在上操作代理 AI AWS

AWS 规范性指导

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS 规范性指导: 在上操作代理 AI AWS

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

简介 1
聚焦领域 1
目标受众 2
目标 2
关于此内容系列 2
代理人工智能的基础 3
聚焦领域 4
意图和范围 4
Strategy 5
业务价值 6
可组合性和协作性 6
Strategy 7
业务价值 8
多租户和控制 9
Strategy 9
业务价值 9
可信的自治权 10
Strategy 10
业务价值 11
生命周期管理 12
Strategy 12
业务价值 12
业务协调 13
Strategy 13
软件交付 15
意图区域 15
不断发展 SDLC 16
为团队做好准备 17
为规模做准备 18
团队和所有权模式 18
变更管理 19
互操作性和协作 20
Governance 20
运营心态 20

扩展 21
结论 22
资源 23
AWS 服务 23
其他 AWS 资源 24
文档历史记录 25
..... xxvi

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

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。