定制的 CI/CD 管道。使用这些管道来测试提示输出、验证工具集成和分析性价比行为。维护提示、策略和模型交互的版本历史记录。

使用来自可观测性数据的反馈回路来启动再训练、提示调整或代理停用。纳入全系统反思机制，例如改进登记册，使学习制度化。

构建性能遥测仪表板，显示决策准确性、延迟、成本和可靠性。要使用 AWS 基础架构简化和加速生命周期管理，团队可以使用代理工具包。Strands Agents SDK 就是一个例子，它提供了结构化工具，用于即时版本控制、工具注册以及AWS CodePipeline与 AWS 服务、和 CI/CD 集成。AWS Cloud Development Kit (AWS CDK)AWS Lambda此外，使用 Amazon S3 和亚马逊弹性文件系统 (Amazon EFS) 来存储代理工件和训练数据。用于自动AWS Step Functions执行复杂的再训练或验证工作流程。当代理需要自定义模型调整或微调 LLM 编排以外的工作流程时，您可以使用 SageMaker Amazon AI。生命周期纪律将代理从实验转变为耐用、不断演变的资产。

随着时间的推移，这种生命周期系统构成了创新的支柱。它可以帮助您灵活地重组、重新训练和重新部署功能。这会将代理层转变为一个生命系统，能够根据反馈和机会而演变。

生命周期管理的商业价值

有效的生命周期管理是代理绩效和成本效益的关键驱动因素。它可确保智能代理在不断发展时继续提供准确、可靠和与价值一致的结果。默认情况下，代理不会保持价值。它们必须与不断变化的业务需求、

工作流程和数据环境同步发展。纪律严明的 AgentOps 团队可以帮助客服人员随着时间的推移保持准确、高效并与企业目标保持一致。

主要业务驱动因素包括以下几点：

- 性能一致性 — 持续测试、及时验证和再培训可帮助代理在不断变化的条件和数据集中保持决策质量。
- 成本优化 — Telemetry 驱动的分析可识别效率低下的工具、高代币提示或不必要的执行。然后，您可以进行调整以降低运营成本。
- 更快的迭代 — 生命周期自动化 CI/CD 可加快开发周期，帮助团队充满信心地试验、部署和改进代理。
- 降低风险 — 即时版本控制、回滚支持和结构化评估机制有助于防止回归并支持安全、可靠的变更管理。

以下是使用案例示例：

- 监控客户支持代理的延迟、模型成本和用户反馈。可观察性揭示了成本峰值，这促使重新调整其嵌入式提示和备用模型逻辑。
- 合同摘要代理会根据法律团队的反馈进行更新。版本化提示在正式发布之前在沙盒环境中进行测试，以支持安全性和质量。

通过结构化的生命周期管理，组织可以从被动维护转向主动、持续的改进。代理成为自适应数字资产，可以根据业务目标进行衡量、完善和重新验证。这种做法将代理生态系统转变为高性能、具有成本意识和弹性的系统，在与变化同步的同时提供持久的价值。

重点领域6：使代理模式与商业模式保持一致

Job to done：“让我看看影响力，这样我就可以证明持续投资是合理的。”

如果不与业务结果挂钩，即使是具有技术能力的代理人也会成为负债。代理商必须服务于效率、盈利或战略差异化。然而，大多数企业都在努力定义代理如何适应定价、包装或使用模式。如果不与业务价值保持明确一致，就很难证明扩大甚至维持投资是合理的。

Strategy

采用产品管理实践。将代理视为可获利的服务，具有可衡量的投资回报率。根据决策、会议或结果定义定价策略。然后，将代理能力打包到与客户群或内部业务部门一致的分层产品中。

为了促进可持续发展，组织必须通过代理部署来捕捉直接价值和增长倍数。考虑使用以下 ROI 指标来衡量即时价值：

- 每项决策的成本 — 将代理处理成本与人工同类成本进行基准比较。
- 时间压缩 — 量化加速周期的价值，例如更快的销售或批准。
- 减少错误-衡量通过提高准确性、一致性和合规性而节省的成本。

除了这些直接收益外，代理商还可以解锁以下长期增长机会：

- 功能堆叠 — 组合代理服务，创建特定领域的垂直解决方案。
- 网络效应 — 通过多智能体生态系统增加价值，在这些生态系统中，配位化合物的效用。
- 市场扩展 — 通过外部消费、支持代理的服务创造新的收入来源。

根据业务指标（例如成本节约、转化率提升或 time-to-resolution）创建反馈循环，以推动代理的持续发展。分析使用情况遥测和用户满意度分数，以完善您的价值一致性和路线图优先级。通过将代理能力直接与商业模式联系起来，组织将自己定位为捕捉可持续、可复合的价值，而不仅仅是技术成果。

以下内容通过提供强大的跟踪和盈利框架来 AWS 服务 支持这种调整：

- AWS Cost Explorer而且 Amazon CloudWatch 可以深入了解每位代理的成本和运营效率。
- Amazon API Gateway 支持代理终端节点按流量计费、速率限制和分层定价。
- AWS Marketplace为出版代理商和代理解决方案作为商业产品提供渠道。

这些服务可帮助您将代理功能转变为可扩展、以价值为导向的数字产品，这些产品与企业增长和盈利战略保持一致。

不断演进的代理人工智能软件交付

现代软件交付是由一个简单的假设塑造的，那就是您可以控制自己交付的系统。您可以定义需求，编写逻辑，根据预期结果进行测试，并部署可预测的服务。即使是敏捷和 DevOps 方法也仍然依赖于这样一个原则，即每个冲刺都能提供确定性、可验证的东西，并且在很大程度上处于人类监督之下。

Agentic AI 颠覆了这一基础。代理系统解释、推理和改编，而不是遵循脚本。他们的行为取决于你编写的代码、他们操作的上下文、他们获得的输入、他们可以访问的工具以及他们被分配的目标。简而言之，他们不听从命令；他们追求结果。

这使得交付与其说是控制，不如说是对齐。与其提供说明，不如塑造它的行为方式。这意味着传统的软件开发生命周期 (SDLC) 不再适合，因为它是为基于逻辑的、人为控制的系统设计的。

本节包含以下主题：

- 代理人工智能的意向区域
- 改变代理人工智能的交付生命周期
- 让团队为代理人工智能做好准备

代理人工智能的意向区域

我们需要的不是定义、构建、测试和发布等僵化的阶段，而是包含自主性、不确定性和出现性的模型。相反，你使用意图区域。意图的 z 定义了一个有限的空间，在这个空间中，代理可以在约束条件下自主地进行操作。目标是从微观管理每项任务转变为设计支持人员可以安全地行动、学习和协作的环境。您可以指定什么（期望的结果）、原因（意图）和护栏（约束、策略和信任边界）。考虑到这些界限和这些信息，代理商会弄清楚如何做。

与其说是装配线，不如将环境视为空域。你可以控制谁可以进入，他们能做什么，以及他们可以去哪里。但是一旦进去，他们就可以根据需要自由导航。这就是代理系统在没有混乱的情况下扩展的方式。

这不仅仅是哲学上的转变；更是一种实际的转变。无法通过单元测试对基于代理的系统的非确定性输出进行全面测试。它不能像静态二进制文件那样进行版本控制。代理会随着时间的推移而变化，适应新数据，并以不可预测的方式与其他系统交互。试图使用传统模型交付它们会导致架构脆弱、无法扩展。在最坏的情况下，它会导致人们对你无法实际治理的系统的错误信心。

当团队采用基于意图的交付时，他们将获得两个优势：

- 控制最重要的地方 ——他们定义界限而不是输出。

- 通过委托实现可扩展性 — 它们使代理能够处理人类无法硬编码的复杂性。

这就是你如何从孤立的原型转变为真实的生产级代理系统，这些系统可以反复可靠地提供价值。

改变代理人工智能的交付生命周期

为了支持智能的自适应行为，必须将 SDLC 从确定性控制转变为自适应意图。以下是发展适用于代理人工智能的传统 SDLC 所必需的更改：

- 规划变成意图设计。团队定义目标、限制和预期的代理行为。政策和成功标准是根据一致性而不是逻辑来制定的。
- 建筑变成了脚手架。团队专注于定义角色、接口、护栏、后备机制和可观察性，而不是编写每条决策路径的脚本。
- 测试变成行为评估。团队不是断言具体的产出，而是验证代理人是否保持在可接受的范围内，并在不同的输入下实现意图。
- 部署变成了持续的编排。Agentic 系统部署了运行时控制、实时监控和反馈通道，可实现实时调整。
- 迭代变成了反馈和适应。团队不是传统的代码更改补丁周期，而是观察代理是如何演变的、他们在哪里成功或何时漂移。必要时，团队会通过更新约束、再培训以及添加或修改控制机制进行干预。

侧重于迭代、实验和快速反馈的现有实践已经过时了。向代理系统的转变并不是对敏捷原则的拒绝。实际上，这是它们的自然演变。敏捷思维强调适应性、反馈和可行的解决方案，而不是僵化的计划。这与代理系统的本质完全一致，代理系统可以实时学习、适应和响应上下文。如果你已经在运行较短的周期，快速验证假设并通过持续交付来管理不确定性，那么你完全有能力领导这种过渡。

但是有一些关键的区别。传统的敏捷方法假设交付的东西是确定性的。它假设，一旦建成，事物就会表现得一致且可预测，对于相同的输入，结果是可重复的。这种可重复性可帮助您放心地进行调试、测试和迭代。代理系统打破了这种模式。它们具有概率性、上下文敏感性，并且能够独立演变。这意味着一些敏捷实践变得不那么有用了，例如基于故事完成的速度跟踪、严格的验收标准或确定性的冲刺计划。

传统 SDLC 的以下方面适用于代理人工智能：

- 迭代开发和交付
- 将客户反馈作为主要信号
- 跨职能协作
- 持续集成和部署

对于代理人工智能，传统 SDLC 的以下方面必须不断发展：

- 将完成重新定义为与意图一致。重点关注代理的行为是否在定义的限制条件下满足其预期目标。
- 从接受标准转向行为护栏。
- 将 d on e 的定义扩展到包括运行时就绪性，其中包括支持持续学习和信任的可观察性、可解释性和反馈机制。
- 优先考虑实时反馈回路和行为跟踪，而不是前期规划

好消息是你不需要抛弃 SDLC 剧本。你只需要将其从管理规范发展到塑造行为即可。在代理系统中，成功不仅在于软件能否运行，还在于软件的行为方式。

让团队为代理人工智能做好准备

软件工程不会消失。它在不断演变。工作从编写函数转向塑造智能行为的框架和控制机制。在代理人工智能的世界中，建筑不再是困难的部分，管理崛起才是困难的部分。对于大多数工程团队来说，这种演变感觉就像是思维方式的转变，而不是技术上的飞跃。而不是问 “系统会做什么？” 问题变成了 “我们授权它追求什么，我们如何知道它是否保持正轨？”

对于工程团队而言，向代理 AI 的演变需要进行以下更改：

- 文化转变 ——团队必须适应他们无法完全控制的系统中的不确定性和自主权。
- 新角色 ——意图设计师、行为测试人员和可观察性工程师成为交付的核心。
- 共享语言 — 团队需要对目标、护栏和成功信号有清晰、共同的理解，就像他们曾经需要规格和测试用例一样。

随着生成式人工智能的成熟，我们将看到更多的代理系统与客户、产品和运营进行交互。成功的组织不会是拥有最佳模式的组织。它将能够自信、控制和快速地将代理集成到现实世界的工作流程中。这意味着交付模型和工程团队必须共同发展。意图区域可以让你抽象地做到这一点。它们可以帮助您在不放弃问责制的情况下实现自主权。他们还提供了跨团队共享的框架，以帮助管理无法硬编码的系统。

有关让团队为代理人工智能做好准备的更多信息，请参阅本指南的为大规模代理 AI 做好业务准备部分。

让企业为大规模代理人工智能做好准备

随着本指南中描述的重点领域的融合，代理人工智能从孤立的功能转变为可以理解为能力平台的统一情报层。这个平台不只是执行任务。它可以跨领域进化、适应和协调。代理成为模块化、可重复使用且可发现的服务，可加速创新，减少认知负担，并在整个企业中推动可衡量的成果。该平台视图为嵌入到整个运营模型中的可扩展智能奠定了基础。

实现代理 AI 需要的不仅仅是部署智能代理。它要求从根本上改变企业组织团队、设计流程和管理技术的方式。正如向云端的转变或 DevOps 重新定义的运营模式一样，agentic AI 引入了决策自动化、持续学习和自主协调的新时代。成功取决于围绕这种新的运营理念调整系统、人员和流程。

本节包含以下主题：

- 协调团队和所有权模式
- 管理变革和组织就绪
- 为互操作性和协作进行架构
- 将治理融入代理结构
- 采用决策为先的运营思维
- 根据目的和意图进行扩展

协调团队和所有权模式

迈向成熟的第一步是跨职能协调。企业必须建立由 AI/ML 从业人员和领域专家组成的 AgentOps 团队，例如分布式系统架构师、软件工程师、产品负责人、合规主管和平台架构师。这些团队共同拥有代理的整个生命周期，从设计和部署到再培训和监控。

代理的配置和发布应遵循云原生实践，例如将AWS Cloud Development Kit (AWS CDK)和AWS CodePipeline用于基础架构即代码和自动部署。这种结构促进了共同的问责制并加速了迭代。正如DevOps统一开发和运营一样， AgentOps 将智能与治理和执行联系起来。

为了提高效率，这些团队还需要一种共同的语言。业务利益相关者必须了解什么是代理人，他们是如何运作的，以及他们推动的结果。培训和内部能力提升至关重要。通过揭开代理人的神秘面纱，将这种心理模型嵌入到日常对话中，组织可以解锁更广泛的参与和更一致的创新。

为了加快代理的开发和集成 AWS 服务，团队可以采用诸如 Strands Agents SDK 之类的框架，该框架为脚手架、配置和打包代理提供了基于 CLI 的工具。Strands Agents 旨在与 AWS 基础设施

（例如 Amazon Bedrock、A AWS Lambdama EventBridge zon、和）无缝协作。 AWS CDK AWS CodePipeline它支持快速原型设计和部署，同时保持生产级标准。

但是，光靠结构和工具是不够的。扩展代理人工智能需要深思熟虑的文化、教育和领导准备，以确保采用在整个组织中扎根。

管理变革和组织就绪

成功扩展代理 AI 需要的不仅仅是部署基础架构或智能代理。它要求对组织变革采取结构化方法。这包括文化准备、技能发展、指标驱动的反馈回路以及高管协调，以确保采用既是有意的，又是可持续的。

促进文化演变

- 将特工定位为队友，而不是替补球员，以减少阻力并建立信任。
- 就代理的能力和限制进行透明的沟通，以设定切合实际的期望。
- 制定明确的移交协议，规定代理人何时应将决策上报给更高权力机构或将部分流程委托给人类合作者。

建立技能发展框架

- 提供专为工程师、产品经理、领域主管和合规官员量身定制的基于角色的培训。
- 创建卓越中心，共享最佳实践、工具模式和可重复使用的资产。
- 通过指导计划将人工智能专家与领域专家配对，以弥合知识差距。

定义指标和反馈循环

- 将技术和业务 KPIs 与战略价值联系起来，以评估影响。价值示例包括决策延迟、分辨率准确性和成本节约。
- 系统地、持续地捕获用户对表面摩擦点和采用挑战的反馈。
- 定期进行回顾，以评估代理的表现、使用趋势和改进机会。

从高层调整领导力

- 通过将代理计划与战略成果和投资回报率联系起来，获得高管的支持。
- 组建包括技术和业务领导在内的跨职能治理委员会。
- 量身定制沟通策略，以提高所有组织层面的清晰度和参与度。

这种系统的变更管理方法可确保技术实施与组织成熟度相匹配。它为信任、采用和长期商业价值奠定了基础。

为互操作性和协作进行架构

孤立的代理部署带来了本地胜利。但是，当代理人能够动态地发现、调用彼此并相互协作时，企业价值就会显现出来。这意味着要为代理注册、身份验证和能力交换定义标准。从架构上讲，这反映了从单体向微服务的转变，微服务是可组合、可重复使用和松散耦合的单元，可以共同解决复杂的问题。

诸如 A2A 和 MCP 之类的新兴协议是基础协议。它们支持代理、工具和内存系统之间的语义互操作性。A2A 支持对等级交互，允许代理协商任务所有权、共享上下文和协调工作流程。MCP 通过提供用于在代理及其环境之间交换上下文数据的共享架构来补充这一点。它标准化了函数的调 APIs 用、访问和状态维护方式。这些协议共同促进了整个代理生态系统的可扩展性、一致性和长期可维护性。

治理仍然至关重要。控制层（例如仲裁服务器代理）可在不引入集中瓶颈的情况下实现策略感知型委派。这些代理充当信托经纪人。他们强制划定边界，同时允许其他代理进行自我组织。代理协作可帮助组织以敏捷性和信任度扩展其代理人工智能生态系统。

将治理融入代理结构

自主权越大，风险越大。治理必须从第一天起就嵌入到代理架构中。这包括定义政策界限，确定允许代理执行的范围，强制实施身份模型来确定他们代表谁行事，以及实现可解释性和可追溯性。可观测性系统必须使用诸如 A ma zon CloudWatch 和之类的服务捕获代理行为的遥测数据 AWS X-Ray，这些服务提供跨代理工作流程的集中日志和分布式跟踪。反射剂可以根据这些遥测数据持续审计和评估性能。

随着代理生态系统的成熟，治理也必须不断发展。随着代理人的能力和自主性越来越强，监督机制必须变得更具适应性。策略更新、功能门控和运行时行为限制必须是动态的，并且可以大规模执行。信任不是一项附加功能。它通过架构、行为和流程不断得到加强。AWS Identity and Access Management (IAM)，并在AWS AppConfig跨代理强制执行安全身份、运行时权限界限和特定于环境的行为切换方面发挥着至关重要的作用。

采用决策为先的运营思维

传统的自动化侧重于流程效率，即更快、更可靠地运行预定义的脚本或工作流程。相比之下，Agentic AI 引入了决策优先的自动化。客服人员实时评估情境、权衡选项并调整行为。这种从执行优先向决策至上的思维方式的转变需要对成功指标和结果进行新的思考。代理人工智能的成功不是完全通过任务完成来衡量成功，而是通过决策与意图、政策和不断变化的条件的一致程度来衡量的。

组织必须评估决策质量和对变化的响应能力，而不是仅仅衡量任务完成情况或周期时间。 time-to-action KPIs 应包括以下指标：

- 决策质量 — 代理对特定用户或场景的个性化响应有多好？ 它是否做出了与业务目标和用户情境相一致的细致入微的决策？
- T ime-to-action — 代理评估情况并做出反应的速度和智能程度如何？ 延迟是否足够低，足以让人感觉自适应和像人一样？
- 认知卸载 — 代理能够代表人类处理多少手动分析、分类或例行决策？ 它减少了精力还是只是转移了精力？

拥抱决策至上的思维方式的企业可以变得更具弹性、适应性，并能够在新的复杂性水平上运营。

根据目的和意图进行扩展

成功扩展代理人工智能并不是要尝试更多工具。它旨在构建一个持久的企业智能层。这需要在平台基础架构、运营文化、治理框架和战略协调方面进行投资。企业必须采取有意的方法。他们不能将代理视为实验，而应将其视为数字运营模型的核心组成部分。

与 Well-Architect AWS ed F ramework 保持一致，可以帮助您的系统在可靠性、安全性、性能效率和成本优化方面达到企业标准。诸如 Strands Agents SDK 之类的工具可以通过提供结构化提示、工具注册和 CI/CD 就绪来加快这一旅程。这有助于团队使用熟悉 AWS 的工作流程从实验转变为可扩展的交付。

Agentic AI 不是工具；而是智能嵌入运营方式的转变。做好相应准备的组织可以实现更多自动化，更智能地运营，更快地适应，并在日益复杂的世界中创造持久的优势。

实现代理人工智能的结论

Agentic AI 所代表的不仅仅是技术转变。它标志着企业新操作系统的出现。接受这种转型的组织会超越狭隘的自动化用例，将智能融入其运营的基础。这种转变旨在重新设计决策的制定方式、系统的适应方式以及如何大规模实现成果。

在复杂性不断增长、实时需求和信息过载的时代，传统的脚本自动化模式已经达到了极限。现在，成功取决于能否将智能直接嵌入到工作流程中，从而打造出能够感知、推理、行动和演变的系统。Agentic AI 可以将自主权与目的相结合，将决策与治理相结合，将适应性与问责制相结合。

这种过渡需要从执行优先转变为决策至上的思维。代理系统不只是按照说明进行操作。他们解释目标，权衡利弊，并在规定的限制范围内追求成果。在这种情况下，衡量成功的标准不仅仅是任务完成情况。它还通过实时决策的质量、敏捷性和可解释性来衡量。组织必须重新考虑指标、激励措施和系统设计，以支持在不确定性下智能运营的代理。

运营代理人工智能不是升级。 plug-and-play这是一场建筑和文化变革。它需要在生命周期管理、信任执行、互操作性以及与业务模式保持一致方面采取纪律严明的做法。它还要求发展交付模型，例如塑造意向区域、嵌入运行时护栏以及不断调整代理行为与战略成果。团队必须采用共同语言、共享所有权，并对工作人员的绩效和安全承担共同责任。

企业准备程度可以决定谁在这个新环境中茁壮成长。Organizations 必须投资于能够扩展和创造长期价值的内部支持、 AgentOps 能力和治理框架。成功者可以构建更智能的系统，也可以建立更具适应性、更具弹性和洞察力的企业。

本指南奠定了基础。它将战略与执行联系起来，为组织构建可扩展的智能代理平台做好准备。关于代理人工智能的更广泛内容系列 AWS 提供了补充指导。要查看本系列中的其他指南，请参阅 AWS 规范性指导网站上的 A gentic AI。该内容系列提供了一个路线图，以纪律和意图实现自主权。

首先，请确定一个具有高影响力的决策空间，在该空间中，代理可以在速度、准确性或响应能力方面实现可衡量的改进。然后部署具有仪器、治理和反馈回路的有针对性的试点代理。使用它来验证价值假设，产生内部动力，并建立对方法的信任。通过学习获得动量。

Agentic AI 不是目的地；它是一个与您的业务一起发展的能力层。它代表了向以情报作为基础设施的长期转变。在该领域处于领先地位的组织可以实现更多自动化，更快地做出响应，更好地适应，并构建能够应对企业规模复杂性的运营模型。

用于实现代理人工智能的资源

AWS 服务

以下内容 AWS 服务 和功能可以帮助您在以下方面构建和操作代理人工智能系统： AWS Cloud

- Amazon API Gateway 可以公开可扩展的代理功能，并提供基于使用量的定价。
- AWS AppConfig为跨租户或环境的代理提供运行时配置管理和功能切换。
- Amazon Bedrock 是一项基础模型服务，代理可以使用它进行推理、生成和迅速执行。
- AWS Cloud Development Kit (AWS CDK)是一项基础设施即代码服务，可用于部署和管理代理堆栈。
- AWS CloudTrail记录事件历史记录，以便您可以跟踪代理活动、审计跟踪和集成行为。
- Amazon CloudWatch 可以管理日志、指标和警报，以监控代理性能和多代理协作行为。
- AWS CodePipeline提供了可用于测试、验证和部署代理代码的 CI/CD 自动化功能。
- Amazon Cognito 是一项身份服务，可用于管理多代理系统中的用户和租户身份验证。
- AWS Config为代理策略和环境配置提供合规性和偏差检测。
- AWS Cost Explorer可以跟踪代理级别的使用情况，并帮助调整成本以最大限度地提高投资回报率。
- Amaz on DynamoDB 是一项存储服务，可用于存储代理内存、改进日志和上下文状态。
- Amazon Elastic File System (Amazon EFS) 是一个共享文件系统，可用于跨工作流程进行代理协作或中间处理。
- Amazon EventBridge 是一个核心事件总线，可用于在代理结构中路由任务和编排通信。
- Amazon Pip EventBridge es 可以简化连接代理和服务的事件摄取和路由。
- Amazon GuardDuty 提供威胁检测和异常监控，可以支持安全代理执行。
- AWS Identity and Access Management (IAM) 可帮助您定义代理执行和数据访问的精细权限。
- AWS Lambda是一种无状态计算服务，可以运行代理逻辑和蜂群无人机。
- AWS Marketplace是一个外部分销平台，您可以使用它作为商业产品提供代理功能。
- AWS Organizations是一项跨账户管理和策略实施服务，可帮助您管理多租户代理基础架构。
- AWS Organizations 服务控制策略充当在账户或组织单位级别控制权限的护栏。
- Amazon Quick 是一个基于人工智能的生成式商业智能 (BI) 平台，可帮助您分析数据、创建可视化效果、自动化工作流程以及与组织中的其他人协作。
- AWS Resource Access Manager (AWS RAM) 可以帮助您在账户和代理服务之间共享功能。

- Amazon SageMaker AI 是一项可用于模型训练、微调和基础模型之外的推断的服务。
- Amazon Simple Storage Service (Amazon S3) 为提示库、模型工件和代理生成的数据提供对象存储。
- AWS Step Functions是一个工作流引擎，可以帮助您协调多代理流程和再训练管道。
- AWS X-Ray提供分布式跟踪，可用于跟踪代理决策流程和服务依赖关系。

其他 AWS 资源

- 代理人工智能的基础 AWS
- Agentic AI 模式和工作流程已开启 AWS
- Agentic AI 框架、协议和工具已启用 AWS
- 为代理人工智能构建无服务器架构 AWS
- 为代理人工智能构建多租户架构 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
初次发布	—	2025 年 8 月 12 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。