



实现制造执行系统 (MES) 的现代化 AWS Cloud

AWS 规范性指导



AWS 规范性指导: 实现制造执行系统 (MES) 的现代化 AWS Cloud

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
架构模式	3
工业边缘计算	3
架构	3
IIoT	4
架构	5
与其他企业应用程序接口	6
架构	6
AI/ML	7
架构	8
数据和分析	9
架构	10
计算容器	11
架构	12
综述	13
将 MES 分解为微服务	14
确定最佳的专用技术	16
计算	17
长时间运行的计算	17
容器	17
事件驱动和无服务器计算	17
数据库	18
关系数据库	18
键值，NoSQL 数据库	18
时间序列数据库	18
云存储	19
用户界面	19
确定微服务的集成方法	20
同步通信	20
异步通信	21
发布/订阅模式	22
混合通信	22
使用云原生技术管理微服务	26
编排	26

审核	27
故障恢复能力	28
可用性	28
灾难恢复	29
结论	31
参考信息	32
AWS 服务	32
AWS 服务家庭	33
其他 AWS 资源	33
作者和贡献者	34
文档历史记录	35
术语表	36
#	36
A	36
B	39
C	41
D	43
E	47
F	48
G	50
H	51
我	52
L	54
M	55
O	59
P	61
Q	63
R	63
S	66
T	69
U	70
V	71
W	71
Z	72
.....	lxxiii

实现制造执行系统 (MES) 的现代化 AWS Cloud

亚马逊 Web Services ([贡献者](#))

2024 年 4 月 ([文档历史记录](#))

制造执行系统 (MES) 起源于1970年代的一组数据收集工具和计划系统的扩展。随着时间的推移，它们已发展成为一种全面的软件解决方案，用于监控、跟踪、记录和控制将原材料转化为成品的生产流程。MES 与现有的车间系统 (例如可编程逻辑控制器 (PLC)、监督控制和数据采集 (SCADA) 系统以及历史记录器) 集成，以实现无缝生产控制。它还与企业资源规划 (ERP) 和产品生命周期管理 (PLM) 系统等企业系统集成，以实现信息从企业到车间的无缝流动。

借助云计算，企业越来越多地希望将 MES 迁移到云端，以提高可扩展性、灵活性和性能效率并降低成本。此外，物联网 (IoT)、人工智能和机器学习 (AI/ML) 以及微服务的出现正在颠覆MES的格局。除了在云端托管传统的单体MES外，制造商和为制造商服务的独立软件供应商 (ISV) 现在还可以选择使用微服务开发模块化MES。在传统的整体式 MES 或现代 MES 之间做出选择可能具有挑战性，需要对组织能力、预算分配、期限预期和业务优先事项进行全面分析。使用API的现代、云原生、基于微服务的MES是利用第四次工业革命 (工业4.0) 概念的企业的的首选，因为它具有敏捷性、可扩展性、灵活性、更快的价值实现时间以及与物联网的兼容性。

现代 MES 具有以下优点：

- 它支持敏捷开发，并通过修改特定服务来支持频繁更新，而不是影响整个应用程序，并且可以适应不断变化的业务流程。
- 微服务通过各种编程语言、数据库和用户界面技术提供技术灵活性并满足独特的需求。
- 它具有可扩展性，因此适用于可能具有不同生产流程的地理位置分散的制造商。
- 它允许快速响应不断变化的客户需求和供应链中断，从而缩短上市时间。

通过采用基于微服务的 MES，企业可以利用工业 4.0 的优势。本指南描述了一种使用 AWS 服务和技术实现基于微服务的 MES 的方法。这种方法包括根据具体的业务结果确定微服务结构，并为每个结果选择正确的技术。该指南提出了集成、增强、监控和管理这些微服务的可能方法。基于微服务的架构在操作上往往很复杂。因此，该指南还分享了制造商如何简化基于微服务的 MES 的运营管理的最佳实践和架构模式。它提供了可用的选项，并为决策者提供了指导。决策的最终责任由架构师、分析师和技术领导者承担，他们必须根据自己的独特情况、预期的业务结果和可用资源来确定最合适的选择。

在本指南中：

- [基于微服务的现代 MES 的架构模式](#)

- [将 MES 分解为微服务](#)
- [确定适用于 MES 的最佳专用技术](#)
- [确定 MES 中微服务的集成方法](#)
- [使用云原生技术管理、编排和监控 MES 的微服务](#)
- [MES 中的弹性](#)
- [结论](#)
- [参考](#)
- [作者和贡献者](#)

基于微服务的现代 MES 的架构模式

为了解锁宝贵的见解、推断模式、预测事件以及自动化质量检查和数据收集等手动流程，MES 可以使用云原生技术，例如工业物联网 (IIoT)、AI/ML 和数字双胞胎。以下各节将讨论一些最常见的用例及其架构模式：

- [工业边缘计算](#)
- [IIoT](#)
- [与其他企业应用程序接口](#)
- [人工智能/机器学习](#)
- [数据和分析](#)
- [计算容器](#)

有关这些架构所包含的微服务的更多信息，请参阅本指南后面的将 [MES 分解为微服务](#) 部分。

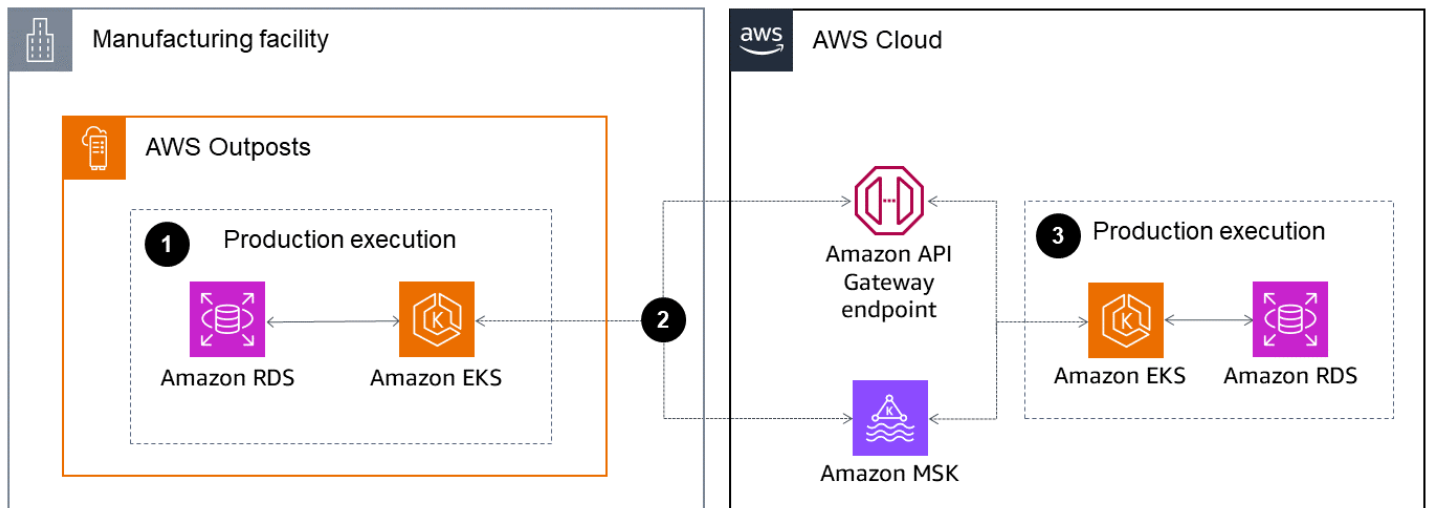
工业边缘计算

MES 对制造运营至关重要。MES 中的某些微服务或功能需要低延迟，并且不能容忍间歇性地连接到云。这些微服务更适合在本地运行。[AWS 边缘服务](#) 将云端提供的基础架构 APIs、服务和工具扩展到本地数据中心或托管空间。AWS 边缘服务可用于基础架构、存储、内容交付、坚固且断开连接的边缘、机器人、机器学习和物联网。

架构

许多 MES 事务都对延迟敏感。本指南后面引用的示例之一是生产执行服务。生产执行服务的功能之一是引导 work-in-progress 货物的流动。由于这是一项敏感活动，因此对延迟的容忍度可能很低，制造商可能需要此微服务的本地组件。

以下是此用例的示例架构。



1. 用于计算的亚马逊 Elastic Kubernetes Service (Amazon EKS) 和用于数据库的亚马逊关系数据库服务 (Amazon RDS) 托管在本地。AWS Outposts您也可以使用自我管理的硬件来托管边缘组件。某些功能，例如 Amazon EKS Anywhere，也可以用于自我管理的硬件。
2. 这些服务的边缘组件可以通过两个容器实例之间的 Amazon API Gateway 终端节点与云组件同步。

另一种选择是在两个容器实例之间设置服务总线以使它们保持同步。您可以使用适用于 Apache Kafka 的亚马逊托管流媒体 Kafka (亚马逊 MSK) 来设置此类服务总线。

3. 制造商可以使用微服务的云组件来处理对延迟不太敏感的案例，例如向PLM系统发送更新以改进流程，将确认发送到ERP系统进行生产，以及将数据导出到数据湖进行报告和分析。由于云的经济性、规模和灾难恢复优势，制造商可以在微服务的云实例中长时间存储数据。

工业物联网 (IIoT)

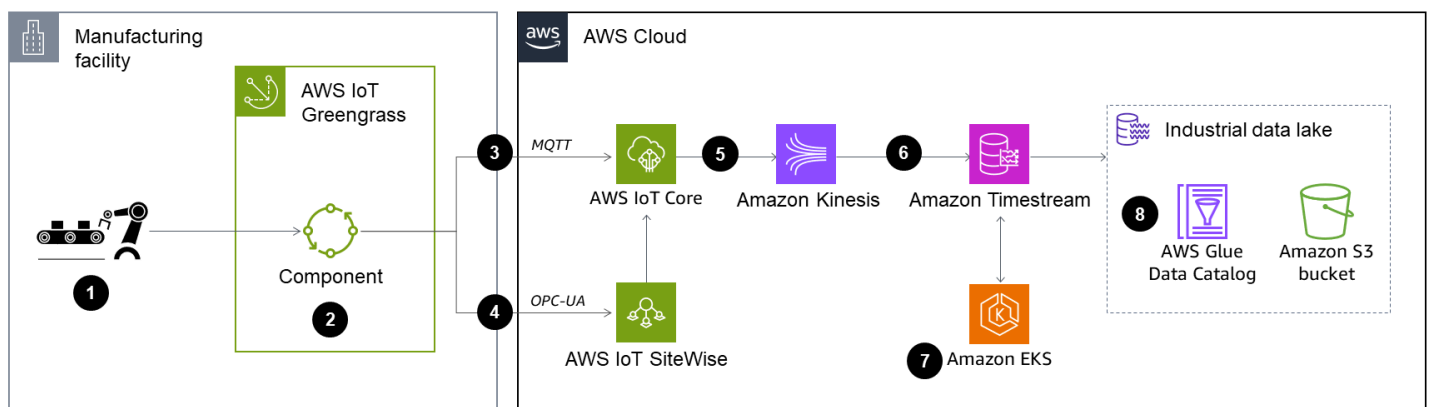
典型的制造工厂有成千上万的传感器和设备，可以生成大量数据。这些数据大部分未被使用。MES 可以将这些数据情境化，并在云原生服务的帮助下使其可用。MES 还可以连接机器和设备，自动收集信息（例如，从过程参数和测试结果中收集信息），并使用这些信息对事件进行实时响应，节省时间，并消除因手动输入而出现错误的风险。例如，您可以从测试机收集结果，确定产品质量，并以自动方式创建不合格记录或二次检测工作流程，而无需手动输入任何数据。随着时间的推移，云原生物联网服务可以帮助找到缺陷的特定模式和根本原因，并且您可以通过修改制造过程来防止缺陷的发生。

AWS 提供广泛而深入的解决方案，用于解锁您的物联网数据并加快实现业务成果。这些解决方案包括 [AWS Partner 解决方案](#) 和 [AWS 服务](#)，它们是基于客户独特需求的架构的基石。您可以作为构建块包含在架构中的 AWS 物联网服务包括以下内容：

- [AWS IoT Greengrass](#) 是一款 IoT 开源边缘运行时和云服务，可帮助您构建、部署和管理设备软件。边缘运行时或客户端软件在本地运行，并且与各种硬件兼容。它支持本地处理、消息传递、数据管理和机器学习推理，并提供预先构建的组件来加快应用程序开发。AWS IoT Greengrass 可以与 MES 的边缘组件交换数据，用于对延迟敏感的用例。
- [AWS IoT Core](#) 是一个托管云平台，可让联网设备轻松安全地与云应用程序和其他设备进行交互。AWS IoT Core 可以安全可靠地支持数十亿台设备和数万亿条消息，并且可以处理这些消息并将其路由到 AWS 终端节点和其他设备。在您使用时 AWS IoT Core，您的应用程序可以随时跟踪您的所有设备并与之通信，即使这些设备未连接也是如此。
- [AWS IoT SiteWise](#) 是一项托管服务，使工业企业能够在多个工业设施中收集、存储、组织和可视化成千上万的传感器数据流。AWS IoT SiteWise 包括在位于设施现场的网关设备上运行的软件，持续从历史学家或专业工业服务机构那里收集数据，然后将其发送到云端。您可以进一步分析云端收集到的数据，并将其用于仪表板，或将其提供给 MES 以获取对结果和趋势的响应。

架构

根据独特的环境因素，典型的物联网数据摄取和处理架构可以采取多种形式。最常见的用例是从本地网络上的计算机收集数据，并将这些数据安全地发送到云端。以下是此用例的示例架构。



1. 机器或数据源：这些可能是连接到网络并可以自己共享数据的智能机器，也可以是其他数据源，例如 PLCs 和历史学家。来自这些来源的数据可以采用不同的协议，例如 MQTT 和 OPC-UA。
2. AWS IoT Greengrass 安装在 Greengrass 核心设备上，其组件用于从数据源收集数据并将其发送到云端。
3. MQTT 协议中的数据将转至。AWS IoT Core AWS IoT Core 根据配置的规则进一步重定向此数据。

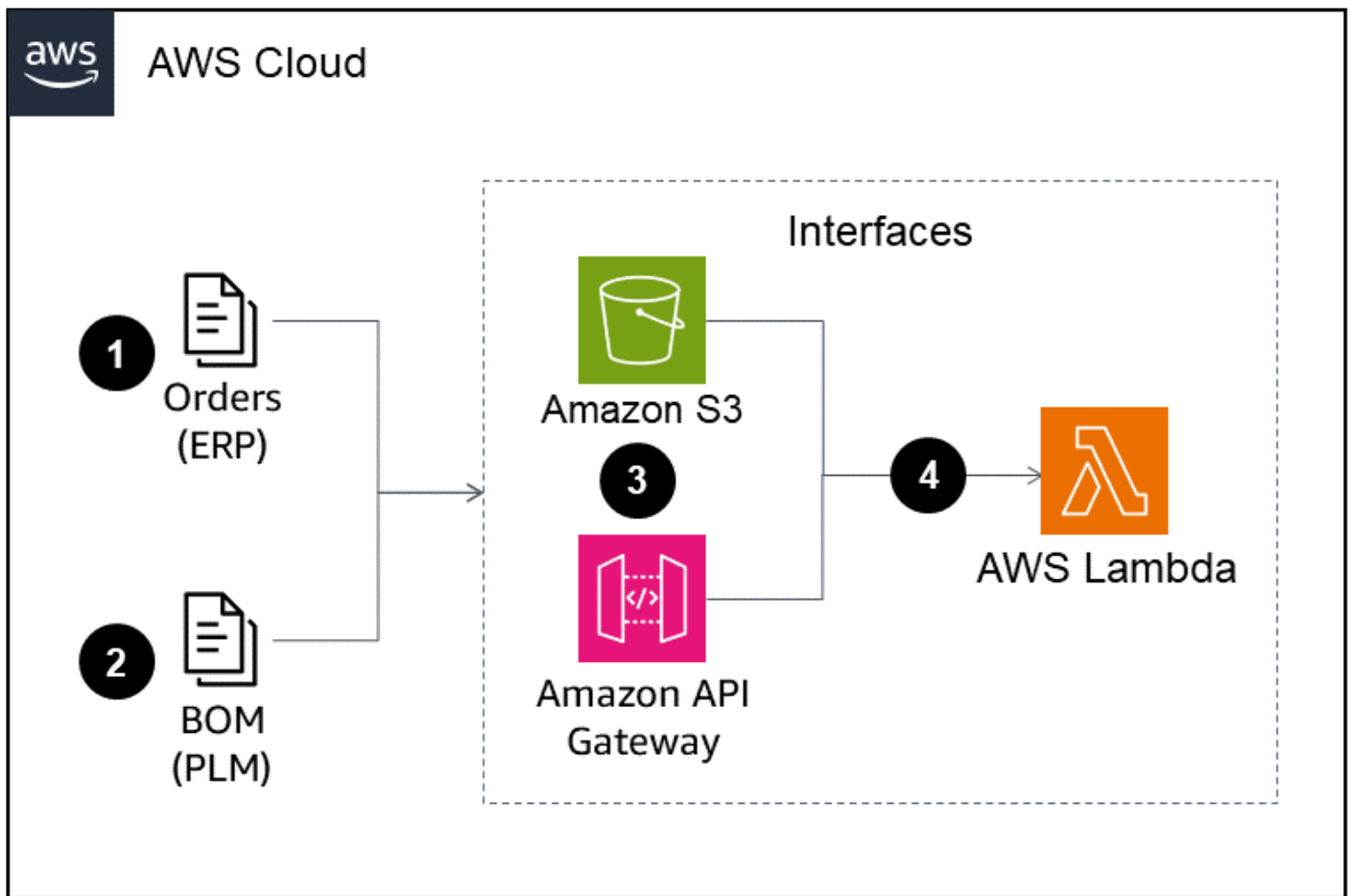
4. OPC-UA 协议中的数据将转至。AWS IoT SiteWise Organizations 可以使用 AWS IoT SiteWise 门户网站对这些数据进行可视化。数据被馈送到数据湖，AWS IoT Core 并最终输入数据湖以进行情境化，并将其与其他系统的数据相结合。
5. Amazon Kinesis 会从中流式传输数据 AWS IoT Core 以进行存储。AWS IoT Core 有一个功能[规则](#)，使其能够与其他人进行交互 AWS 服务。
6. 亚马逊 Timestream 数据库存储数据。这只是一个示例，你可以根据数据的性质使用任何其他类型的数据库。
7. Amazon EKS 管理微服务中 Kubernetes 控制平面节点的可用性和可扩展性。
8. 您可以将从机器和其他运营技术 (OT) 数据源摄取的数据馈送到数据湖。

与其他企业应用程序接口

由于 MES 处于运营技术 (OT) 和信息技术 (IT) 的边缘，因此它必须与企业应用程序和 OT 数据源进行交互。根据组织解决方案格局，MES 可以与 ERP 交互以获取生产和采购订单信息、有关零件和产品的主数据、库存可用性以及物料清单。MES 还将向 ERP 报告订单状态、生产过程中的实际物料和劳动力消耗以及机器状态。如果存在 PLM，MES 可以与之交互以获取详细的流程清单 (BOP)、工作指令，在某些情况下还包括物料清单 (BOM)。MES 还将向 PLM 报告流程执行信息、不合格和 BOM 差异。

架构

考虑到 PLM 和 ERP 系统的种类繁多，这种模式的设计会根据 MES 与之交互的系统而有所不同。下图说明了一个示例架构。



1. Organizations 可能在 AWS Cloud 或其他地方有 ERP 实例。
2. 与ERP一样，PLM系统可能位于 AWS Cloud 或其他地方。
3. 组织可以将数据从 ERP 和 PLM 导入到亚马逊简单存储服务 (Amazon S3) 存储桶。如果这些系统托管在中 AWS Cloud，则文件存储库可能是另一个 S3 存储桶，可以为 MES 进行复制。连接这些应用程序的另一种方法是使用 Amazon API Gateway 通过 API。
4. 无论组织如何从 ERP 和 PLM 导入数据，AWS Lambda 函数都可以处理接收到的信息并将数据路由到微服务数据库，因为 ERP 和 PLM 接口以及此类数据处理主要由事件驱动。

人工智能和机器学习 (AI/ML)

通过在 MES、机器、设备、传感器和其他系统生成的数据上使用人工智能 (AI) 和机器学习 (ML)，您可以优化制造运营并为您的业务获得竞争优势。AI/ML 将数据转化为见解，您可以主动使用这些见解来优化制造流程、实现机器的预测性维护、监控质量以及自动化检查和测试。AWS 为所有技能水平提供全

面的 [AI/ML 服务](#)。机器学习的 AWS 方法包括三个层次。随着时间的推移，大多数拥有强大技术能力的组织都将使用这三种能力。

- 底层包括面向机器学习专家和从业者的框架和基础架构。
- 中间层为数据科学家和开发人员提供机器学习服务。
- 顶层是模仿人类认知的人工智能服务，适用于不想构建机器学习模型的用户。

以下是一些面向工业企业的著名 AWS 机器学习服务：

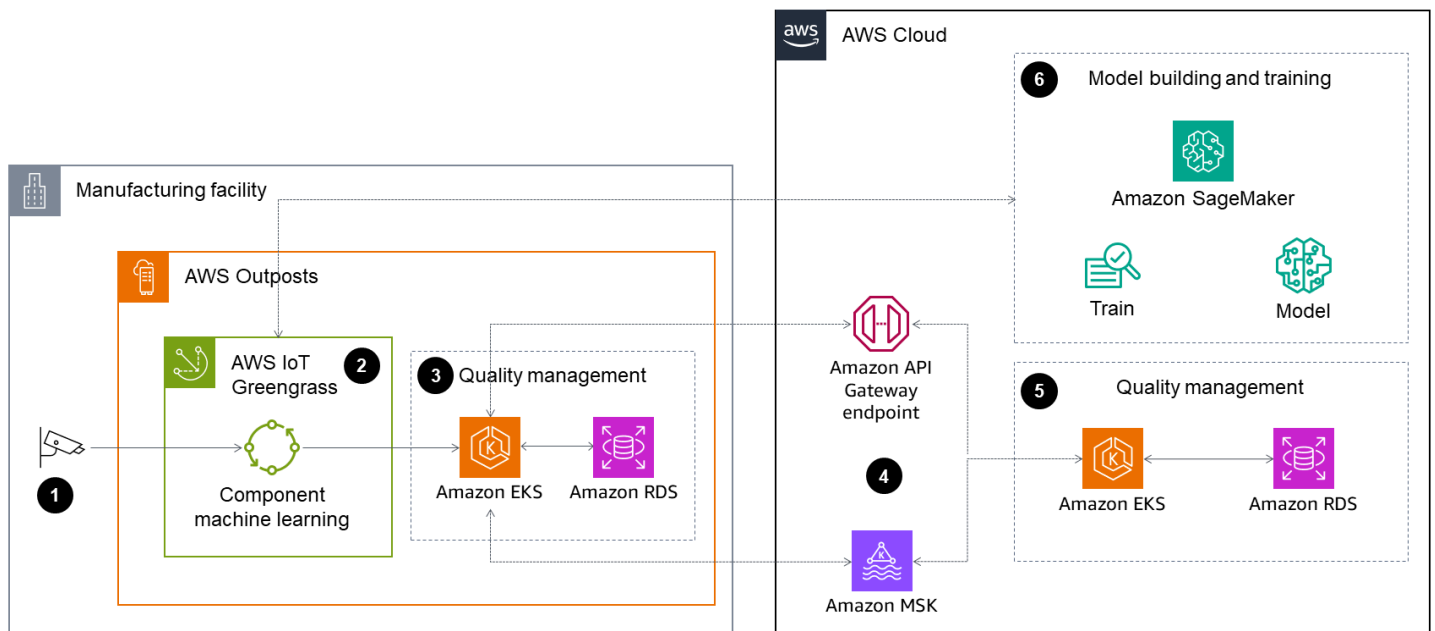
- [Amazon SageMaker AI](#) 是一项完全托管的服务，可使用完全托管的基础架构、工具和工作流程为任何用例准备数据并构建、训练和部署机器学习模型。
- [AWS Panorama](#) 提供机器学习设备和 SDK，可将计算机视觉 (CV) 添加到您的本地摄像头，从而以高精度和低延迟进行自动预测。借 AWS Panorama 助，您可以在边缘使用计算机功能（无需将视频流式传输到云端）来改善操作。AWS Panorama 自动执行监控和目视检查任务，例如评估制造质量、发现工业过程中的瓶颈以及评估工厂内的工人安全。您可以将这些自动化任务的结果传送 AWS Panorama 到 MES 和您的企业应用程序，以进行流程改进、质量检验计划和竣工记录。

终止支持通知

2026 年 5 月 31 日，AWS 将终止对的支持。AWS Panorama 2026 年 5 月 31 日之后，您将无法再访问 AWS Panorama 控制台或 AWS Panorama 资源。有关更多信息，请参[阅 AWS Panorama 终止支持](#)。

架构

在制造质量管理中，自动质量检测是计算机视觉和机器学习最受欢迎的用例之一。制造商可以在传送带、搅拌机滑槽、包装站、库存室或实验室等位置放置摄像头以获得视觉效果。该相机可以提供高质量的视觉缺陷或异常图像，帮助制造商对多达 100% 的零件或产品进行检测，提高检测精度，并解锁洞察力以进一步改进。下图显示了自动质量检测的典型架构。



1. 能够在网络上通信的摄像机共享图像。
2. AWS IoT Greengrass 托管在本地，并提供用于推断图像中任何异常的组件。
3. 对于延迟敏感的用例，质量管理边缘服务在本地处理上一步的推理输出的结果。AWS Outposts 托管计算和数据库资源。制造商可以扩展此组件架构，根据推理结果向利益相关者发送警报或消息。制造商还可以使用其他兼容的第三方硬件在边缘托管服务。
4. 这些服务的边缘组件可以通过两个容器实例之间的 Amazon API Gateway 终端节点与云组件同步。另一种选择是在两个容器实例之间设置服务总线以使它们保持同步。你可以使用适用于 Apache Kafka 的亚马逊托管流媒体 Kafka (亚马逊 MSK) 来设置此类服务总线。
5. 制造商可以使用微服务的云组件来处理对延迟不太敏感的案例，例如处理质量检查以填充历史表，以及将更新发送到 PLM 系统以获取未来流程和零件设计改进的质量结果。由于云的经济性、规模和灾难恢复优势，客户可以在云微服务实例中长时间存储数据。
6. 您可以使用 Amazon SageMaker 等云原生机器学习服务在云中构建和训练模型。您可以将最终训练好的模型部署在边缘进行推理。边缘组件还可以将数据反馈到云端以重新训练模型。

数据和分析

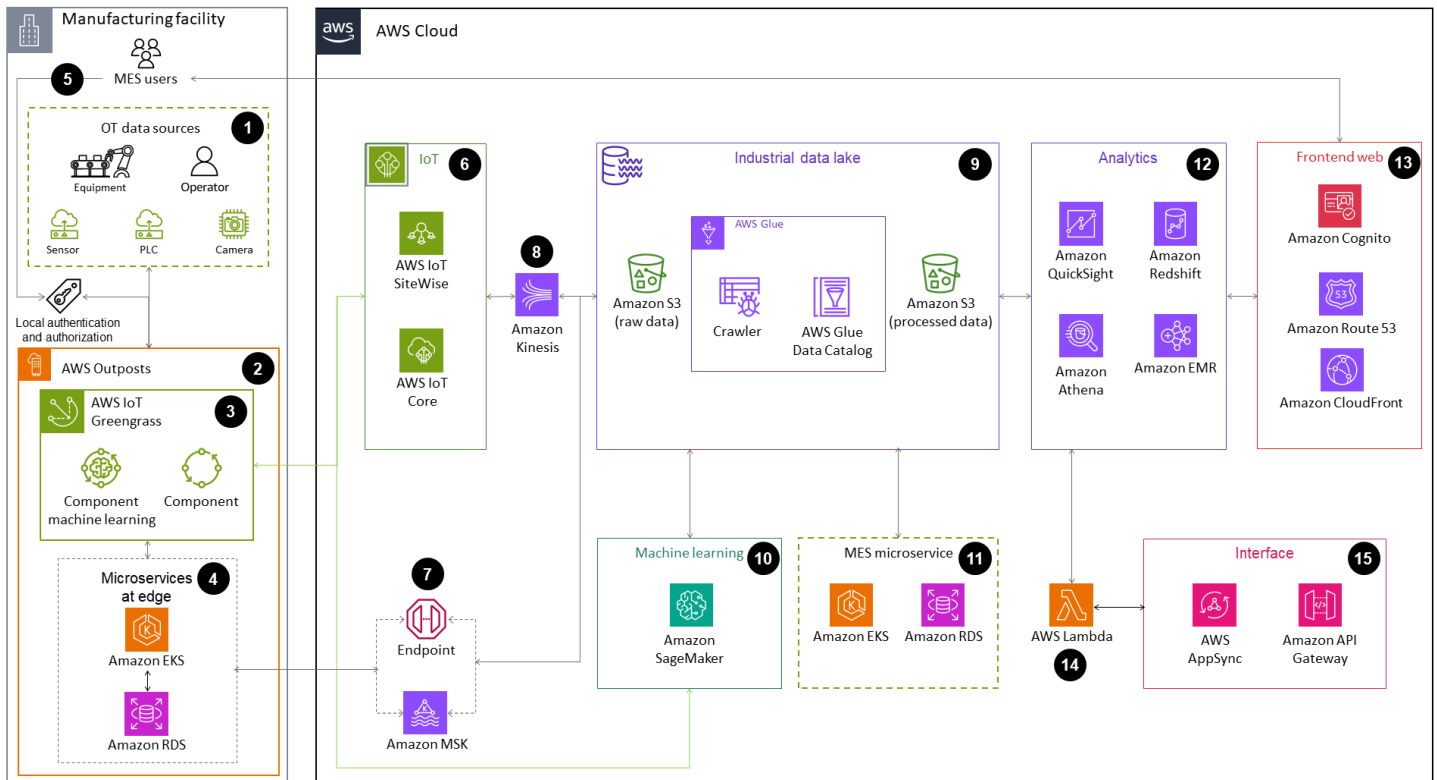
传统的整体式 MES 系统的分析能力有限或根本没有。制造商不得不依靠昂贵的第三方工具或复杂的方法将后端数据提取到电子表格中，以获取基本报告，例如每日产量、库存水平、质量结果等。将 MES 数据与其他应用程序和系统数据结合起来进行分析的可能性很小。基于微服务的 MES AWS 可以解决 MES 面临的典型分析挑战，并提供额外的分析功能，为制造商提供竞争优势。AWS Cloud 它为制造

商提供了从一系列专门构建的分析服务和内置分析平台中进行选择，还为工业客户提供专门构建的解决方案，例如工业数据结构。

- [AWS 分析服务](#) 专为使用最合适的工作工具快速提取数据见解而设计，并经过优化，可提供最佳性能、规模和成本以满足业务需求。
- [Industrial Data Fabric](#) 有助于大规模管理来自多个数据源的数据。企业可以通过将 MES 数据与制造业各系统中孤立的数据相结合，优化整个价值链和职能的运营。传统上，制造业中的系统和应用程序要么不通信，要么严格按照层次结构进行通信。例如，PLM 系统不与 OT 系统（例如 SCADA 或 PLC）通信。因此，来自生产和流程设计的数据不会合并，因为这些系统不是为协同工作而设计的。MES 将两者联系起来，但传统的单体 MES 在与企业应用程序和 OT 系统的通信方面也受到限制。上的 Industrial Data Fabric 解决方案可 AWS 帮助您创建数据管理架构，该架构支持可扩展、统一和集成的机制，从而有效地使用数据。

架构

下图显示了数据和分析的示例架构，该架构结合了来自物联网、MES、PLM 和 ERP 的数据。此架构仅建立在 AWS 服务之上。但是，如前所述，您可以使用 AWS Partner 解决方案进行数据分析，并通过结合来自 AWS 和 AWS 合作伙伴的服务来满足您环境的独特需求。



1. 要组合的 OT 数据源可在本地网络上使用。

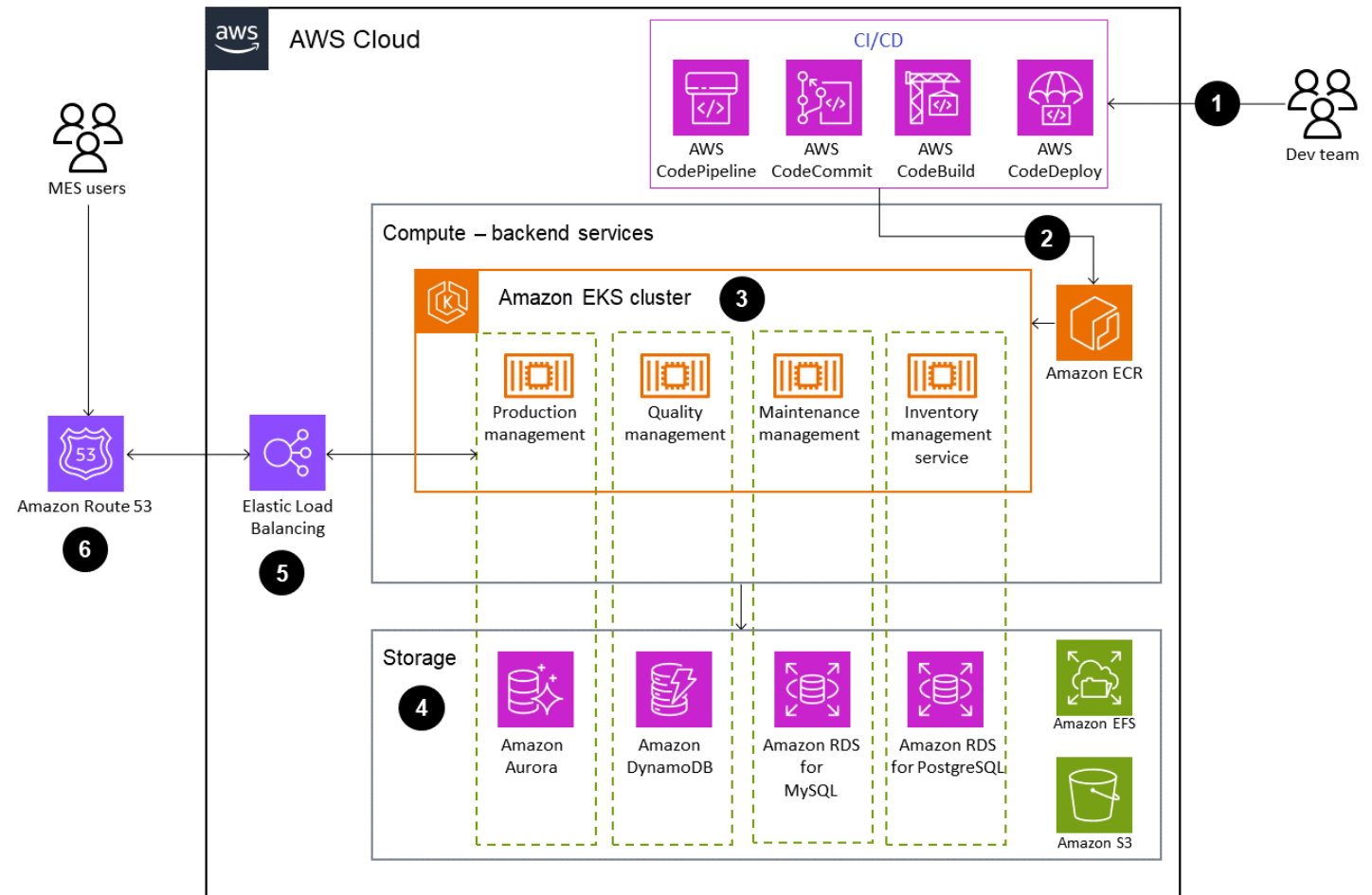
2. AWS Outposts 提供边缘硬件。
3. AWS IoT Greengrass 服务包括用于本地推理的 ML 组件和其他用于数据摄取、处理、流式传输等的组件。
4. MES 微服务的本地实例可以是任何微服务，而且，根据要求，边缘可以有多个微服务。
5. 本地身份验证和授权允许 MES 用户在延迟敏感的用例（例如实时生产报告）或连接中断时安全地访问本地微服务。
6. 物联网服务，例如在云端 AWS IoT Core 接收数据，以及 AWS IoT SiteWise 存储和处理数据。
7. Amazon API Gateway 终端节点和 Amazon MSK 选项可使微服务的云和边缘组件保持同步。
8. Amazon Kinesis 将物联网服务中的数据流式传输到亚马逊 S3 存储桶。Kinesis 允许在将数据存储到 S3 存储桶之前对其进行缓冲和处理。
9. 工业数据湖包括 S3 存储桶、AWS Glue 爬行器和 AWS Glue Data Catalog。AWS Glue 爬虫会扫描包含原始数据的 S3 存储桶以自动推断架构和分区结构，并使用包含已处理数据的 S3 存储桶中的相应表定义和统计数据填充数据目录。
10. 诸如 Amazon SageMaker 之类的机器学习服务用于分析数据湖中的数据，并得出预测未来事件的模式。
11. MES 微服务由 MES 中微服务的云组件组成。
12. 分析服务支持无服务器查询来自数据湖、数据仓库（Amazon Athena）的数据、使用商业智能服务的交互式可视化（Amazon QuickSight）、用于运行复杂查询的可选云数据仓库（Amazon Redshift）以及可选的高级数据处理（Amazon EMR）。
13. 前端网络服务包括用于对用户进行身份验证的 Amazon Cognito、作为 DNS 服务的亚马逊 Route 53 以及用于以 CloudFront 低延迟向最终用户交付内容的亚马逊。
14. AWS Lambda 启用分析服务与其他应用程序之间的接口。
15. 接口服务包括用于管理、整合 APIs 和 AWS AppSync 创建端点的 API Gateway。

计算容器

容器是包含微服务的现代 MES 的热门选择。容器是 MES 开发人员打包和部署其应用程序的强大方式——它们轻量级，可为 MES 应用程序提供一致、便携的软件，便于在任何地方运行和扩展。容器也是运行批处理作业（例如接口处理）、为自动质量检查等用例运行机器学习应用程序以及将传统的 MES 模块迁移到云端的首选。几乎所有 MES 模块都可以使用容器进行计算。

架构

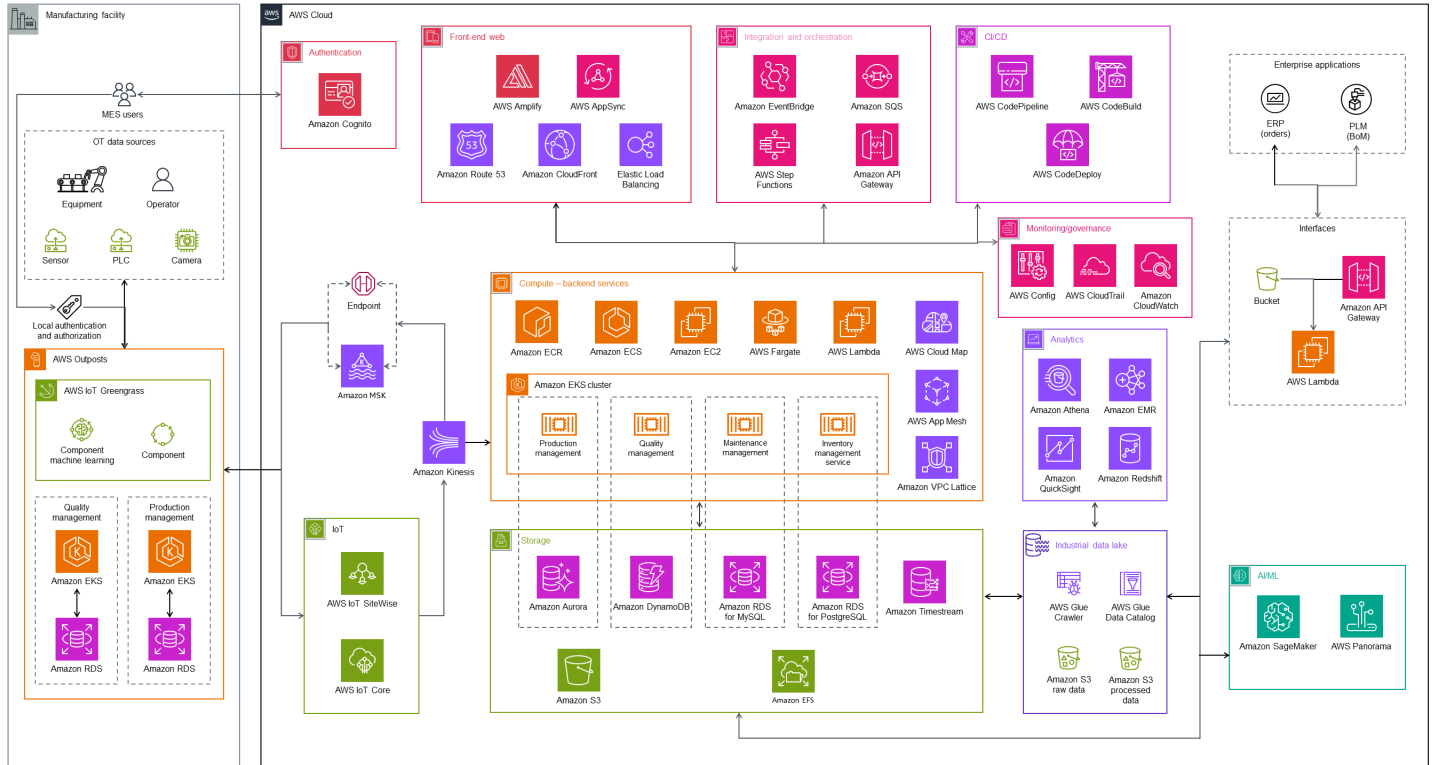
下图中的架构将 DNS 和负载均衡相结合，通过后端容器化计算提供一致的用户体验。它还包括用于持续更新的持续集成和持续部署 (CI/CD) 管道。



1. MES 开发团队使用 AWS CodePipeline 来构建、提交和部署代码。
2. 新的容器镜像被推送到亚马逊弹性容器注册表 (Amazon ECR) Container Registry。
3. 完全托管的 Amazon Elastic Kubernetes Service (Amazon EKS) 集群支持 MES 微服务的计算功能，例如生产管理和库存管理。
4. AWS 数据库和云存储服务用于支持微服务的独特需求。
5. Elastic Load Balancing (ELB) 会自动将 MES 模块的传入流量分配到一个或多个可用区的多个目标上。有关更多信息，请参阅 Amazon EKS 文档中的[工作负载](#)。
6. Amazon Route 53 可用作 DNS 服务，用于解析主服务器中向负载均衡器发送的传入请求 AWS 区域。

综述

基于微服务的成熟的 MES 架构结合了本指南中描述的所有用例、集成工具以及编排服务和方法。但是，架构的细节可能会因独特的环境因素而异，例如用于确定微服务边界、演变和 MES 随时间推移的增强的标准。下图说明了一种典型的架构，该架构结合了前几节中讨论的使用场景。

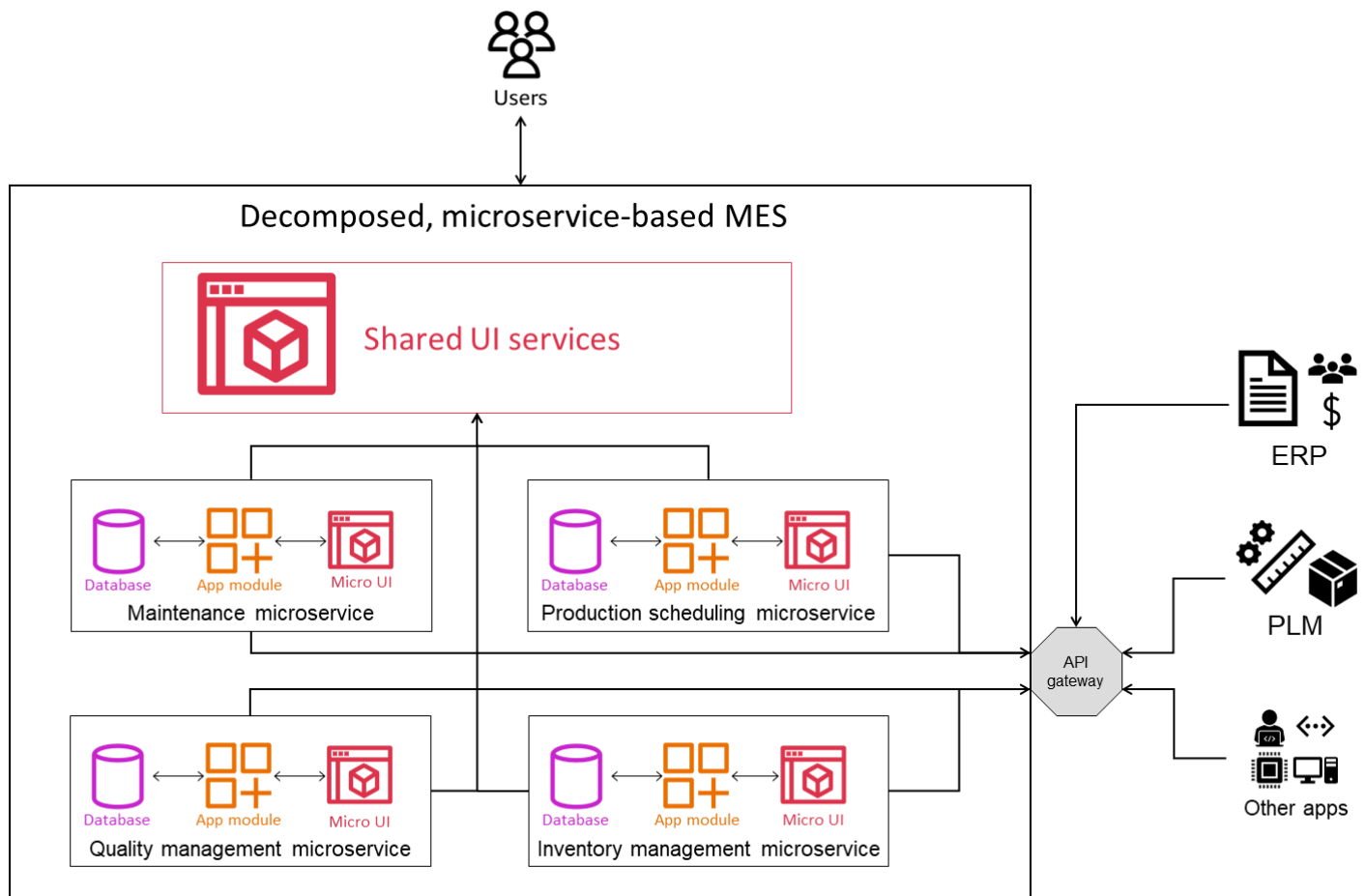


将 MES 分解为微服务

制造现场的 MES 部署可能从几个月到几年不等，因为 MES 通常需要大量的定制和配置，以适应组织流程的独特要求。部署包括映射和配置工作流程、定义用户角色和权限、设置数据收集、集成车间和企业系统以及制定报告和分析要求。制造现场必须详细定义其工作流程，并采用可以数字化和自动化的结构。这可能涉及重大的组织变革、流程重新设计和广泛的再培训。还需要进行严格的测试，以识别和解决任何问题或差异。这些实施挑战、集成和功能可能会阻碍 MES 的实施。

为了缓解 all-in-one MES 部署的实施挑战，制造商可以采用渐进的方法。首先，优先考虑一组可显著有利于制造运营的有限功能。将 MES 分解为更小、可管理的微服务，这些微服务专为满足优先需求而量身定制。然后，随着系统的成熟，逐步添加更多功能和微服务。这种模块化方法增强了灵活性，可以根据制造需求进行有针对性的改进。这样可以使实施过程更加顺畅和有效。

下图显示了 MES 中基本微服务的示例。



这些微服务包括：

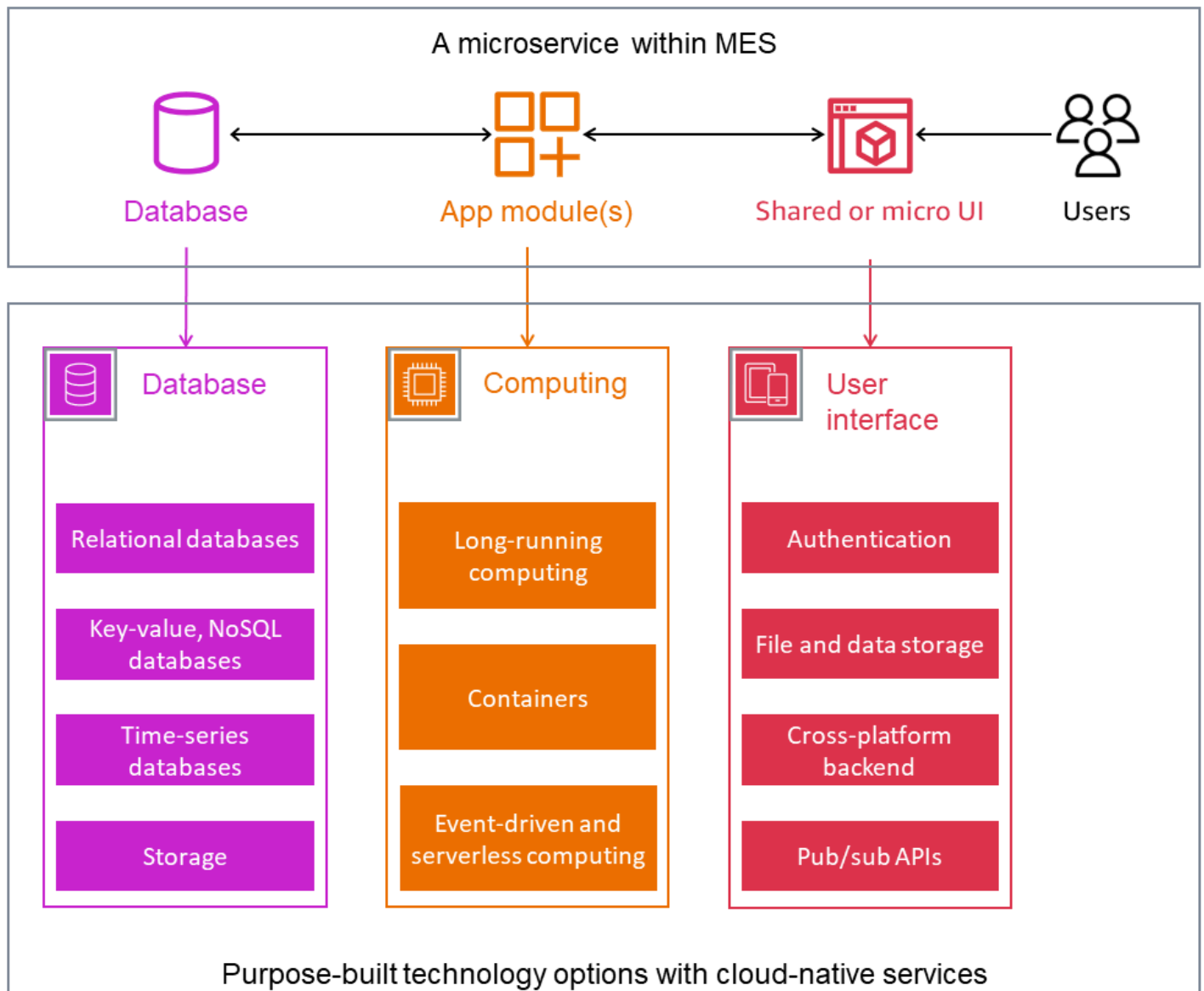
- 生产调度服务可创建工作单并安排生产运行。它可能会连接到其他系统或微服务以跟踪生产状态并确保适当的资源分配。
- 库存管理服务跟踪和管理生产所需的库存水平。它还可能与生产调度服务连接，以确保库存可用于计划的生产运行。
- 维护管理服务监控设备的运行状况，跟踪其使用情况，创建预测性维护警报，跟踪维护并捕获维护历史记录。
- 质量管理服务负责质量控制活动，例如产品和材料检验以及质量保证。它可以帮助管理质量控制工作流程，捕获测试结果并生成质量报告。它还可能与生产计划服务连接以安排检查任务，并与库存管理服务连接以进行物料检查和跟踪。
- 生产执行服务管理生产订单的执行并跟踪生产活动。它可以捕获与生产运行相关的所有数据，包括机器状况、操作员操作和物料消耗。它还可能与生产计划服务连接以获取有关生产订单的信息，与库存管理服务连接以跟踪物料的可用性和消耗情况，以及用于特定质量工作流程的质量管理服务。

除了制造运营特定的服务外，还需要标准服务来管理整个服务堆栈中的共享功能。以下是共享服务的一些示例：

- 用户管理服务处理用户身份验证和授权。它为用户相关操作提供了 API，为其他服务提供了用户上下文。
- 报告和分析服务为其他服务生成的所有数据提供报告和分析功能。它支持性能监控，并允许制造商做出以数据为依据的决策。
- 用户界面服务提供了用于与 MES 系统交互的标准用户界面。它与其他服务连接以检索数据和发送命令。它提供仪表盘、报告和可视化工具，供用户配置应用程序并与之交互。

确定适用于 MES 的最佳专用技术

在将 MES 分解为微服务并根据对业务结果的影响确定开发优先级之后，下一个任务是确定特定微服务和整个系统的技术堆栈。通常，MES 及其本质上的微服务是两层应用程序，包括应用程序或计算层以及持久层或数据库层。用户界面通常是所有微服务之间的共享服务。UI 的不同组件可以是每个微服务所独有的，或者每个微服务可以有自己的 micro-UI 组件。这些微服务将具有不同的计算和数据存储要求，可能需要其他技术堆栈，如下图所示。例如，使用关系数据库进行长时间运行的计算可能是某些微服务的最佳选择，而事件驱动的按需计算和 NoSQL 数据库可能更适合其他微服务。AWS 为每个技术层提供了广泛的选项，因此您可以根据微服务的目的选择最佳服务。



以下各节描述了计算和数据库的可用选项，并说明了如何根据微服务的功能要求选择适当的技术。

计算

传统上，企业总是使用实例（长时间运行的计算）来运行计算操作。这些实例允许您将应用程序的所有资源放在一个盒子上。使用云计算，您可以采用多种计算方式。除了传统的长时间运行计算外，您还可以使用较小的计算单元，例如容器，在容器中构建较小的微服务以实现快速移动和便携性，或者使用事件驱动的计算，其中服务器和集群均由管理。AWS

长时间运行的计算

MES中一些计算密集型和长时间运行的微服务需要高性能或持久的计算资源，例如，处理从PLM接收的大型设计文件，处理机器学习模型的质量检查图像和视频，通过合并来自所有微服务的数据来执行数据分析，或者使用机器学习根据历史数据预测模式。当微服务需要长时间运行的计算能力来实现低延迟应用程序和功能（例如自动扩展、广泛的操作系统支持和硬件支持）时，Amazon [Elastic Compute Cloud \(Amazon EC2\)](#) 是一项在云中提供安全、可调整大小的计算容量的服务。Amazon EC2 还可用于架构组件，这些组件继承自传统应用程序并迁移到云端，无需立即进行现代化改造。

容器

MES 中的大多数微服务，例如生产调度、生产执行、质量管理等，都不需要高性能计算。这些服务不是由事件驱动的，而是始终如一地运行。在这种情况下，容器是基于微服务的架构中最受欢迎的计算资源选择之一，因为容器具有便携性、隔离性和可扩展性优势，尤其是在需要一致的运行环境和高效的资源利用率的情况下。

当容器可以满足微服务的计算要求时，您可以使用来自 AWS 的 [容器编排服务](#)，例如亚马逊 Elastic Kubernetes Service (Amazon EKS) 或亚马逊弹性容器服务 (Amazon ECS) Amazon ECS。这些服务使您可以更轻松地管理底层基础架构，从而构建安全的微服务，选择正确的计算选项，并以高可靠性 AWS 进行集成。

事件驱动和无服务器计算

基于微服务的架构包括根据事件启动的任务，例如处理来自 ERP 和 PLM 的数据，以及生成警报，让维护经理或主管派遣机械师到现场。 [AWS Lambda](#) 对于此类情况，可能是一个不错的选择，因为它是一种事件驱动的计算服务，可按需运行应用程序任务。Lambda 不需要对运行时和服务器进行管理或管理。要创建 Lambda 函数，您可以使用它支持的语言之一编写代码，例如 NodeJS、Go、Java 或 Python。有关支持的语言的更多信息，请参阅 [Lambda 文档中的 Lambda 运行时](#)。

数据库

传统的单体 MES 主要使用关系数据库。关系数据库非常适合大多数用例，但只有少数用例才是最佳选择。借助基于微服务的 MES，您可以为每项微服务选择最佳的专用数据库。AWS 提供[八个数据库系列](#)，包括关系数据库、时间序列数据库、键值数据库、文档数据库、内存数据库、图形数据库和账本数据库，目前还有超过 15 个专门构建的数据库引擎。以下是适用于特定于 MES 的微服务的数据库示例。

关系数据库

某些 MES 微服务必须维护数据完整性、原子性、一致性、隔离性和耐久性 (ACID) 合规性，以及事务数据的复杂关系。例如，可能需要微服务来存储工单与产品 BOMs、供应商等之间的复杂关系。关系数据库最适合此类服务。[Amazon Relational Database Service \(Amazon RDS \)](#) 可以满足所有这些需求。它是一系列托管服务，可帮助您在云中设置、操作和扩展数据库。[它提供八种流行的数据库引擎可供选择 \(兼容亚马逊 Aurora PostgreSQL 的版本、兼容亚马逊 Aurora MySQL 的版本、适用于 PostgreSQL 的亚马逊 RDS、适用于 MySQL 的亚马逊 RDS、适用于 SQL Server 的亚马逊 RDS、适用于 Db 2 的亚马逊 RDS \)](#)。

键值，NoSQL 数据库

某些 MES 微服务与来自机器或设备的非结构化数据进行交互。例如，在现场进行的各种质量测试的测试结果可以采用多种格式，并且可能包括不同类型的数据，例如通过/未通过值、数值或文本。有些甚至可能有参数来支持材料分析中的内容或成分测试。在这种情况下，关系数据库的僵化结构可能不是最佳选择，NoSQL 数据库可能更合适。[Amazon DynamoDB](#) 是一个完全托管的无服务器键值的 NoSQL 数据库，专为运行任何规模的高性能应用程序而设计。

时间序列数据库

机器和传感器在制造过程中会生成大量数据，以测量随时间推移而变化的值，例如过程参数、温度、压力等。对于此类时间序列数据，每个数据点都由一个时间戳、一个或多个属性以及一个随时间变化的值组成。企业可以使用这些数据来深入了解资产或流程的性能和运行状况，检测异常情况并识别优化机会。企业必须经济实惠地实时收集这些数据并对其进行高效存储，这有助于组织和分析数据。传统的单片MES不能有效地使用时间序列数据。时间序列数据的收集和存储主要是历史学家和其他较低级别的OT系统的职能。微服务和云提供了使用时间序列数据并将其与其他情境化数据相结合的机会，以解锁宝贵的见解和流程改进。[Amazon Timestream](#) 是一项快速、可扩展且无服务器的时间序列数据库服务，它可以更轻松地每天存储和分析数万亿个事件，速度最高可提高 1,000 倍，成本仅为关系数据库的十分之一。另一种处理时间序列数据的托管服务是[AWS IoT SiteWise](#)。这是一项托管服务，使工业企业能够在多个工业设施中收集、存储、组织和可视化成千上万的传感器数据流。AWS IoT SiteWise

包括在位于设施现场的网关设备上运行的软件，持续从历史学家或专业的工业服务器收集数据，然后将其发送到云端。

云存储

MES 处理许多非结构化数据格式，例如工程图纸、机器规格、工作说明、产品和车间图像、培训视频、音频文件、数据库备份文件、分层文件夹和文件结构中的数据等。传统上，企业将这些类型的数据存储在 MES 应用程序层中。云存储解决方案提供业界领先的可扩展性、数据可用性、安全性和性能。云存储的显著优势是几乎无限的可扩展性、更高的数据弹性和可用性以及更低的存储成本。通过使用云存储服务为工业数据湖、分析和机器学习应用程序提供支持，企业还可以更好地使用 MES 数据。AWS 提供诸如[亚马逊简单存储服务 \(Amazon S3\)](#)、[亚马逊弹性区块存储 \(Amazon EBS\)](#)、[亚马逊弹性文件系统 \(Amazon EFS\)](#)、[亚马逊弹性文件系统 \(亚马逊 EFS\)](#) 和[亚马逊等存储服务](#)。FSx 为微服务选择正确的存储选项取决于您对延迟和速度、操作系统、可扩展性、成本、使用量和数据类型的要求。从架构的角度来看，您也可以为同一个微服务选择多个选项。

用户界面

MES 用户组可以多种多样。他们可能包括收货和仓库文员、物料搬运工、机器操作员、维护人员、生产调度员和生产经理。这些用户及其任务会影响 MES 的用户界面 (UI) 设计。例如，在办公室办公桌上工作的职员的用户界面与在车间使用手持设备的物料处理人员的用户界面不同。这种多样的用户界面要求也决定了底层技术的选择。在基于微服务 UIs 的 MES 架构中，它们会经常升级，并且会经历自己的生命周期阶段，例如开发、交付、测试和监控以及用户参与。AWS 为[前端 Web 和移动 UI 提供了一系列广泛的服务，可支持 UI 生命周期阶段的挑战](#)。用户界面生命周期中使用的两项重要 AWS 服务是：

- [AWS Amplify](#) 提供了一组工具，用于前端 Web 或移动应用程序中的数据存储、身份验证、文件存储、应用程序托管，甚至是 AI 或 ML 功能。你可以为你的 iOS、Android、Flutter、Web 或 React Native 应用程序创建具有实时和离线功能的跨平台后端。
- [AWS AppSync](#) 创建无服务器 GraphQL publish/subscribe 和 (pub/sub APIs)，通过单个端点来安全地查询、更新或发布数据，从而简化应用程序开发。

确定 MES 中微服务的集成方法

在基于微服务的 MES 中，service-to-service 通信对于交换数据、共享信息和确保无缝操作至关重要。MES 微服务可以定期交换有关特定事件的数据。例如，用户可能在生产确认交易期间提供生产数量。这样的事务可以在后台启动多项事务，例如将信息发送到 ERP、捕获机器的运行时间、捕获有关产品的质量信息以及报告工时。不同的微服务可能负责这些任务，但单个事件会通过一个微服务启动所有微服务。

此外，MES 还与外部系统集成，以优化制造运营、连接 end-to-end 数字线程和过程自动化。在构建基于微服务的 MES 时，必须决定处理与内部和外部服务集成的策略。

以下功能模式为根据所需的通信类型选择正确技术提供了指导。

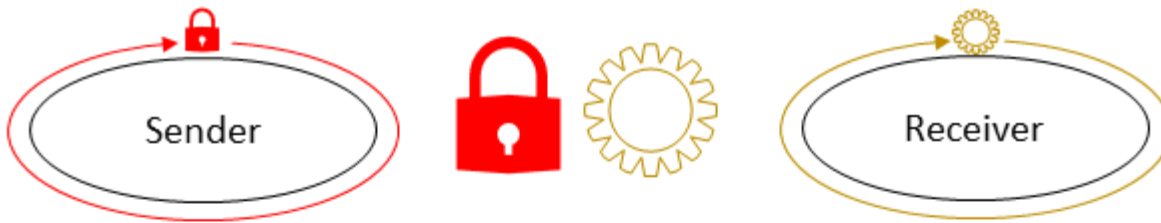
同步通信

在同步通信模式中，调用服务会被阻止，直到它收到来自端点的响应。终端节点通常可以调用其他服务进行额外处理。对于延迟敏感型交易，MES 需要同步通信。例如，假设一条连续生产线，其中一个用户完成订单的操作。下一位用户希望看到该订单立即送达，以便进行下一次操作。此类交易的任何延迟都可能对产品的周期时间和工厂性能产生负面影响 KPIs，并可能导致等待时间增加和资源利用不足。

Stage 1: The sender sends a request to the receiver.



Stage 2: The sender remains blocked while the receiver is processing.



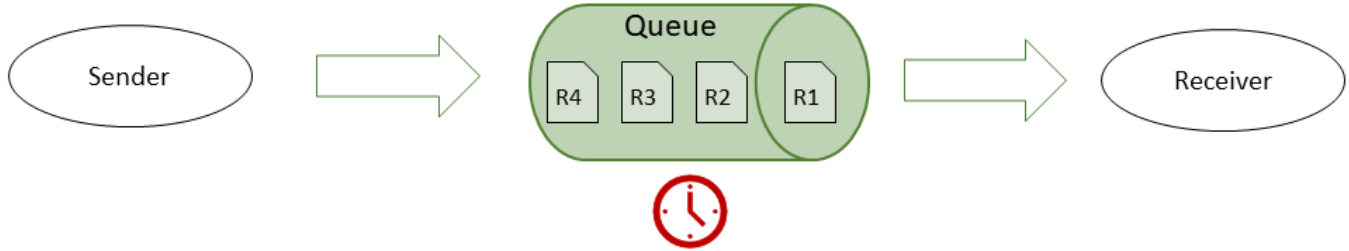
Stage 3: The sender is unblocked when the receiver sends a response.



异步通信

在这种通信模式中，调用者不会等待来自端点或其他服务的响应。当MES可以容忍延迟而不会对业务交易产生负面影响时，它就会采用这种模式。例如，当用户使用计算机完成操作时，您可能需要向维护微服务报告该计算机的运行时间。这种通信可以是异步的，因为更新运行时间不会立即启动事件或影响操作的完成。

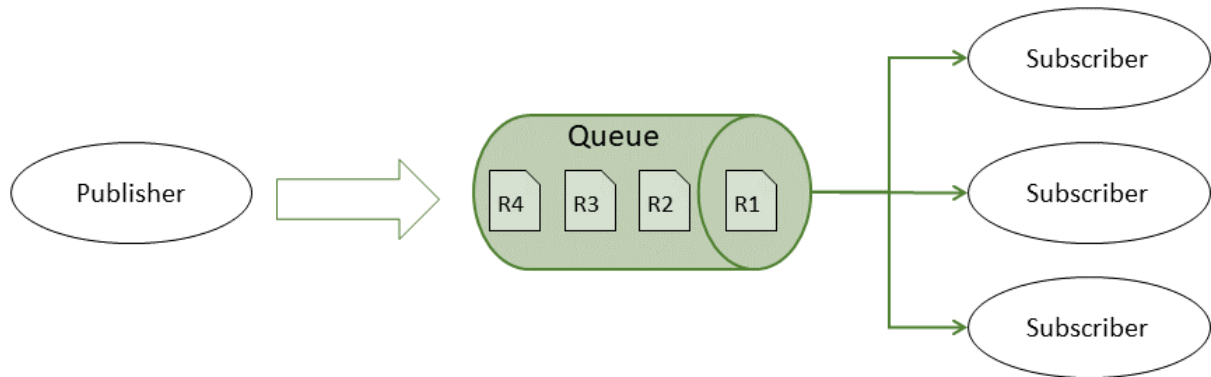
The sender sends a request to the queue and doesn't get blocked while the receiver is processing the request.



发布/订阅模式

publish-subscribe (pub/sub) pattern further extends asynchronous communications. Managing interdependent communications can become challenging as the MES matures and the number of microservices grows. You might not want to change a caller service every time you add a new service that has to listen to it. The pub/sub 模式通过在多个微服务之间实现异步通信而无需紧密耦合来解决这个问题。在这种模式下，微服务将事件消息发布到订阅者微服务可以收听的频道。因此，当您添加新服务时，您无需更改发布服务即可订阅该频道。例如，生产报告或操作完成的事务可能会更新多个日志和事务历史记录。无需在为机器、人工、库存、外部系统等添加新的日志服务时修改这些事务，而是可以为每项新服务订阅原始事务的消息并单独处理。

The sender sends a request to the queue. More than one receiver can subscribe to the queue.



混合通信

混合通信模式结合了同步和异步通信模式。

AWS 提供多种[无服务器服务](#)，[这些服务](#)可以通过不同的方式组合以生成所需的通信模式。下表列出了一些突出的 AWS 服务及其主要功能。

AWS 服务	描述	支持图案		
		同步	异步	Pub/Sub
Amazon API Gateway	使微服务能够访问来自其他微服务的数据、业务逻辑或功能。API Gateway 接受并处理所有三种通信模式的并发 API 调用。	✓	✓	✓
AWS Lambda	提供无服务器、事件驱动的计算功能，无需管理服务器即可运行代码。企业可以使用 Lambda 在数据库和存储 AWS 服务等其他服务之间解耦数据、处理和传递数据。	✓	✓	✓
Amazon Simple Notification Service (Amazon SNS)	支持 application-to-application (A2A) 和 application-to-person (A2P) 消息传递。A2A 在分布式系统、微服务和无服务器应用程序之间提供高吞吐量、基于推送的消息传递。A2P 功能允		✓	✓

AWS 服务	描述	支持图案		
		同步	异步	Pub/Sub
	许您通过短信、推送通知和电子邮件向他人发送消息。			
Amazon Simple Queue Service (Amazon SQS)	允许您在任何音量的软件组件之间发送、存储和接收消息，而不会丢失消息或要求其他服务可用。		✓	✓
Amazon EventBridge	无需编写代码，即可实时访问由微服务或微服务中的 AWS 服务中的数据变化引起的事件。然后，您可以接收、过滤、转换、路由此事件并将其传送到目标。		✓	✓

AWS 服务	描述	支持图案		
		同步	异步	Pub/Sub
Amazon MQ	托管消息代理服务，可简化上 AWS 消息代理的设置、操作和管理。消息代理允许软件系统（通常在不同的平台上使用不同的编程语言）进行通信和交换信息。			✓

有关更多信息，请参阅 AWS 规范指南网站上的[使用 AWS 无服务器服务集成微服务](#)。

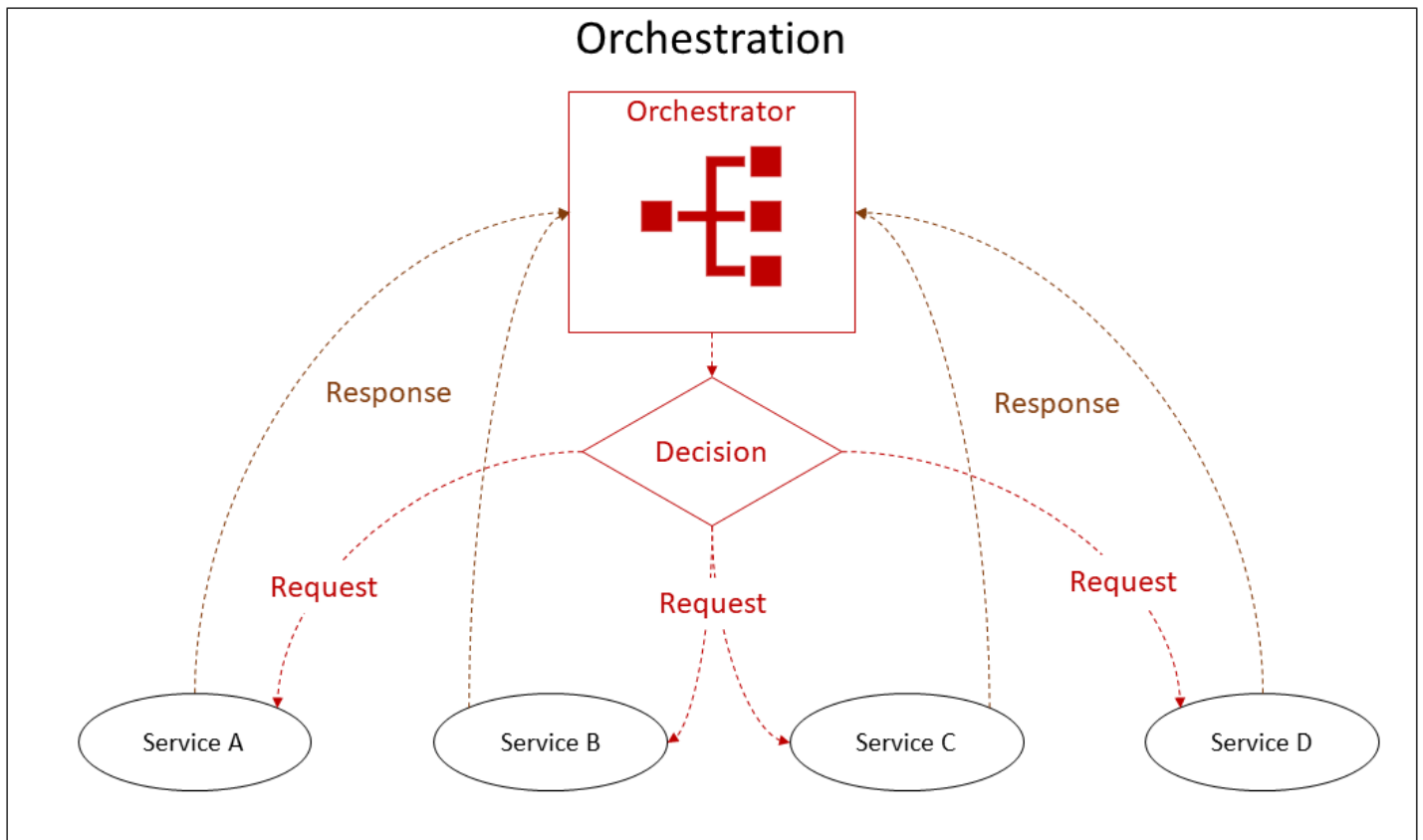
使用云原生技术管理、编排和监控 MES 的微服务

在为各个微服务设计架构后，应重点确保所有微服务都能无缝运行。基于微服务的 MES 是一个敏捷、不断发展的系统，它具有动态的分布式组件，例如容器镜像、数据库 APIs、对象存储和队列。这种不断的变化在编排、监控和管理这些分布式组件方面带来了另一组架构挑战。

编排

MES 中的某些交易可能涉及来自生产、质量、库存、维护和其他领域的多种微服务，用于报告操作完成、根据采购订单接收库存或完成质量检查等任务。这些交易包括多个子交易，需要编排。编排代码不应放在特定的微服务中，而应出现在更高级别的控制平面上。

为了简化如此复杂的编排，AWS 提供 [AWS Step Functions](#)。这项完全托管的服务使用可视化工作流程，可以更轻松地协调分布式应用程序和微服务的组件。它提供了一个图形控制台，用于将应用程序的组件按一系列步骤进行排列和可视化，如下图所示。可视化排列使构建和运行多步骤应用程序变得更加容易。



审核

由于不断变化和演变，基于微服务的 MES 架构是动态的。Organizations 必须强制执行安全政策和其他企业政策以实现合规和监管。要确保诸如MES之类的系统内有许多用户、多个微服务以及每个微服务中的许多资源，则需要对所有用户操作和微服务交互进行可视化。

AWS 提供以下服务来解决审计和监控方面的挑战：

- [AWS CloudTrail](#)通过跟踪用户活动和 API 使用情况，实现审计、安全监控和操作故障排除。CloudTrail 日志会持续监控和保留与整个 AWS 基础架构中的操作相关的账户活动，并允许您控制存储、分析和修复操作。
- [Amazon CloudWatch](#) 是一项针对 AWS Cloud 资源和应用程序的 AWS 监控服务。您可以使用 CloudWatch来获得对资源利用率、应用程序性能和操作健康状况的全系统可见性。它可以收集和跟踪指标、收集和监控日志文件以及设置警报。
- [AWS Config](#)为安全和治理提供资源清单、配置历史记录和配置变更通知。您可以使用 AWS Config 来发现现有 AWS 资源、记录第三方资源的配置、导出包含所有配置详细信息的完整资源清单，以及随时确定资源的配置方式。
- [适用于 Prometheus 的亚马逊托管服务是一项针对](#)指标的无服务器监控服务，它与开源 Prometheus 数据模型和查询语言兼容。它可以监控本地以及混合云和多云环境中的容器工作负载并生成警报。

AWS

MES 中的弹性

弹性是指MES系统能够从基础设施或服务中断中恢复，动态获取计算资源以满足需求，并缓解配置错误或临时网络问题等中断。弹性是 [Well-Architected Framework 的可靠性支柱所依赖的主要因素](#)。

弹性可以分为两个主要因素：可用性和灾难恢复。这两个领域都依赖一些相同的最佳实践，例如监控故障、部署到多个位置以及自动故障转移。但是，可用性侧重于 MES 微服务的组件，而灾难恢复则侧重于整个微服务甚至整个 MES 系统的离散副本。

可用性

我们将可用性定义为微服务可供使用的时间百分比，如以下公式所示。此百分比是在一段时间内计算的，例如一个月、一年或之后三年。

$$A = \frac{\textit{uptime}}{\textit{uptime} + \textit{downtime}}$$

这个公式需要了解制造和设备维护中常见的三个指标：

- 平均故障间隔时间 (MTBF)：从微服务开始常规操作到随后出现故障之间的平均时间。
- 平均检测时间 (MTTD)：从发生故障到开始修复操作之间的平均时间。
- 平均修复时间 (MTTR)：从因子系统故障而导致微服务不可用到其修复或恢复服务之间的平均时间。MTTD 是 MTTR 的一个子集。

下图说明了这些可用性指标。



弹性、高度可用的 MES 旨在降低 MTTR 和 MTTD 并提高 MTBF。尽管理想的设计可以消除故障，但它并不现实。传统的整体式 MES 故障很难被发现，需要更长的时间才能修复。现代、云原生 MES 可

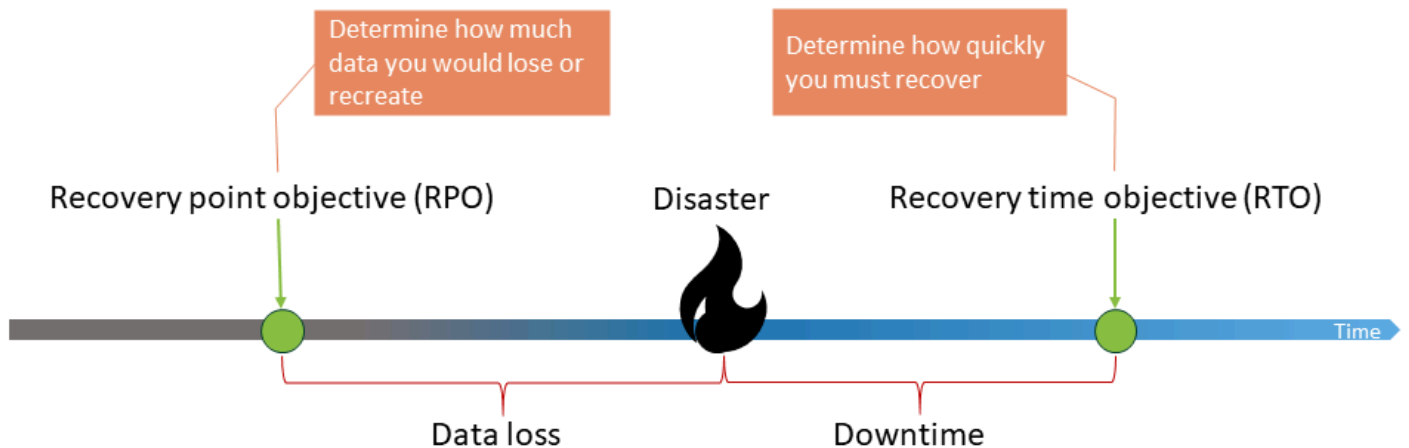
通过多可用区部署实现更快的检测、快速修复和业务连续性。有关具有相关 AWS 服务的高可用性现代系统的最佳实践，请参阅白皮书《[可用性及其他：了解和提高分布式系统的弹性](#)》[AWS](#)。

灾难恢复

灾难恢复是指为与技术相关的灾难（例如重大硬件或软件故障）做好准备并从中恢复的过程。阻止微服务（MES）在其主要部署位置实现其业务目标的事件被视为灾难。灾难恢复不同于可用性，它由以下两个指标来衡量：

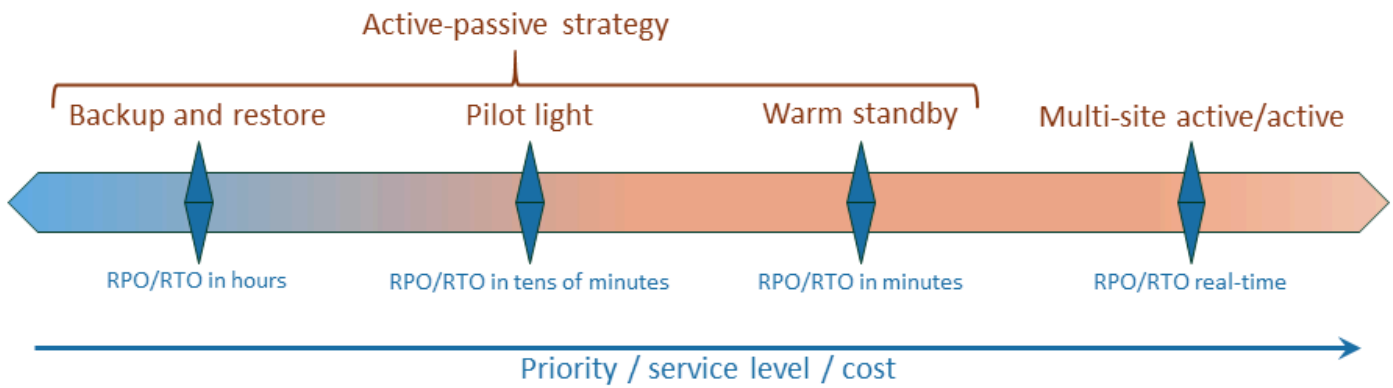
- 恢复时间目标 (RTO)：微服务中断和微服务恢复之间的可接受延迟。RTO 决定了当服务不可用时，什么时间段被视为可接受的时间窗口。
- 恢复点目标 (RPO)：自上次数据恢复点以来的最大可接受时间。RPO 决定了在最后一个恢复点和微服务中断之间，哪些数据丢失被认为是可接受的。

下图说明了这些灾难恢复指标。



下图描述了不同的灾难恢复策略。

Disaster recovery strategies



您可以在 Well-Architecte AWS d Framework 指南 [《工作负载灾难恢复：云端恢复》](#) 中找到 AWS 有关实施这些策略的详细指南。

结论

基于微服务的架构有助于克服传统的单片 MES 带来的限制。构建基于微服务的应用程序会遇到一些挑战，例如架构复杂性和运营开销。为了充分发挥基于微服务的 MES 的潜力，我们建议您探讨以下问题：

- 您想要解决的当前架构的局限性是什么？
- 您是否有足够的专业知识来做出业务和架构决策？
- 您是否有或计划建立治理结构？
- 您是否有用于测试和部署的自动化？
- 你有变更管理和培训计划吗？

AWS [现代化加速](#)、[评估](#)、[研讨会](#)、[解决方案指导](#)和[沉浸式体验日](#)等资源使制造商能够从其现代化工作中获得最大的收益。

参考信息

AWS 服务

- [AWS Amplify](#) (全栈应用程序开发)
- [亚马逊 API Gateway](#) (API 管理)
- [AWS AppSync](#) (无服务器 GraphQL APIs)
- [AWS CloudTrail](#) (API 日志)
- [亚马逊 CloudWatch](#) (APM 工具)
- [AWS Config](#) (托管配置服务)
- [亚马逊 DynamoDB](#) (非关系数据库)
- [亚马逊 EBS](#) (云块存储)
- [Amazon EC2](#) (可调整大小的计算网络服务)
- [亚马逊 EFS](#) (共享文件存储)
- [亚马逊 EventBridge](#) (事件侦听器)
- [Amazon FSx](#) (托管文件服务器)
- [AWS IoT Core](#) (托管物联网云平台)
- [AWS IoT Greengrass](#) (开源边缘运行时和云服务)
- [AWS IoT SiteWise](#) (IIoT 数据收集、存储和监控)
- [AWS Lambda](#) (无服务器、事件驱动的计算)
- [适用于 Prometheus 的亚马逊托管服务](#) (托管容器监控)
- [亚马逊 MQ](#) (消息代理)
- [亚马逊 RDS](#) (关系数据库)
- [亚马逊 S3](#) (云对象存储)
- [亚马逊 SageMaker AI](#) (机器学习建模)
- [亚马逊 SNS](#) (推送通知)
- [亚马逊 SQS](#) (消息队列)
- [AWS Step Functions](#) (工作流程编排)

AWS 服务家庭

- [开启 AI/ML AWS](#)
- [分析服务开启 AWS](#)
- [集装箱位于 AWS](#)
- [数据库开启 AWS](#)
- [边缘服务已开启 AWS](#)
- [开启前端 Web 和移动版 AWS](#)
- [物联网服务开启 AWS](#)
- [无服务器开启 AWS](#)

其他 AWS 资源

- [AWS 评估工具](#)
- [AWS 物联网能力合作伙伴](#)
- [AWS 迁移加速计划](#)
- [AWS 解决方案库](#)
- [AWS 以解决方案为重点的沉浸式体验日](#)
- [AWS Well-Architected Framework](#)
- [AWS 工作坊](#)
- [AWS 云计算概念中心](#)
- 出版物：
 - [可用性及其他：了解和提高分布式系统的弹性 AWS](#) (AWS 白皮书)
 - [工作负载的灾难恢复 AWS：云端恢复](#) (AWS 白皮书)
 - [工业数据结构](#) (AWS 合作伙伴解决方案和指南)
 - [使用 AWS 无服务器服务集成微服务](#) (AWS 规范性指南)
 - [亚马逊 EKS 上的负载均衡](#) (亚马逊 EKS 文档)
 - [AWS Outposts 使用时运行 AWS Lambda 函数 AWS IoT Greengrass](#) (AWS 博客文章)

作者和贡献者

以下人员 AWS 撰写了本指南并为其做出了贡献。

作者：

- Ravi Soni，首席工业制造解决方案专家
- 史蒂夫·布莱克威尔，全球制造业技术负责人
- Nishant Saini，首席合作伙伴解决方案架构师
- Pratik Yeole，解决方案架构师

贡献者：

- Darpan Parikh，可组合应用程序解决方案负责人
- Jan Metzner，首席工业制造解决方案专家
- Bhavisha Dawada，高级解决方案架构师

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
更新	更新了数据和分析部分中的架构图和 说明 。	2024 年 4 月 2 日
初次发布	—	2024 年 2 月 23 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。此帐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

laC

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 文档中的[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。您可以使用 [Amazon S CloudWatch ynthetic](#) 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。