亚马逊云科技 FOR SOFTWARE AND TECHNOLOGY

白皮书

Agentic 时代 SaaS 的反思与重构

深入剖析 Agent 如何在业务、技术和战略层面影响 SaaS 服务提供商

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

亚马逊云科技 FOR SOFTWARE AND TECHNOLOGY

白皮书

Agentic 时代 SaaS 的反思与重构

深入剖析 Agent 如何在业务、技术和战略层面影响 SaaS 服务提供商

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

目录

摘要 3
1. 核心概念 4
引入多租户机制 5
Agent 模式 6
多 Agent 系统 7
新兴的用户角色与交互模型 8
2. 产品战略 9
加速产品开发 9
产品的战略演进路径 10
其他战略性考量 12
实施框架 12
3. 多租户 Agent 架构 14
身份与租户 14
构建能够感知租户的 Agent 15
Agentic SaaS 环境下的 SaaS 控制平面 17
强制执行租户隔离 18
4. SaaS Agent 运维 (AgentOps) 20
定义 Agent 的新一代 KPI 20
5. 业务战略 21
市场范围规划策略 21
定价策略 22
Agentic SaaS 定价的实施框架 25
市场进入策略 27
总结 28
声明 28
贡献者 29

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

摘要

Agentic AI 的出现，为 SaaS 行业的产品构建、运营和盈利模式带来了全新的思考维度。Agentic AI 技术有望颠覆整个行业，这也为软件公司提供了一个重新审视系统构建、集成、交付及差异化策略的契机。对软件即服务 (SaaS) 提供商来说，这种影响尤为显著，他们正着手评估 Agent 将如何彻底重塑其产品的形态。同时，这也引出了更深层次的议题：SaaS 企业应如何将 Agent 全面融入其业务、技术及战略体系？

面对 Agentic AI 为 SaaS 服务提供商客户带来的技术与业务挑战、问题及机遇，我们始终致力于运用自身的专业知识和解决方案，助力他们取得卓越成果。我们的愿景，是在保障安全、可靠和卓越运营的坚实基础上，平衡并推动业务的快速创新。亚马逊云科技之所以是 Agentic AI 的理想之选，关键不仅在于我们的前沿技术，更在于我们久经考验的安全、治理、规模和可靠性基础，这些都是成功部署自主系统的必备要素。

无论客户的 Agentic AI 之旅处于哪一阶段，亚马逊云科技都能与之同行。无论客户是希望借助 Amazon Q 和 Kiro 这类开箱即用的 Agent 应用程序提升生产力，还是有兴趣利用 Amazon Bedrock 和 AgentCore 开发定制化 Agent，抑或是想采用包括我们自研的 Amazon Nova 在内的业界优秀基础模型，乃至希望构建具备顶尖安全与治理能力、并为 Agentic AI 工作负载深度优化的基础设施，我们都能全方位地提供支持。

亚马逊云科技 Agentic AI 产品组合

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

Gartner 预测，至 2028 年，将有至少 15% 的工作决策由 Agentic AI 自主完成。面对这一趋势，亚马逊云科技致力于帮助软件和技术领域的客户充满信心地部署这些颠覆性技术，并始终坚守在信任、安全与性能方面的高标准。

本白皮书将聚焦于构建支持 Agent 的 SaaS 系统（下称“Agentic SaaS”），并深入探讨其在业务、技术及运营层面带来的具体影响。本白皮书旨在全面梳理 Agent 将在哪些关键领域影响甚至重塑企业构建下一代 Agent 驱动型服务的方式。本白皮书将着重阐明，Agent 的引入将在哪些方面对 SaaS 解决方案的产品打包、用户消费、市场发现、商业变现及日常运营等环节带来颠覆性的影响。我们希望借此帮助 SaaS 服务提供商更清晰地洞察 Agentic SaaS 的宏观格局，并揭示出那些在制定未来发展蓝图时必须考量的关键因素。

1. 核心概念

Agent 技术有望对 SaaS 行业格局带来深远影响，这要求企业必须从根本上重新审视自身的核心战略。Agent 很有可能重塑 SaaS 体验的本质，它将赋能服务提供商开拓新市场、探索新的盈利模式、解锁新的集成范式，并最终成为业务增长的核心引擎。对众多服务提供商而言，Agent 将颠覆他们向客户交付价值的传统方式，标志着软件交付模式的下一次革新。

然而，每家 SaaS 服务提供商释放这一新潜能的方式不尽相同。将 Agent 引入现有 SaaS 环境的方式，会因多个维度的差异而有所区别。Agent 能否作为独立产品存在？Agent 是否应完全嵌入到 SaaS 环境内部？Agent 是否可以充当一种“门面”，为现有 SaaS 解决方案提供全新的交互入口？以上这些，是每家 SaaS 服务提供商都必须直面的战略抉择。您需要找到一种集交付、集成和交互于一体的理想模型组合，能够精准匹配您自身的业务及客户的需求。

为厘清这一议题，我们不妨从推动 SaaS 交付模式得以广泛应用的核心价值谈起。SaaS 模式的魅力，主要源于其在提升敏捷性、成本与运营效率、推动创新及实现规模经济方面的强大能力。在此模式下，SaaS 服务提供商能够利用敏捷与高效这两大优势，驱动业务增长、拓展市场，并提升用户采纳度。

Agentic SaaS 的出现，并不会改变上述任何一条核心原则。Agent 应被视为对现有体验的增强与补充，为扩展 SaaS 产品的能力与价值开辟了全新的维度。引入 Agent，意味着业务和技术团队必须积极采纳全新的战略、交付模型与集成方案。借助 Agent 的自主性与智能，您能够为客户交付前所未有的增强功能，进而开拓全新的市场和机遇。

AI Agent：
AI Agent 是一种软件系统，能够在几乎不依赖人工干预的情况下，通过推理、规划与行动自主完成既定目标。

Agentic AI：
Agentic AI 标志着 AI 的一次重大演进：从被动响应的助手进化为主动、自主的系统，能够理解、决策并执行任务，几乎无需人类持续监管。这类 Agent 具备思考、规划、记忆和从经验中学习的能力，正在金融、医疗保健、零售、客户服务等各行各业推动效率提升与模式创新。

Agentic SaaS：
Agentic SaaS 是指将 AI Agent 深度嵌入到 SaaS 平台中，使其能够规划多步骤工作流、跨系统采取行动、基于结构化和非结构化数据进行推理，并与人类及其他 Agent 协同合作，最终创造价值。

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

从宏观层面来看，这些概念都清晰易懂。然而，我们仍需深入探讨，将 Agent 引入多租户 SaaS 环境究竟意味着什么——我们既要拥抱 Agent 带来的强大能力，又必须坚守 SaaS 的核心原则。为此，我们必须从一个基本问题入手：在 Agentic SaaS 的整体体验中，用户会在哪些场景、以何种方式来使用 Agent？图 1 为我们接下来的讨论提供了一个很好的起点。

图 1：专用与共享 Agent 模型

此处描绘了两种截然不同的 Agent 模型。图左侧展示的是专用模型，其中每个 Agent 都部署在与特定客户绑定的专属环境中。在这种模式下，Agent 的能力及其知识获取范围，都将局限于单个客户的环境之内。同时，该模型也继承了为客户维护专用环境所固有的复杂性与优势。尽管此模式在某些场景下可行，但其可扩展性并不理想。在此模型下，服务的客户越多，您的 Agent 环境在运营、成本和效率方面的优势就越会被削弱。

与之相对，图右侧的共享模型中，单个 Agent 统一部署在服务提供商的环境里。该 Agent 能够处理来自多个客户的请求，并基于所有客户的综合使用经验不断演进。每增加一个新客户，都只是为这个共享 Agent 增添了一个新的有效客户端。这种方式能够实现更佳的规模效应与运营效率，摒除了为每个客户单独部署一次性 Agent 的需要。对于这种共享 Agent 模型，我们很自然会认为，这种跨多个客户工作负载共享基础设施的模式，与传统的 SaaS 模型如出一辙。在此模型下，Agent 资源的消耗量与用户的活跃度直接相关，最终转化为显著的规模经济效益。此外，该模型采用集中式的管理和运营，降低了专用 Agent 模型分布式部署产生的复杂性。

为 Agent 引入多租户机制

尽管这种方法与传统 SaaS 模型高度一致，但它在处理请求时，并未将客户的具体情境纳入考量。从根本上来说，每个客户调用 Agent 的同一个操作，都会得到完全相同的结果，缺乏个性化。而真正的根本性转变在于，Agent 的内部机制能够为每个传入的请求附加特定的上下文，并依据该上下文生成精准定制的结果。在这种新模型下，Agent 的使用者不再是笼统的“客户”，而是具有明确身份的“租户”。这一转变，正式将“多租户”的概念引入了 Agent 的核心设计。

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

“客户”与“租户”这一用词上的区别，表面看可能微不足道。然而，一旦引入“租户”的概念，就为 Agent 的实现、用户引导、部署、授权、隔离及运营等方方面面，都增添了全新的维度。它还创造了新的机遇：我们得以在单个 Agent 内部，为每个租户打造专属体验，让 Agent 的能力、知识、数据和工作流都能产出针对特定租户的定制化结果。Agent 的体验不再千篇一律，而是能根据不同租户的具体情况进行响应与调整，从而为每个租户提供具有差异化的独特价值。

归根结底，这关乎一种思维模式的转变。我们不仅需要关注 Agent 的功能本身，还要系统性地思考其规模化、部署和运营模式。这种转变也让我们必须去思考，如何在整体的 Agent 模型中，实现我们所追求的规模经济、效率、创新与敏捷性。向 Agent 多租户模式的转变，无疑将对您的 Agent 架构、商业模式及整体战略产生清晰而深远的影响。您的产品战略和愿景必须充分考虑租户机制将如何塑造 Agent 体验的最终形态。

Agent 模式

租户机制如何融入 Agent，取决于您所采用的宏观 Agentic SaaS 模式。尽管该领域仍在快速演进，但一些核心的设计模式已初见端倪。图 2 展示了这些主流模式。

图 2：Agent 模式

图左侧展示的是“产品化 Agent”模式。在此模式中，Agentic SaaS 体验本身就是通过一个独立的、代表特定 SaaS 产品功能的 Agent 来交付的。Agent 本身即构成了一个以“Agent 即服务”模式交付的 SaaS 产品。这类 Agent 通常通过应用市场进行分发，并可能需要与其他第三方 Agent 集成协作，以实现最终目标。

图中间展示了一种截然不同的理念——“嵌入式 Agent”模式，即将 Agent 植入到现有的 SaaS 产品中，使其与环境中的其他组件共存。在此模式下，Agent 的强大能力被无缝融入现有产品，从而实现对核心价值主张的替换或扩展。

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

最后，图右侧是“门面模式”，即在现有 SaaS 应用前端部署一个或多个 Agent，由它们来编排和调度底层的应用体验。本质上，这相当于一个由生成式 AI 驱动的“叠加层”，它能在不大幅改动现有系统的前提下，提供更为丰富的用户体验。

但需要注意的是，无论采用哪种模式，系统都可能仍需支持租户上下文。服务的部署位置与方式，会直接影响租户上下文在 Agent 中的传递与处理机制。例如，一个产品化 Agent 的授权模型，很可能与嵌入式 Agent 的授权模型大相径庭。同样，二者的用户引导流程也可能截然不同。但无论采用何种模式，您都必须确保所设计的 Agent 符合 SaaS 的核心原则，即实现规模经济、敏捷性与高效率。

在后续的章节中，我们将进一步探讨这些模型在架构层面的具体影响。但从根本层面来说，Agentic SaaS 的本质，就是在 SaaS 环境的各个环节引入 Agent，利用其智能与自主性为用户体验带来全新能力，并渗透到 SaaS 业务的每一个角落。这包括制定产品战略、运营 SaaS 业务、分析市场趋势、设计产品定价以及推广产品的方式等方方面面。

多 Agent 系统

在此过程中，我们还必须考虑如何将自己的 SaaS Agent 集成到一个更宏大的 Agent 协作生态中。图 3 从宏观视角展示了一个由多个 SaaS Agent 协作的系统。

图 3：构建多 Agent SaaS 系统

该模型代表了一个多 Agent 系统。在许多人看来，这代表了 Agent 发展的更高阶段，它使服务提供商和客户能够构建一个组合式的 Agent 系统，让多个 Agent 协同工作以达成目标。您会注意到图中每个 Agent 都标有“业务 1”、“业务 2”等字样，这旨在说明这些 Agent 可以跨越多个业务实体进行协作。有些多 Agent 系统可能仅限于单个服务提供商内部，而另一些则可能需要依赖第三方 Agent。

在这种模式下，我们还必须思考它将如何影响我们之前提到的“Agentic SaaS 思维模式”。在此场景下，Agent 依然需要以多租户模式运行，并在与分布式 Agent 系统中的其他成员交互时，正确传递和使用租户上下文。如此一来，您的 SaaS 体验边界便与该 Agent 系统中的其他 Agent 深度交织在一起。这就要求，必须在整个 Agent 系统中支持租户隔离模型，以严格限制租户对 Agent 资源的访问权限。

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

新兴的用户角色与交互模型

在 Agentic SaaS 模型中，用户的最终体验可能会因解决方案的具体性质而有所不同。可以预见到的是，许多 SaaS 服务提供商需要设计新的用户界面 (UI)，以便用户能更便捷地访问 Agent 的各项功能并与之交互。聊天机器人就是一个典型的例子，其中用户（即租户）通过编写文本请求，来引导 Agent 产出结果。当然，这只是其中一个例子。重点在于，为了充分释放 Agent 体验的全部价值，我们可能需要对这些 Agentic SaaS 体验的交互模型进行某种程度的重新设计。当然，对于采用嵌入式 Agent 模型的服务提供商而言，这种交互变革所带来的影响会相对较小。

随着交互体验的演变，预期中会与 Agentic SaaS 环境进行互动的用户角色 (Persona) 也可能随之改变。这意味着，租户的具体角色以及产品的潜在消费群体，都可能因系统打包和交付 Agent 体验的方式不同而发生变化。在开发 Agentic SaaS 的过程中，您需要重新审视现有的用户角色，并评估由 Agent 赋能的解决方案将如何影响您的目标用户群体。

宏观视角

需要明确的是，Agentic SaaS 并未改变 SaaS 业务的本质，它只是通过引入 Agent，扩展并增强了您所能交付的用户体验。Agent 的核心设计，仍需遵循那些一直以来指导着 SaaS 业务发展的基本原则。它们必须在延续 SaaS 规模经济、运营效率和敏捷性优势的同时，积极拥抱由 Agent 驱动的产品所带来的全新可能性。

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

2. 产品战略

Agentic AI 正在引发软件开发领域的革命，并从根本上改变 SaaS 服务提供商制定产品战略的思路。这种转变最显著地体现在产品开发周期的缩短上，它使团队能够以前所未有的速度进行创新、响应市场需求，并采纳由生成式 AI 驱动的全新流程与机制。Agent 体验还有可能催生出新的用户角色。例如，与第三方 Agent 的集成可能会引入新的用户角色，这些角色关注的不再是人机交互，而是 Agent 之间的交互与体验。

加速产品开发

AI 正在重塑软件开发的效率，让您能够以空前的速度交付新功能和产品，从而加速产品战略的落地。这种加速得益于两大助力：一是生成式 AI 编程助手，它能帮助开发者提升编码速度；二是 Agent，它能自动处理代码维护和版本升级等常规任务，从而为团队减负。这两大能力相结合，将您的开发团队从繁琐的事务中解放出来，使其能够把更多精力投入到创新性问题解决与新功能开发上。Andy Jassy 分享了这一变革在亚马逊内部产生的巨大影响：“过去，将一个应用程序升级到 Java 17 通常需要 50 人天的工作量，而现在平均耗时已骤降至短短数小时。我们估算，这为我们节省了相当于 4,500 人年的工作量。”

当常规任务交由 AI 驱动的 Agent 处理后，您的开发者便能专注于那些能够推动产品差异化、奠定市场领先地位的高价值活动。在功能极易同质化的快节奏 SaaS 市场中，将人力重新投入到创新领域，是保持竞争力的关键所在。开发速度的提升，直接转化为更短的产品和功能上市时间。

然而，这种速度的提升并非以牺牲质量为代价。通过 AI 辅助开发进行快速原型设计和迭代，可以迅速制作和优化原型，让团队能在产品发布前，更迅速地验证想法，并进行更多轮次的优化。生成式 AI 还能提升测试能力，实现更全面、更高效的测试，包括高级 A/B 测试和“金丝雀”部署。这些能力让您可以在全面上线前，先在一小部分用户中验证新功能，从而降低用户体验不佳的风险。AI 驱动的代码审查能够比人工审查更迅速、更准确地发现潜在的错误、安全漏洞和性能问题。这种自动化能够确保在整个开发流程中，代码始终保持高质量和高一致性。

开发速度的提升与质量保证的强化相辅相成，使您能够更敏捷地响应市场需求，从而在竞争中保持领先。然而，这种新范式也要求您的产品团队在运作方式和优先级排序上做出相应调整。

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

产品的战略演进路径

作为 SaaS 服务提供商，在 Agent 时代如何演进自身的产品组合，是您面临的一项关键决策。但这些决策并不是非此即彼。成功的服务提供商往往会同时推行多种策略，依据市场机遇、自身能力及资源状况，在不同路径间动态地平衡投资。图 4 从概念层面展示了不同的产品演进策略。

图 4：产品战略

此图清晰展示了与不同产品战略相关的各类选项。图表顶部展示了两种主要的战略路径：开发新功能或推出新产品。图表左侧则列出了可应用于这些路径的不同市场进入 (GTM) 策略：市场扩张与现有市场增长。以下各节将逐一阐述每种策略背后的考量因素。

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

新产品机遇

Agent 能力为服务提供商开辟了新的机遇，主要体现在两种战略路径上：一是为全新市场“构建”新产品；二是为现有客户“拓展”补充性产品。

构建策略的核心，是创造净新增产品，瞄准全新的客户群体或用户角色。这类产品以 Agent 能力为核心进行设计，其典型特征是能够将复杂的专业领域知识，普及给以往不具备相关技能的用户。例如，以往需要专家才能操作的企业软件，如今借助自然语言界面和自动化工作流，普通业务人员也能轻松上手，这极大地拓展了产品的潜在用户基础。这类新产品通常还内置了用户采纳机制，例如由 Agent 引导用户完成上手过程，并鼓励用户分享成功案例。试想一下，SaaS 服务提供商可以创建专门的 Agent，在用户引导的整个生命周期中，自动完成身份预配、资源配置和指标捕获等一系列任务。

拓展策略则是指，为现有客户群创造能够补充现有产品线的新产品，这些新产品通常具有不同的价值主张或成本结构。这种方法的优势在于，能充分利用公司现有的客户关系和领域知识，同时开辟新的收入来源。例如，一家金融软件服务提供商可以为其现有客户群，创建一个独立的、由 AI 驱动的财务规划产品，其目标用户可以是业务分析师，而非传统的会计人员。这种策略允许服务提供商在维持核心产品定价不变的同时，为新的 Agent 功能建立一套独立的定价模型。

Agent 还能为您的“**产品驱动增长 (PLG)**”战略注入新活力，提供分析用户行为模式和识别业务扩张机会的新机制，从而创造一种更为精细的 PLG 模式，其中 Agent 会主动参与，以推动 SaaS 解决方案的采纳与功能拓展。随着 Agent 能够引导用户完成复杂的工作流、根据使用模式推荐相关功能，并通过个性化用例自动展示产品价值，自助服务将变得前所未有的高效。最终，这将实现一种更具可扩展性的增长模式，降低对人工销售和客户成功团队的依赖，同时有望提升客户转化率与业务扩张率。

新用户与新功能

在对现有产品进行升级时，服务提供商通常会选择两条路径之一：**“扩展”**或**“嵌入”**。采用何种路径，以及何时采用，通常取决于多方面因素，包括竞争压力、Agentic AI 功能所能带来的价值，以及企业自身的市场地位。第一种“扩展”路径，旨在通过产品延伸来触达现有客户组织内的新用户群体，并且通常会对这些新增功能采用增值定价策略。第二种“嵌入”路径，则是将 AI 功能无缝集成到现有产品中且不额外收费，其核心目标在于提升客户留存率和打造差异化优势。

通过为现有产品增加新功能，能够充分利用您已有的市场地位和客户关系，同时通过引入 Agentic SaaS 能力来放大产品的核心价值。通过将 Agent 嵌入现有产品，您不仅可以自动化处理常规任务，还能通过模式识别与行为自适应，为用户打造更具个性化的体验。这一策略的关键优势在于企业已构筑的“数据护城河”。多年积累的客户数据与领域知识，可以用于训练高度专业化的 Agent。例如，一家 CRM 服务提供商可以嵌入这样一个 Agent：它不仅能自动完成数据录入，还能基于整个客户群的历史数据模式，提供客户行为的预测性洞见，并推荐下一步的理想行动方案。对现有产品进行升级，不仅能服务于已有的客户基础，还能通过增加产品价值稳固当前的收入来源，实现稳健的防御策略。

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

其他战略性考量

在为 Agent 时代调整产品战略时，您必须权衡若干影响成败的关键因素。这些考量因素涵盖了市场格局、用户体验转型以及开发优先级三大方面。

市场格局的变化可谓日新月异，因为生成式 AI 正在普及软件的开发与部署，大幅降低了新竞争者的进入门槛。借助 AI 驱动的开发工具和开箱即用的组件，复杂应用的开发速度比以往快得多。这种加速使得快速创新成为必须，而非仅仅是一种选择。虽然成熟的 SaaS 服务提供商目前凭借其积累的数据和深厚的领域知识享有优势，但当 Agent 开始提供差异化的价值时，这些所谓的“护城河”可能很快就会变得不堪一击。

用户体验转型则代表了用户与软件交互方式的根本性变革。传统的网页界面正逐步演变为对话式、Agent 驱动的体验，用户只需用自然语言表达意图，Agent 就能将其转化为具体行动。这种转变为前所未有的个性化创造了条件，Agent 能从用户行为中学习，并据此调整自身的响应和推荐内容。而其中的一大机遇在于跨产品集成：Agent 能够跨越多个工具和系统来编排工作流，从而创造出以往无法实现或需要大量人工介入才能完成的无缝体验。

您需要仔细权衡**开发优先级**，因为产品团队必须在各种相互冲突的需求之间找到平衡点。您的企业需要持续投入资源维持客户期望的核心产品能力，同时引入 Agent 功能，以提升或变革现有能力。这种两手都要抓的重点不仅体现在功能开发上，还包括管理技术债务、确保 AI 赋能环境下的安全与合规，以及在 Agent 交互日趋复杂时维持系统的大规模性能。成功的关键在于明确的优先级划分，确保在满足客户当前需求与实现长期战略目标之间，取得恰当的平衡。

实施框架

成功实施 Agent 产品战略，需要一个能够平衡机遇与执行能力的结构化框架。这一框架能为软件或技术公司提供一幅路线图，帮助评估、排序、执行并持续优化相关的 Agent 计划。遵循此框架，企业既能为 Agent 计划制定清晰的产品战略，也能保持在瞬息万变的市场中获胜所必需的战术灵活性。

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

评估阶段始于对现有产品组合的全面审查，旨在识别出 Agent 能力能发挥显著影响力的领域。在 Agent 的思维模式下，重要的是要着眼于构建 Agent 系统所带来的更广泛的可能性。团队需要思考 Agent 将如何影响客户体验、产品发现、用户引导、日常运营等一系列环节。Agent 还有可能重塑您现有的自动化流程、交互模式以及与第三方的集成方式。技术就绪度评估同样至关重要，它为产品的数据可用性、安全性及基础设施可扩展性等方面都带来了新的考量维度。这一阶段还应充分利用并明确现有优势，立足于当前的数据壁垒和核心竞争力。

优先级划分阶段的重点，是筛选出那些能够在价值创造、实施风险和资源限制之间达到理想平衡的项目。在 Agent 的背景下，团队需要权衡一系列新因素，这些因素都可能影响到 Agent 战略的优先级。在规划您的 Agent 转型之旅时，风险、价值、机遇、技术就绪度等方面可能都需要采用新的思维方式。极具吸引力的机遇，通常是那些兼具高潜在影响与可控的技术及运营风险的项目。尽管“短平快”的项目对于建立势头、验证价值至关重要，但仍需与 Agent 愿景中的长期战略目标相协调。新兴 Agent 产品的涌现、争夺市场关注度的需要，以及您所在领域的行业趋势，都可能影响您向 Agent 型解决方案转型的紧迫性与进程。优先级的划分应始终以客户价值为中心，确保所选项目能带来可衡量的成果和客户的成功。

执行阶段同样会因 Agent 的引入而发生改变。设计、构建、部署和运营 Agent 的整个过程，都将为企业的执行模式带来新的挑战。当团队将功能迁移至 Agent 时，必须考虑如何在充分发挥 Agent 潜力的同时，确保 Agent 产品仍然满足客户及所在行业对合规、准确、规模和安全性的严格要求。在 Agent 环境中，用于管理执行的生命周期、流程和工具也可能随之改变，这要求团队采纳新方法、新思维和新战略，这可能会重塑团队的协作方式。Agent 的引入可能还需要对现有的衡量框架进行革新，以便更好地追踪技术指标与业务成果。团队的运营模式也可能受到 Agent 的影响，从而改变团队追踪、分析和管理 Agent 环境的方式。

迭代是最后一个，且将持续进行的阶段，它能够确保项目的长期成功与可持续性。这包括持续监控竞争格局，紧跟 AI 与 Agent 技术的新进展，并为客户不断演变的期望做好准备。企业应当预见到，在快速变化的 Agent 领域，迭代和演进将变得更加频繁。定期重新评估产品组合与市场地位，有助于发现新的 Agent 机遇和潜在威胁，从而让您能够前瞻性地调整 Agentic SaaS 战略。

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

3. 多租户 Agent 架构

向 Agentic SaaS 转型将引入一系列全新的架构考量因素。这一点在 Agent 以多租户模式运行时尤为突出，因为租户身份会直接影响到 SaaS 系统中每个 Agent 的设计、安全、部署及运营。

正是在多租户 Agent 的内部机制中，Agentic SaaS 服务提供商能够利用租户上下文，为每个租户交付量身定制的体验。这种上下文可被用来在单个 Agent 的范畴内，为特定租户创造精准的价值。事实上，这正是多租户 Agent 体验引人注目的一个方面，它为实现差异化和创新开辟了全新的路径。本质上，这是将传统 SaaS 的核心能力与理念，直接嵌入到 Agent 的设计之中，从而将 SaaS 的高效率与 Agent 的高智能完美结合。

重点在于，许多构成多租户体系的核心概念，都可以直接映射到以“Agent 即服务”模式运行的 Agent 之上。租户隔离、用户引导、数据分区、服务分层、身份认证、用量统计——所有这些基础的多租户架构策略，都可以在您 Agent 的底层设计中扮演重要角色。

身份认证与租户上下文

在 Agentic SaaS 环境中，身份认证与租户上下文将具有更为重要的意义，尤其是在考虑到 Agent 可能采用的各种租户入驻、部署及分布式集成模式时。为了更具体地阐述这一点，我们来看图 5，该图从概念层面展示了租户身份如何影响 Agent 实现的核心要素。

图 5：向 Agent 注入租户上下文

图左侧是两个不同的租户，他们都在使用同一个 Agent。此处的前提是，租户的身份已经过验证，并被映射到一个特定的上下文中。这个租户上下文会流入 Agent 内部，并被一致地应用于其所有内部资源，以确保对该租户的请求进行适当的处理。这一简化视图清晰表明，Agent 及其依赖的资源，正是利用租户上下文来驱动其智能决策与最终产出。Agent 的日志、计费、计量、路由等其他功能模块，同样会利用这一上下文信息。

在多 Agent 系统中，当上下文需要在多个独立的 Agent 之间流转时，问题会变得更加复杂，尤其是在某些 Agent 由第三方创建和运营的情况下。图 6 展示了一个 Agent 系统的示例，其中认证与授权逻辑分布在一系列相互集成的 Agent 之中。

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

图 6：租户上下文与分布式系统

在这样的系统中，每个 Agent 的服务提供商都必须建立一套认证和授权机制，以应对在 Agent 系统中运营所带来的独特的安全、合规及授权模型挑战。在一个由客户或其他 Agent 动态组合而成的 Agent 系统中，这一挑战尤为严峻。每个新加入该系统的 Agent，都必须将其自身的身份和租户上下文，与贯穿整个 Agent 体验的交互流程关联起来。图 6 重点展示了在 Agent 之间 (A2A) 的交互中可能出现的一些身份和租户上下文问题，因为不同的 JSON Web Token (JWT) 会在 Agent 之间传递。

构建能够感知租户的 Agent

如前所述，租户上下文能够帮助我们创建“可感知租户”的 Agent，这类 Agent 可以利用该上下文来驱动针对特定租户的产出与体验。在实际部署多租户 Agent 时，这具体意味着什么呢？图 7 概括性地展示了一个 Agent 的构成元素，并通过列举多种可被编排以实现目标的工具和机制，概览了 Agent 在实现目标时的多种可能性。

图 7：Agent 的构成元素

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

在图左侧，您会发现该 Agent 依赖于多种组件，包括用于存储信息的“记忆”、用于定义行为策略的“护栏”，以及可用于执行特定任务的“工作流”。有些人或许会认为工作流不应出现在这里，但在某些场景下，这些预定义的工作流能在 Agent 体验中发挥重要作用。

在图右侧，是可作为 Agent 流程输入的“知识”与“工具”，它们能提供额外的洞察和上下文，以增强 Agent 处理请求的能力。“行动”也在此列出，代表了 Agent 可以调用的各种机制。

图底部则代表了可能与当前 Agent 集成的其他各类 Agent。这些 Agent 既可以是内部的，也可以是为实现目标而协同工作的第三方 Agent。

在此背景下，让我们来探讨将多租户机制引入 Agent 的设计中，究竟意味着什么。租户的存在要求我们必须思考 Agent 应在何处、以何种方式引入相应策略与机制，从而针对不同租户提供定制化的行为和行动。这为我们思考 Agent 的方式增添了一个新维度，其知识、学习、工具和记忆等能力均需扩展，方能支持“可感知租户”（或至少受租户影响）的个性化体验。

让我们回顾图 7 中展示的通用 Agent 模型，并思考需要做出哪些改变才能使其支持多租户。现在，在图 8 中，您可以看到租户概念是如何改变该 Agent 的设计与实现的。

图 8：一个可感知租户的 Agent

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

租户概念的引入，对整个 Agent 产生了连锁反应，最终改变了其周边组件的性质。例如，在图左侧，Agent“记忆”的形式已被修改，以增加对租户专属记忆的支持。图右侧和上方同样如此，也增加了对租户专属知识、工具和护栏的支持。

这个例子所代表的应用场景可能比较极端，因为并非多租户 Agent 的每个方面都需要为每个租户提供专属资源。但重点在于，Agent 的实现范围、资源配置和内在性质，都应充分考虑租户的特殊性，思考在哪些环节为租户提供定制化能为 Agent 的能力带来更大的影响、价值和上下文。秉持这种思维模式，您就能创建出能够针对其所支持的特定租户角色进行学习、演进并处理任务的 Agent。

关键在于，租户上下文能够直接影响 Agent 的构建方式。同时，它也塑造了租户与外部实体（包括其他 Agent）的交互模式。同样重要的是，这也带来了构建任何多租户解决方案时都必须面对的传统考量因素。“吵闹的邻居”问题、租户隔离、服务分层、请求限流、成本分摊等一系列基础的多租户概念，都将成为您 Agent 设计与架构中不可或缺的一部分。后文将对这些概念进行阐述。

Agentic SaaS 环境下的 SaaS 控制平面

在多租户架构的最佳实践中，系统通常被划分为两个独立的部分：控制平面和应用平面。控制平面作为一站式界面，提供跨环境中所有租户的运营、管理与编排功能。应用平面则是解决方案的业务逻辑、产品特性及核心功能所在的层面。

同样的职责划分原则，也可应用于 Agentic SaaS 领域。多租户 Agent 同样需要某种程度的集中化管理、运营与洞察，因此，继续沿用控制平面模型来满足这些需求是合乎逻辑的。图 9 以简化的方式，展示了在 Agentic SaaS 环境中如何对这两个平面进行划分。

图 9：控制平面与应用平面

此图描绘了在 Agentic SaaS 环境中，使用控制平面与应用平面的两种不同模式。图左侧的示例代表了一种“产品化 Agent”的 SaaS 体验，其中 Agent 被公开销售、消费和变现（可能作为某个 Agent 系统的一部分）。在此模式下，控制平面负责集中管理隶属于同一 Agent 服务提供商的每一个产品化 Agent。

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

图右侧则是“嵌入式 Agent”模型，其中 Agent 内嵌于 SaaS 应用程序之中。实际上，该示例还展示了嵌入式 Agent 与现有服务并存运行的场景。对于嵌入式模型，控制平面的角色依然保持不变，但其管辖范围更广，既要管理 Agent，也要管理应用平面中运行的其他所有租户资源。

对于只拥有单个 Agent 的组织而言，控制平面是其生命周期所有编排与管理任务的枢纽。租户入驻、租户管理、计费、指标收集、Agent 洞察、Agent 准确性评估等功能，都由控制平面提供。这为 Agent 服务提供商带来了敏捷、高效和可复用的能力。随着更多 Agent 的引入，它还能有效扩展，为所有 Agent 实现集中化的部署、运营与管理。这些价值对产品化模型和嵌入式模型同样适用。不过，在嵌入式模型中，控制平面的范围与职责将更为复杂。

尽管模型中出现了新的复杂性，但其基本原理并未发生根本性改变。控制平面仍将提供相同的核心能力，包括编排 Agent 配置、提供租户与 Agent 的可观测性、为计费目的收集消耗与计量数据，以及管理租户策略。

值得注意的是，Agent 的应用并不局限于应用平面。控制平面服务的实现与编排，同样可以由 Agent 来支持。试想一下，我们可以引入一组 Agent 来管理新租户的入驻流程。在此场景下，Agent 能为新租户环境的部署和配置带来更高的智能性、敏捷性与自动化水平。指标与分析功能也可以由 Agent 支持，由它们来聚合数据、洞察趋势并依据策略采取行动。最终的目标是让我们的思维超越应用程序本身的业务逻辑，将 Agent 的应用范围延伸至 SaaS 环境的运营层面。

强制执行租户隔离

租户隔离是任何多租户环境都需遵循的核心原则。它指的是用一系列策略与机制，来确保一个租户无法访问另一个租户的资源。对于多租户 Agent，您需要在多个层面引入相应的构造与机制，以强制执行其租户隔离要求。

应用租户隔离的模式，与传统多租户系统中所采用的策略大同小异。通常，在构建 SaaS 架构时，您需要识别出系统中任何可能跨越租户边界的资源访问请求点。例如，一个微服务可能依赖于为每个租户专设的 Amazon DynamoDB 数据表。这就要求必须引入策略，以确保租户 1 无法访问租户 2 的数据表。

现在，让我们从 Agent 的视角，审视其与各租户专属资源交互时的租户隔离问题。图 10 简要展示了 Agent 如何应用租户隔离策略，来控制对这些租户资源的访问。

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

图 10：隔离租户资源

在此示例中，Agent 为每个租户提供的专属知识，分别存储在独立的向量数据库中（如右侧所示）。当 Agent 处理请求时，它会检查租户上下文，并应用相应的租户隔离策略，以确保租户无法跨越其访问边界。

如果您的 Agent 架构中采用了模型上下文协议 (MCP)，那么它可以被用来实现您的租户隔离模型。图 11 的示例展示了 MCP 为何非常适合用于引入并应用隔离策略。

图 11：使用 MCP 实现租户隔离

MCP 能够为 Agent 提供一套标准化协议，用来简化与 Agent 自身需要访问的任何工具、数据或资源的集成过程。在此示例中，系统通过一个 MCP 客户端和一个 MCP 服务器，来与图右侧的租户专属知识及工具进行交互。租户上下文从客户端传递至服务器，服务器利用该上下文从亚马逊云科技身份与访问管理 (IAM) 服务中，获取该租户专属范围的凭据。随后，服务器将使用这些凭据来访问租户资源。这一机制能够确保任何一个租户都无法访问属于其他租户的资源。

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

关键在于，当您的 Agent 采用多租户模型时，就必须同步引入相应的机制，来保护和隔离每个租户的资源。您可以利用 IAM 或其他工具及框架来定义并应用您的租户隔离策略，确保访问权限被严格限制在该租户允许的资源范围之内。

4. SaaS Agent 运维 (AgentOps)

在构建 Agentic SaaS 环境时，关键在于深入思考如何运营和管理所有这些 Agent（即 AgentOps）。作为 Agent 服务提供商，您需要通过指标、数据、洞见和日志，来持续监控 Agent 的运行状况、扩缩规模及活动。在多租户 Agent 环境中，这一需求更为迫切，因为您需要准确了解每个独立租户如何消耗您的 Agent 资源。

在多 Agent 协作的场景下，洞察 Agent 之间的交互情况则更具战略意义。能够剖析和追踪 Agent 间的活动，对于排查那些影响多 Agent 系统规模、准确性及效能的问题来说，是必不可少的。

当多 Agent 系统由来自不同服务提供商的 Agent 构成时，其复杂性会呈指数级增长。不同的服务提供商可能会采用截然不同的运营策略、身份模型等。如果您的 Agent 需要依赖第三方集成，那么您必须设计一套能够兼容所有这些差异的运营体验。您需要尽可能地设法解决这些分布式依赖带来的问题，以确保提供一致的用户体验。

运营团队最好能对 LLM 的交互进行剖析，以便更清晰地了解每个 Agent 给 LLM 带来的负载情况。这些数据对于优化 Agent 的部署至关重要。它还能让您的团队洞察 Agent 和租户身份如何影响系统整体成本，特别是能够捕获到租户的工作负载对 LLM 成本的具体影响数据。

定义 Agent 的新一代 KPI

Agentic AI 旨在作为自主的智能体运行，通过不断学习来驱动正确的结果，并持续自我演进。作为一家 Agentic SaaS 企业，您需要在很多方面将相当一部分的业务价值托付给这些 Agent。这意味着一份重大的责任。如果再叠加一层需求——即需要根据租户专属的记忆、访问权限和数据来实现定制化的结果——这项任务就更具挑战性了。

因此，了解这些 Agent 是否如预期般运行，将对您的 Agentic SaaS 业务能否持续成功起到决定性作用。这就要求您必须投入资源，去发掘并持续追踪那些能够揭示 Agent 行为与演进规律的关键指标。

Agentic SaaS 服务提供商应专注于定义一套指标体系，以获取关于其 Agent 行为的关键业务、成果及技术方面的信息。Agent 的扩展性如何？它在驱动与租户上下文相关的结果方面，成功率有多高？它达成这些结果的效率如何？以上这些，只是您需要整合到运营模型中的众多洞见之一。在探索这条路径时，您不妨也思考一下，如何以及在何处利用生成式 AI 工具，来丰富这些信息的收集与分析过程，从而帮助您加速创造并彰显价值。

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

5. 业务战略

与所有 SaaS 业务一样，向 Agentic SaaS 的转型之旅，始于清晰的业务愿景。当您开启 Agent 转型之路时，必须首先对自身的 Agent 战略有一个清晰的定位。这一切都始于对架构、Agent 协议和运营模型进行定义之前。事实上，Agent 带来了一系列全新的考量因素，这些因素将深刻影响 Agent 模型塑造企业增长与创新的方式。本节将探讨在引入 Agent 后，三个受到深远影响的基础策略领域，即产品策略、定价策略和市场进入策略。

市场范围规划策略

对于每家 Agentic SaaS 服务提供商而言，影响其市场战略的市场格局与商业现状都将截然不同。您所处的行业性质、面临的竞争压力、客户的期望以及新兴的市场趋势，都将在某种程度上塑造您的 Agent 市场战略。在 Agentic SaaS 这个新兴领域，这一点尤为突出，因为整个行业格局仍在演变，各大组织也仍在探索 Agent 赋能解决方案的真正潜力。团队必须投入时间来剖析自身所处的市场，并制定一个能够指引其迈向成功的 Agentic SaaS 愿景。以下是团队在制定市场战略时应重点关注的几个关键领域。

Agent 机遇评估

评估您的 Agent 机遇的规模与潜在影响，是至关重要的一步。对一些企业而言，向 Agent 转型将是颠覆性的。这一转型能开辟全新的细分市场和增长机遇，从而拓展业务的覆盖面与影响力。

而对另一些企业来说，这种转变则可能更为温和，主要体现在丰富解决方案的功能和提升产品价值上，并不会对整体商业模式造成根本性的改变。

关键在于，要明确您期望通过 Agentic SaaS 实现的目标，而非局限于对生成式 AI 和 Agent 的追捧；要更多地着眼于数据与市场现实，以判断向 Agent 模型的转型是否能真正带来实质性的商业价值、市场拓展和行业影响力。团队必须评估宏观市场格局，并建立一个能将 Agent 的潜力与自身解决方案的现实相结合的分析模型。

作为这项工作的一部分，团队应预估其 Agent 模型可能带来的影响。一个由 Agent 赋能的体验，能否为业务开辟新的增长与加速路径？是否有明确迹象表明，您所在的行业及客户能够从 Agent 解决方案中切实获益？您是否面临被新兴或已有的 Agent 产品颠覆的风险？

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

Agent 市场与细分市场趋势

作为市场战略的一部分，您必须评估市场中正在显现的 Agent 趋势。您的 Agent 解决方案，是会作为独立产品销售，还是会嵌入到现有产品中？又或者，它将成为一个更宏大的领域生态系统的一部分，需要与其他 Agent 协作来创造价值？这两种情况比较极端，但它们都凸显了在您的领域或行业内理解如何触达客户的重要性。

您在这一方面做出的选择，将对您的 Agentic SaaS 产品的形态乃至成败，产生深远影响。您必须明确您的 Agent 体验应如何实现价值与影响力的最大化，并认识到，通往市场和客户的路径将因您的 Agentic SaaS 体验的性质而异。由于目前在构建成功的 Agent 解决方案方面，既缺乏成熟的经验也缺少足够的数据。因此企业必须深思熟虑，思考如何让 Agent 真正落地并与目标市场及细分客户产生共鸣。

定价策略

传统上，SaaS 服务提供商主要依赖几种定价策略：基于用量、基于价值，或二者的某种混合模式。订阅费用、坐席数、用户量、功能模块、吞吐量——这些都曾是 SaaS 服务提供商的定价手段。这些模型在过去基本满足了 SaaS 业务的需求，为供应商和客户双方都提供了相对可预测的成本结构。

Agent 正在从根本上推动 SaaS 定价模式的转变，使其从传统的基于坐席的收费模式，转向基于用量乃至基于结果的收费模式。Agentic AI 之所以能推动这一演变，是因为它能够自动化任务、提供个性化体验，并能在几乎无需人工干预的情况下交付特定业务成果。从根本上说，向 Agent 模式的转型，可能会极大地改变您的企业制定定价策略的方式。

一些具有前瞻性的 SaaS 服务提供商，已经开始采纳基于价值的定价结构，将收费标准与客户从其产品中获得的实际收益直接挂钩。更具体地说，这种价值可以在 Agent 实现的业务成果和效率提升上得到直接体现。

Agentic SaaS 模型下的定价考量

新的定价考量因素正在涌现，为传统的基于价值和成本的定价模式带来了全新的维度。有了 Agent，用量、功能和价值的定义都可以由 Agent 承担 。是 Agent 在调用资源、评估选项、判断需求并最终驱动业务成果。从这个角度看，Agent 成为了为客户实现价值的直接载体。

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

尽管资源消耗量可能仍是定价的核心因素，但在 Agentic SaaS 模型下，消耗的分配方式与行为模式可能会大不相同。系统固然仍会计量服务器、存储、带宽等基础资源的消耗，但 Agent 的定价模型还必须能够计量更复杂的单位，例如处理的 token 数量、Agent 运行时长、完成的任务数或关键决策点。Agentic SaaS 解决方案可以基于 Agent 的能力与自主性水平提供分级定价，对于那些只需更少人工监督就能处理更复杂任务的 Agent，可以设置更高的价格。

最近，一个名为“自动化价值定价”的新概念应运而生，与自动化任务所取得的商业价值成正比，而不仅仅是与所消耗的计算资源挂钩。这代表着一次重大的转变，它奖励的是准确性、自主性与智能，是用 Agent 驱动的效率、价值和准确性，来取代单纯的资源消耗定价。

多 Agent 系统则为 Agentic SaaS 环境引入了全新的定价维度。当 Agent 以分布式模式部署时，您的定价就必须考虑额外的复杂性。如果您的多 Agent 系统中还包含了第三方 Agent，情况会变得更加复杂。这意味着，您的客户最终支付的价格，可能取决于系统中所有 Agent 成本的总和。这可能会变得相当棘手。试想一下，在多 Agent 场景下，向基于结果和价值的定价模式转型会如何展开：价值是源自整个系统的协同产出吗？还是应将价值归功于某个独立的 Agent？这些 Agent 应如何变现？随着企业开始制定各自的定价策略，这一领域亟待更深入的探索。

此处概述的种种情况清楚地表明，Agentic SaaS 的定价格局正在经历快速演变。随着 Agent 市场对于供需双方都日渐成熟，成本、价值、成果与定价策略之间的映射关系，可能会逐步走向标准化。在此过渡期间，Agentic SaaS 服务提供商将在其定价策略中面临一种微妙的平衡。如果定价过高，用户采纳率将受限。但如果定价过低，或基于错误的维度定价，则无法实现盈利。在定价转型期间，客户流失是另一个不容忽视的风险。您创建的定价模型既要足够灵活，能够适应多变的市场条件，又要足够稳定，从而维持客户信任和可预测性。

此外，定价策略还可能在很大程度上受到您自身 Agentic SaaS 转型路径性质的影响。如果您正在对现有产品进行扩展、嵌入或向 Agent 体验迁移，您就需要找到一种对客户而言合情合理的过渡方案。清晰的沟通策略可以使这一过程更加顺畅。在进行重大定价变更时，您也可以考虑为老客户保留原有条款或推迟新政实施，以帮助在过渡期内维系客户关系。

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

新增成本因素

衡量并归因成本，对于了解租户如何给您的环境带来负载、消耗资源至关重要，这最终会影响到您业务的利润率和净收益。在 SaaS 环境中引入 Agent，会创造出全新的活动与消耗模式，这有可能改变您解决方案的整体成本结构。新旧因素相结合的成本，既包括传统的基础设施成本，也涵盖了因使用 AI 模型而产生的额外成本。

推理成本自然是首要考虑的因素。作为 Agentic SaaS 服务提供商，您必须考虑与 token 相关的推理成本（包括输入和输出），这些成本会影响您环境的整体成本状况。同时，还必须了解这项成本在不同租户之间可能存在的差异，并可能需要通过服务分层策略来控制模型的消耗量。这些因素还可能促使您在批处理（成本更低、延迟更高）与实时操作（价格更高、响应即时）之间做出抉择，从而在设计 Agentic SaaS 服务时实现成本与效益的平衡。

Agent 的消耗要高于单纯的 AI 模型。它们通常还依赖于其他用于丰富 Agent 体验的基础设施。这可能包括用于保留上下文的内存和存储开销、提示词缓存机制，以及 Agent 执行所需的计算资源。此外，还可能包括使用专门的亚马逊云科技的 AI 计算实例（例如 Trainium、Inferentia 芯片或用于 AI 工作负载的 GPU）的成本。此外，您可能还需要考虑模型上下文协议 (MCP) 服务器、向量数据库以及集成中间件的潜在成本。以上仅是构成 Agent 成本并可能影响 Agentic SaaS 产品定价模型的部分资源示例。

运营成本会显著影响 Agentic SaaS 系统的经济效益。随着您将更多智能赋予 Agent，您必须评估这些 Agent 将如何融入解决方案的整体运营体系。除了监控核心功能外，您的 Agent 可能还需要额外的可观测性支持。Agent 应提供以下可视化数据：Agent 的性能、响应的可解释性、结果的准确性及合规性。这些数据应包含租户上下文，以便深入了解单个租户如何影响您环境的整体成本结构。同样的运营数据，还可用于制定战略性的模型选择方案，从而创造优化模型成本的机会。

多 Agent 系统则引入了单个 Agent 实现范畴之外的额外成本维度，包括：协调 Agent 活动的编排开销、Agent 间的通信成本、用于在 Agent 间维护上下文的共享内存系统、维护专门的 Agent 池，以及更复杂的监控需求。

通过使用一组协同工作的小型、专用 Agent，而不是依赖单一的大型模型，或许能够提升性能并降低成本。这一点对于那些需要不同类型专业知识或特定领域知识的复杂任务而言，尤其有效。多 Agent 系统的经济可行性，取决于能否在这些基础设施成本与通过 Agent 专业化所获得的性能优势之间，取得理想的平衡。

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

Agentic SaaS 定价实施框架

成功实施 Agentic SaaS 定价，需要可靠的衡量体系、优化策略及清晰的客户互动方案。该框架主要在以下几个关键领域提供指导：监控与衡量、成本优化及验证方法。

监控与衡量

有效的衡量与监控，是任何可持续的 Agentic SaaS 定价方案的基石。建立实时的、精细到每个租户的 token 消耗仪表盘，提供跨 Agent、功能和客户的细粒度可见性，有助于您将成本控制在合理范围内。预测性分析有助于预判使用模式和潜在的成本超支，而异常检测系统则能在出现意外的使用高峰或低效的 Agent 行为时，及时向相关方发出警报。此外，带有可配置阈值和自动熔断机制的 token 消耗护栏，也能有效帮助控制成本。

为识别优化机会，您可以追踪任务达成率和 token 效率指标（即输入与输出的比率），并为 Agent 任务建立一套具有明确成功标准的、可靠的验证机制。这套系统应包含自动化质量保证检查、由人工验证样本来优化自动化系统的反馈循环，以及对任务达成率与客户满意度指标之间的关联性追踪。

客户互动

客户教育与用户采纳策略，必须兼顾免费套餐的设计与持续的用户互动两大方面。免费或入门级套餐的设计，应旨在控制成本的同时，充分展示产品价值。您可以考虑根据交付的价值，而非 token 消耗量，来设置使用限制。此外，基于已验证的投资回报率 (ROI) 设定清晰的升级路径、提供引导式的上手体验，以及确保消耗模式的透明可见性，都有助于促进客户的成功增长。持续的客户教育和有效的工具，能让客户清晰了解其所获价值与所付成本之间的关系。通过使用情况仪表盘和优化建议等途径，阐释 Agentic SaaS 解决方案的经济效益，帮助客户真正理解其价值所在。此外，专注于价值最大化而非消耗量的客户成功计划，也能起到积极作用。最终目标是，为不同的应用场景建立清晰的投资回报率 (ROI) 衡量框架，以证明业务影响，并为客户的持续投入提供充分理由。

成本优化

成本优化必须同时兼顾 token 效率与架构设计两大方面。您可以采用附带标准化模板的提示词工程最佳实践，并部署上下文压缩算法，以减少 token 消耗。为降低负载，您可以为重复性查询启用缓存机制，并为时效性不强的工作负载采用批处理策略。在多租户环境中，如果提示词并非为单个租户专门定制，这种做法将尤为有效。系统还可以采用“最低能力适用模型”的选择逻辑和渐进增强模式，先由更简单、资源消耗更少的模型进行初步处理。这些混合方法可以将检索增强生成 (RAG) 与直接推理相结合，以提升效率。使用带有优化提示词和小型模型的专用 Agent，并为每个工作流设置 token 预算，便能构成一套全面的成本管理方案。

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

验证方法

“按达成付费”的 SaaS 定价模型，需要有可靠的验证方法作为支撑，包括自动化检查和审计追踪，并可能因任务类型而异。这类模型适用于那些产出清晰、可衡量的服务，例如文档处理或内容创作。但对于连续性服务或复杂的战略性工作，这种模型则较难应付。Agentic AI 的引入，既带来了机遇，也带来了挑战。虽然您能够创造出更精细复杂的产品，但您也可能面临比传统 SaaS 软件更显著的运营成本增幅。这些成本使您必须仔细思考如何设计所有的服务层级，尤其是那些 AI 功能受限的免费层级，并建立能清晰展示递增价值的升级路径。

定价过渡策略

定价模型的过渡策略，应力求在业务可持续性与客户价值感知之间取得平衡。最佳实践是让分级定价与交付的价值对齐，而非与 token 消耗量挂钩，并在用户达到用量上限时，提供平滑的服务降级路径（例如从 Amazon Nova Premier 版切换至 Amazon Nova Lite 版）。您可以通过实施“软上限”（在自动升降级前发出通知）、提供客户可控的月度预算节流选项，以及展示清晰投资回报率的升级路径，来创造更友好的客户体验（甚至可能带来额外的收入机会）。“按用量付费”模式的实施，则需要复杂的 token 计量系统（具备实时可见性与控制能力）、“额度包”策略（支持自动续费和支出上限），以及与业务成果而非消耗量挂钩的价值定价法。将基础订阅与按用量付费相结合的混合模式，则可以激励用户形成更高效的 Agent 使用习惯。

定价要点总结

遵循此框架，您既能为自身的 Agentic SaaS 计划制定清晰的定价策略，又能保持必要的战术灵活性，以便在瞬息万变的市场中抓住新兴机遇。这种方法有助于您在商业盈利与客户价值之间找到平衡点，从而促进可持续增长以及高利润 Agentic SaaS 解决方案的持续采纳。

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

市场进入策略

向 Agentic SaaS 转型，需要您重新构想市场进入 (GTM) 策略的核心要素，其中涵盖产品定位、产品结构设计、功能定价、渠道利用（例如通过亚马逊云科技 Marketplace 分发和变现 Agent）以及价值沟通等方方面面。如前文产品战略部分所述，您的 GTM 策略会受到一个关键决策的影响：是专注于**市场扩张**，还是**现有市场增长**，又或是二者的结合。

任何 Agentic SaaS 产品的 GTM 模型，都将高度依赖于其 Agent 体验的具体性质。将一个独立的、产品化的 Agentic SaaS 解决方案推向市场，其 GTM 策略与推广一个完全采用嵌入式 Agent 的 SaaS 解决方案截然不同。产品化 Agent 的 GTM 策略必须考虑一系列新的用户发现、引导及获取模型；而嵌入式 Agent 的 GTM 策略则相对简单，更侧重于销售一个“由 Agent 赋能”的整体解决方案。两种模式都需要全新的 GTM 思维，但具体方法会因您提供的产品形态而异。

总而言之，GTM 策略是一个更为广泛的话题，超出了本白皮书的讨论范围。需要注意的关键是，团队需要思考 Agent 可能在何处、以何种方式影响企业的整体 GTM 布局。

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

总结

Agentic AI 正在改变 SaaS 服务提供商构建系统的方式。它有潜力重塑 SaaS 领域的格局，深刻影响系统的构建、运营、变现及交付模式。

在许多情况下，Agent 将从根本上重塑客户体验，改变客户消费产品以及从中获取价值的方式。对许多企业而言，这将是一次颠覆性的变革，要求您必须在一个技术与商业模式仍在飞速演进的领域中，迅速采取行动。

在 SaaS 环境中引入 Agent，远非创建几个 Agent 并将其置入您的产品体验那么简单。您的团队需要仔细评估，Agent 应在何处、以何种方式才能契合市场需求，从而交付出能够充分利用 Agent 潜能、具有差异化价值的产品。

这也意味着，您需要思考引入这些 Agent 将对您 SaaS 业务的运营、规模、效率、敏捷性及创新能力带来何种影响。您的团队必须为其 Agent 愿景增添更多思考维度，以明确构建多租户 Agent 究竟意味着什么——既要增加价值，又要实现规模经济与高效率这些业务的根本目标。

本白皮书旨在揭示与新兴 Agentic SaaS 体验相关的一系列关键考量，重点阐述那些可能影响您决策的关键业务、战略及技术因素。核心要点在于，Agentic SaaS 要求团队必须经过深思熟虑，明确 Agent 将在何处、以何种方式为您的业务和客户带来显著影响，从而让您能够成功引入 Agent，并开始塑造企业未来的发展轨迹与增长路径。如需了解更多信息，请访问 https://aws.amazon.com/ai/agentic-ai/。

声明

客户须对本文档中的信息自行进行独立评估。本文档：(a) 仅作信息参考之用；(b) 代表当前的亚马逊云科技产品和实践，如有更改，恕不另行通知；(c) 不构成亚马逊云科技及其附属公司、供应商或许可方的任何承诺或保证。文中涉及的亚马逊云科技产品或服务均“按原样”提供，不附带任何形式的明示或暗示的保证、声明或条件。亚马逊云科技对客户的责任和义务受双方协议约束，本文档与亚马逊云科技和客户之间签订的任何协议无关，亦不影响任何此类协议。

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

贡献者

对本文档做出贡献的人员包括：

Tod Golding，亚马逊云科技高级首席解决方案架构师

Mahdi Sajjadpour，亚马逊云科技解决方案架构部总监

Wallace Printz，亚马逊云科技首席解决方案架构师

Erik Anderson，亚马逊云科技首席解决方案架构师

BK Chaurasiya，亚马逊云科技解决方案架构部高级经理

Gaurav Acharya，亚马逊云科技首席解决方案架构师

Sindhura Palakodety，亚马逊云科技高级解决方案架构师

Rielah DeJesus，亚马逊云科技首席解决方案架构师

Lon Miller，亚马逊云科技高级解决方案架构师

© 2025, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

亚马逊云科技 FOR SOFTWARE AND TECHNOLOGY

扫码获取白皮书电子版

*前述特定亚马逊云科技生成式人工智能相关的服务目前在亚马逊云科技海外区域可用。
亚马逊云科技中国区域相关云服务由西云数据和光环新网运营，具体信息以中国区域官网为准。