



分阶段实现云端应用程序现代化的方法 AWS

AWS 规范性指导



AWS 规范性指导: 分阶段实现云端应用程序现代化的方法 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
目标业务成果	1
现代化进程	3
步骤 1：评估您的应用程序	3
使用现代化诊断手册	3
识别指标	5
步骤 2：从小处着手，建立势头	6
验证优先级驱动因素	6
敲定细节	6
构建基础平台服务和实现应用程序现代化	7
步骤 3：创建可扩展的现代化路线图	7
现代化准备情况因素	9
代码	9
构建和测试	9
版本	9
操作	10
优化	10
准备就绪	10
后续步骤	11
资源	12
相关指南	12
AWS 资源	12
文档历史记录	13
.....	xiv

分阶段实现云端应用程序现代化的方法 AWS

Vijay Thumma 和 Ashish Ameta , Amazon Web Services (AWS)

2023 年 5 月 ([文档历史记录](#))

现代化需要采用多维方法来采用和使用新技术，更快地交付产品组合、应用程序和基础设施价值，并使组织能够以最优的价格进行扩展。它涉及优化、维护应用程序，并在不中断的情况下在现代化模型中运行，并要求您简化业务运营、架构和整体工程实践。

现代化不仅仅是应用程序；它需要一个能够提供安全灵活的操作框架的现代化基础架构。在业务流程质量、可用性和敏捷性方面，应用程序和基础设施是不可分割的。在不考虑基础架构的情况下实现应用程序现代化会导致高昂的总体成本，并对性能和质量产生负面影响。现代应用程序是结合新的架构模式、运营模型和软件交付流程构建的。它们可以从零扩展到数百万用户，管理 TB (如果不是 PB 级) 的数据，可在全球范围内使用，并在毫秒内做出响应。当您对在 Amazon Web Services (AWS) 云中管理的工作负载组合进行现代化改造时，您可以使用容器、无服务器技术、专用数据存储和软件自动化来更换平台、重构或替换这些工作负载，从而获得 AWS 所提供的最大灵活性和总成本优化 (TCO) 优势。

本指南适用于应用程序所有者、企业主、架构师、技术主管和项目经理。指南中探讨了如何为现代化评估阶段中选定的应用程序开发基础能力，以及如何通过分阶段方法来加速现代化工作。

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

- [实现云端应用程序现代化的策略 AWS](#)
- [评估 AWS 云端应用程序的现代化准备情况](#)
- [将单体分解为微服务](#)
- [使用 AWS 无服务器服务集成微服务](#)

目标业务成果

分阶段实现应用程序现代化的方法预期会产生以下结果：

- 通过使用该 build-and-prove 方法和云原生架构（例如微服务），组织能力和加快创新的能力。
- 一种变更管理和运营模式，通过培训和工具改进来建立组织准备状态。
- 团队方法有助于在短短 12 周内取得初步结果，提供体验式学习，并实现独立、持久的客户成功。
- 基于微服务、APIs 可重用组件和容器化的可组合应用程序架构。

- 适用于特定战略应用程序的可扩展现代化路线图，其中包括在split-and-seed模型中工作的规范性指导。在此模型中，现代化能力和服务跨多个专注于业务成果的工程团队进行扩展。在定义新的最低可行产品 (MVPs) 时，最初的团队成员分手创建新的产品团队。

现代化进程

分阶段实现现代化的目标是通过使用现代化维度提供增量价值，将其应用于一部分核心的、与业务差异化的应用程序，并加快现代技术的采用。分阶段方法包括三个步骤：

1. 使用现代化诊断手册评估应用程序的成熟度。采用全面的方法进行采用，并制定与预期业务结果相一致的流程。
2. 从小处着手，为交付最小化可行产品 (MVP) 做好准备，通过尽早和渐进地推动业务业绩来获得动力。此阶段通常需要 12 到 16 周。
3. 创建可扩展的现代化路线图，在 split-and-seed 模式下工作。

以下各节将更详细讨论这些步骤。

主题

- [步骤 1：评估您的应用程序](#)
- [步骤 2：从小处着手，建立势头](#)
- [步骤 3：创建可扩展的现代化路线图](#)

步骤 1：评估您的应用程序

本阶段的目标是：

- 彻底了解您的应用程序格局，为现代数据平台做好应用程序准备，这样您就可以在不影响业务的情况下加快价值实现时间，然后进行现代化、优化和扩展。
- 概述您的应用程序格局，以确定与变更相关的效益、风险和成本。
- 提供一套端到端的服务：从战略和规划；到部署、迁移和应用程序现代化；再到持续支持。
- 制定政策、建议和控制措施，提供可重复使用的实践和工具，以实现持续的业务价值。

在评估阶段，应用程序所有者和架构师使用现代化诊断手册来验证其现代化目标和优先事项。

使用现代化诊断手册

现代化诊断手册提供了一个确定企业从当前状态转变为未来状态的价值的过程。这包括现代化所涉及的技术变革。

您可以使用诊断手册来确定云现代化的应用程序或应用程序套件的优先级，并确定现代化过程中需要解决的组件。

诊断维度

现代化诊断手册可帮助您了解应用程序或一组应用程序的当前状态和目标（迁移后）状态的以下方面：

- 应用程序分组 — 是否有理由对应用程序进行分组（例如，按技术或运营模式）进行现代化？
- 排序 — 根据依赖关系，应用程序进行现代化是否有先后顺序？
- 技术 — 技术类别有哪些（例如，中间件、数据库、消息收发）？
- 依赖关系 — 应用程序是否与其他系统或中间件有关键依赖关系？
- 环境 — 使用了多少开发、测试和生产环境？
- 存储 — 存储要求是什么（例如，测试数据的副本数量）？
- 运营模式 — 应用程序的所有组件能否采用持续集成和持续交付 (CI/CD) 管道？
 - 如果是，应将哪些基础设施职责分配给应用团队以及分配给谁？
 - 如果不是，运营团队还应承担哪些基础设施责任（例如修补）？
- 交付模式：
 - 根据应用程序或应用程序组，您应该更换平台、重构、重写还是替换？
 - 现代化的哪一部分应该使用云原生服务？
- 技能组合 — 需要什么专业知识？例如：
 - 云应用程序背景，用于从头开始使用容器和无服务器技术构建具有模块化架构的应用程序。
 - DevOps 通过使用开源以及 AWS 工具和服务，在 CI/CD 流程、基础设施即代码以及自动化或应用程序可观察性领域开发解决方案的专业知识。
- 现代化方法 — 考虑到应用程序的当前状态、云技术选择、当前的技术债务 CI/CD、监控、技能和运营模式，需要完成哪些技术迁移工作？
- 现代化时机 — 哪些业务组合的时机考虑因素或其他计划中的工作考虑因素可能会影响现代化时机？
- 基础设施的单位成本和总成本 — 根据经济分析，在内部维护工作负载的年度成本与在 AWS 本地维护工作负载的成本是多少？

根据这些维度评估应用程序可以帮助您在推动云现代化的同时，始终扎根于业务、技术和经济领域。

构建块

当您对应用程序进行现代化改造时，您可以将观察结果分为三个组成部分：业务敏捷性、组织敏捷性和工程效率。

- **业务敏捷性**：与企业内部将业务需求转化为需求的有效性有关的实践。交付组织对业务请求的响应程度，以及企业在向生产环境发布功能方面拥有的控制权。
- **组织敏捷性**：定义交付流程的实践。示例包括敏捷方法和 DevOps 仪式、角色分配和清晰度，以及整个组织的整体协作、沟通和赋能。
- **工程效率** — 与质量保证、测试、配置管理 CI/CD、应用程序设计和源代码管理相关的开发实践。

识别指标

要了解您是否是在为客户提供重要的服务，您必须实施推动改进和加快交付的措施。目标、问题、指标 (GQM) 为确保您的衡量标准符合这些标准提供了一个有效的框架。按照以下步骤，使用此框架从目标出发：

1. 确定您正在实现的目标或成果。
2. 推导出必须回答的问题，以确定目标是否已实现。
3. 决定应该或可以衡量哪些内容才能充分回答问题。有两种类别的衡量标准：
 - **商品指标**，可确保您向买家提供重要内容。
 - **运营指标**，可确保您改善软件交付生命周期。

产品指标

产品指标侧重于业务成果，应在确定新工作范围的投资回报率 (ROI) 时确定。建立产品指标的一种有用方法是询问实施新的工作范围后，业务会发生什么变化。将这种想法正式化为测试的形式很有帮助，该测试侧重于提供现代化功能时的真实情况。

例如，如果您认为将交易迁出遗留系统将为新客户带来新的机会，那么有什么改进呢？必须创建多少容量才能加载下一个客户端？如何构建测试来验证该结果？对于此场景，您的产品指标可能包括以下内容：

- 确定业务价值测试或假设（例如，释放 x% 的交易容量将带来 y% 的新业务）。
- 确定基准（例如，当前的 x 笔交易容量支持 y 个客户）。
- 验证结果（例如，您的容量提高了 x%，那么您现在能否将 y% 的新业务纳入进来？）

运行指标

要确定您是否在改善软件交付生命周期并加快现代化，您必须知道交付软件的交付周期和实施时间。也就是说，您能以多快的速度将业务需求转化为生产中的功能？

有用的运营指标包括：

- 交货时间 — 工作范围从请求到生产需要多长时间？
- 周期时间 — 实施工作范围从头到尾需要多长时间？
- 部署频率 — 您多久向生产部署一次变更？
- 恢复服务的时间 — 从故障中恢复需要多长时间（以平均修复时间或 MTTR 来衡量）？
- 更改故障率 — 平均故障间隔时间（MTBF）是多少？

步骤 2：从小处着手，建立势头

此步骤的目标是提供最初的最小化可行产品 (MVP) 以获得动力。这种方法使您能够尽早并逐步推动业务成果。

验证优先级驱动因素

在开始与应用程序团队合作进行现代化工作之前，我们建议您验证先前确定的优先驱动因素。按照以下步骤进行操作：

1. 从诊断手册中整理您需要的信息。
 - 从优先应用程序列表中收集优先驱动因素和可行性评测。
 - 收集应用程序的过渡和目标状态配置。
 - 确定云现代化规划中的应用程序所有者、架构师和利益相关者。
 - 索取有关依赖关系或应用程序套件顺序的信息（如果已知）。
 - 确定清单条目与依赖关系或应用程序套件分组的关系。应用程序可能有单独的组件，这些组件与其他组件紧密结合或依赖于其他组件，因此您可能希望将这些组件一起现代化。
2. 与第 1 步中的人员安排一小时或两小时的会议，以验证优先驱动因素。
 - 尝试按解决方案工程师或架构师对多个（最多三或四个）应用程序进行分组，然后根据应用程序依赖关系或应用程序套件信息，在一个会议上讨论这些应用程序。
 - 确定每位团队成员在即将举行的会议中的角色和期望。
3. 举行会议。

敲定细节

按照上一节中的流程验证优先驱动因素后，您可以收集详细信息以确定现代化方法和时机。

在此阶段，核心团队将与应用团队并肩合作，进行为期两天的短暂冲刺，为其 AWS 云端应用程序设计未来状态。活动包括产品定义、产品发现、故事写作、价值流映射和设计 CI/CD 流程。下面是一些观点：

- 对应用程序的每个组件进行建模（例如，网络配置、存储配置、数据库、服务器以及应用程序在服务器上的部署方式）。
- 使用容器或无服务器技术等工具，将该模型解构为不同的构建块和配置。
- 将应用程序功能与对底层基础设施的任何依赖关系分开。将应用程序的功能抽象为无需更改任何源代码即可移动的组件。
- DevOps 通过使用 CI/CD 工具和机制与之紧密集成。

构建基础平台服务和实现应用程序现代化

在这个 12 周的阶段中，核心团队将得到全栈团队的支持，以交付按优先顺序排列的业务用例。这项工作由多个双披萨团队进行。例如，成立了一个平台工程团队来开发基础平台服务，并组建了一个产品团队来交付新的业务成果：

- 平台工程团队配置、集成和定制支持云基础、开发人员工作流程和数据分析功能的 AWS 服务。规模更大、更复杂的企业可能有多个团队来为每项功能提供支持。
- 产品团队为初始阶段优先考虑的业务成果开发新的服务和体验。随着产品团队开发新服务，他们还会对核心业务能力进行现代化改造。

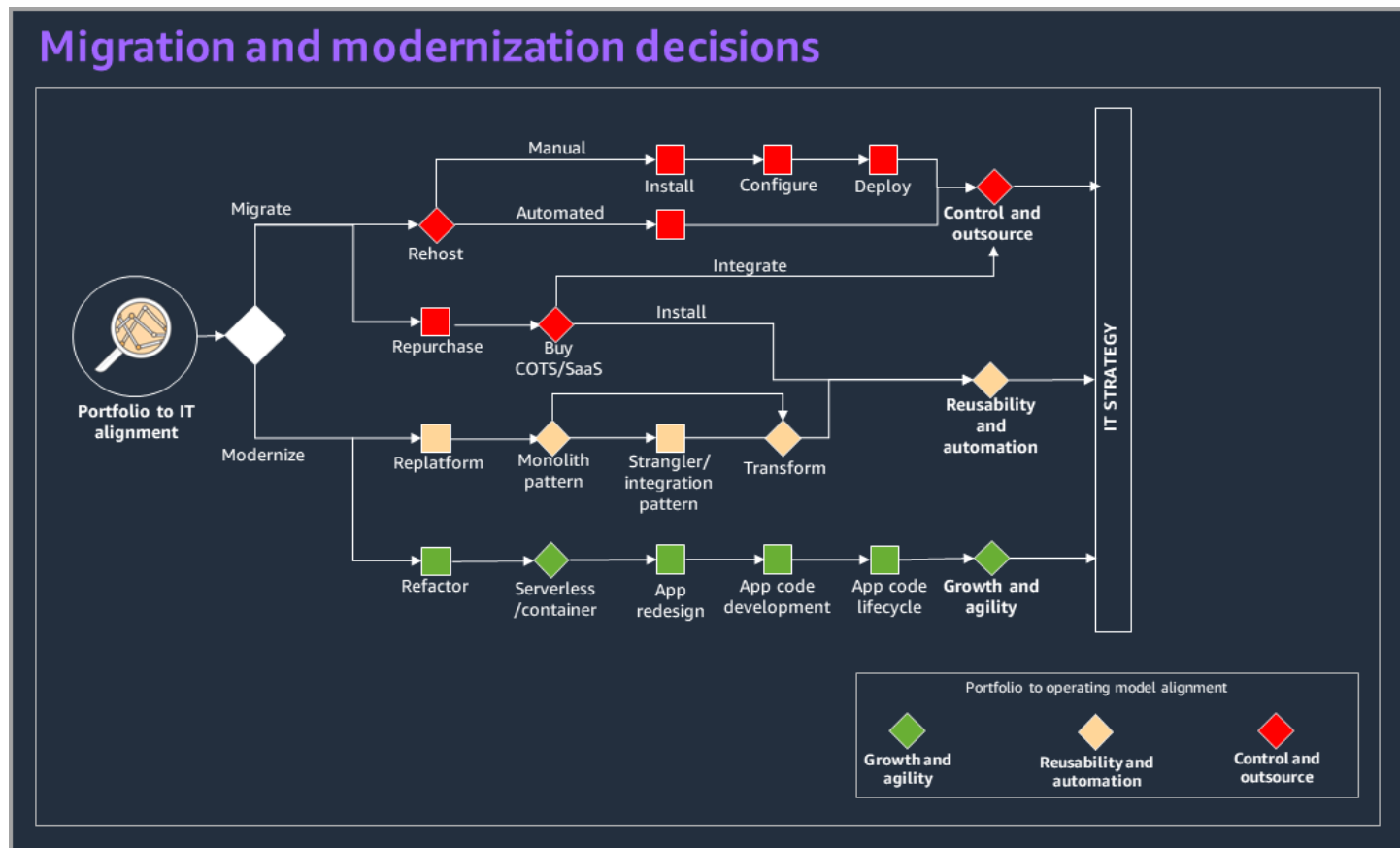
平台工程和产品团队提供您可以评估的最小化可行产品 (MVP)。在最初的 MVP 获得成功后，您可以使用 split-and-seed 方法来扩展现代化计划，即确定新的应用程序，将初始团队成员分组以创建新的产品团队。

步骤 3：创建可扩展的现代化路线图

在首次 MVP 发布优先级结果和应用程序之后，我们建议您制定路线图，以扩展和加快现代化工作、提高开发人员工作效率和快速进行创新。核心团队分裂和培育新的团队，以便将您组织的能力和服务扩展到多个专注于业务成果的工程团队。通过随着时间的推移采用 split-and-seed 方法，您的组织可以进行更多的发展并加快现代化的速度。

现代化路线图应概述一种务实而持续的应用程序现代化方法，并具有明确定义的模式，例如事件驱动、绞杀者、域驱动的设计、分解、现代数据库选项等。

路线图应包括决策树矩阵，如下图所示，用于识别应用程序的组件并将其移至不更改业务逻辑的托管服务（例如数据库服务），或者进行代码级更改以提高性能、可扩展性、可管理性、可靠性和资源使用率。



现代化准备情况因素

对应用程序进行现代化改造时，请遵守以下标准和最佳实践。

主题

- [代码](#)
- [构建和测试](#)
- [版本](#)
- [操作](#)
- [优化](#)
- [准备就绪](#)

代码

- 提供记录软件功能的代码注释，并使用它们来生成文档。
- 遵循代码管理和部署流程，这些流程支持频繁的代码检查和功能请求的可追溯性。
- 构建包含单元测试、功能测试、性能测试和关键路径测试的测试套件，代码覆盖率为 100%。
- 鼓励重复使用代码，以便在代码库中提供相同或相似的功能。
- 在投资完整代码开发之前，开发原型以与用户一起验证功能。

构建和测试

- 根据测试重新定义功能完整性，以提高质量并防止问题反复出现。
- 自动化验收测试。
- 监控所有自动测试，并建立处理故障的流程。
- 跟踪生产和非生产环境中的性能，根据实际流量和负载测试定义服务级别目标 (SLO)，并提供扩展以满足性能要求的能力。
- 从配置文件中提取敏感数据，并提供自动配置和监控配置的工具。

版本

- 通过支持依赖关系（例如数据库发布）、回归测试和跟踪，实现自动化部署。

- 每次成功构建后，以增量方式将代码发布到生产环境。
- 有效地管理功能标志（切换）：支持运行时配置，监控使用情况，在整个开发周期中维护标志，并按类别分配所有者。
- 在构建管道中提供可追溯性，以跟踪触发器、失败通知和成功完成。
- 运行自动部署流程并测试持续交付中的“零接触”代码更新。
- 使用零停机时间、全自动蓝/绿部署方法。
- 确保在所有开发和生产环境中一致实施数据库架构更改。

操作

- 创建与您的通知系统集成的 DevOps 分类运行手册。
- 确保您的监控和通知系统符合服务级别目标 (SLO)，并支持阈值、运行状况检查、非标准 HTTP 响应和意外结果。
- 建立有效的风险管理和灾难恢复流程。
- 制定符合您的业务和法律要求的日志轮换和保留策略。
- 开发仪表盘，用于跟踪产品性能，衡量新功能的成功程度，并在指标未达到预期时显示警报。

优化

- 根据绩效和质量衡量标准，定期审查和改进流程。
- 实施根本原因分析和预防流程，防止问题再次出现。
- 提供数据驱动的指标，以捕捉产品运行状况，并确保所有通知和操作都基于这些指标。

准备就绪

- 让跨职能团队（包括业务合作伙伴、开发人员、测试人员和架构师）专门负责您的现代化工作。

后续步骤

本指南提供了一种分阶段和渐进的方法来实现应用程序的现代化。我们建议您开始使用短期、中期和长期的技术和交付来确定工作流的范围。您可以期望在工具、框架和实践上进行更多投资，以推动跨团队更有效的交付和工程实践。为现代化过程设定目标，了解您计划实现现代化的每个应用程序，并为每个应用程序选择最佳的现代化方法。监控和优化您的进度。

AWS 将与您组织的业务开发团队合作，推动技术现代化并管理您的端到端需求，包括应用程序架构、设计和开发；移动支持；测试和协作解决方案。这些服务结合了成熟的流程、智能自动化、数据和模式的使用、开放标准以及适当的人员，可帮助您实现原有应用程序的现代化。主要目标为：

- 协助开发一个平台，为传统技术现代化提供全自动化、基于工具的方法，并将其与现代化的全面知识相结合。
- 与全栈团队一起打造 MVP，利用现代化工程功能，专注于客户成果和规模。

AWS 专业服务和 AWS 合作伙伴直接与高级技术和业务客户利益相关者合作，推动企业现代化，同时以迭代方式为最终客户交付价值。

资源

相关指南

- [实现云端应用程序现代化的策略 AWS](#)
- [评估 AWS 云端应用程序的现代化准备情况](#)
- [实现云端运营现代化 AWS](#)
- [将单体分解为微服务](#)
- [使用 AWS 无服务器服务集成微服务](#)
- [AWS 变革加速 6 点框架和组织变革管理工具包](#)

AWS 资源

- [AWS 文档](#)
- [AWS 一般参考](#)
- [AWS 术语表](#)

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
扩展了指导方针	—	2023 年 5 月 25 日
初次发布	—	2020 年 12 月 18 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。