



切断网络流量的最佳实践 AWS

# AWS 规范性指导



# AWS 规范性指导: 切断网络流量的最佳实践 AWS

Copyright © 2024 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
割接阶段概述 .....	2
沟通与治理 .....	3
沟通与治理策略 .....	3
沟通与治理渠道 .....	3
割接前阶段 .....	5
制定割接计划 .....	5
创建割接运行手册 .....	6
割接阶段 .....	7
割接方法 .....	7
一种II-at-once 方法 .....	7
分阶段割接 ( 金丝雀部署 ) .....	7
选择正确的方法 .....	7
何时执行割接 .....	9
割接前可以测试的内容 .....	9
割接前无法测试的内容 .....	9
运营准备情况审查 .....	9
回滚 .....	9
在割接过程中或没有新数据的情况下回滚 .....	10
使用更改后的数据回滚 .....	10
割接后阶段 .....	11
设置监控控制面板 .....	11
了解和沟通成功 .....	11
定义保证期 .....	11
结论 .....	12
资源 .....	13
文档历史记录 .....	14
术语表 .....	15
# .....	15
A .....	15
B .....	18
C .....	19
D .....	21

---

E .....	24
F .....	26
G .....	27
H .....	28
I .....	29
L .....	31
M .....	31
O .....	34
P .....	36
Q .....	38
R .....	38
S .....	41
T .....	43
U .....	45
V .....	45
W .....	45
Z .....	46
.....	xlvii

# 在迁移过程中将网络流量割接到 AWS 的最佳实践

Damien Renner、Ernest Asare、Martin Fluck、Khurram Mahmood 和 Mark Willis , Amazon Web Services ( AWS )

2022 年 10 月 ( [文档历史记录](#) )

割接阶段是云迁移中最关键的阶段之一。在割接时，将网络流量从源系统重定向到 Amazon Web Services ( AWS ) 上托管的目标系统。割接期间，系统和用户很可能会受到一定程度的干扰。为了最大限度地减少中断并降低割接风险，我们建议您制定符合本指南中介绍的最佳实践的割接计划。本指南适用于迁移顾问、应用程序架构师、项目经理、应用程序所有者以及实施或领导向 AWS 云迁移的任何其他角色。

## 目标业务成果

本指南的最终目的是帮助您最大限度地减少中断，降低云迁移的割接阶段相关的风险。本指南可帮助您通过实现以下目标业务成果来达成这一总体目标：

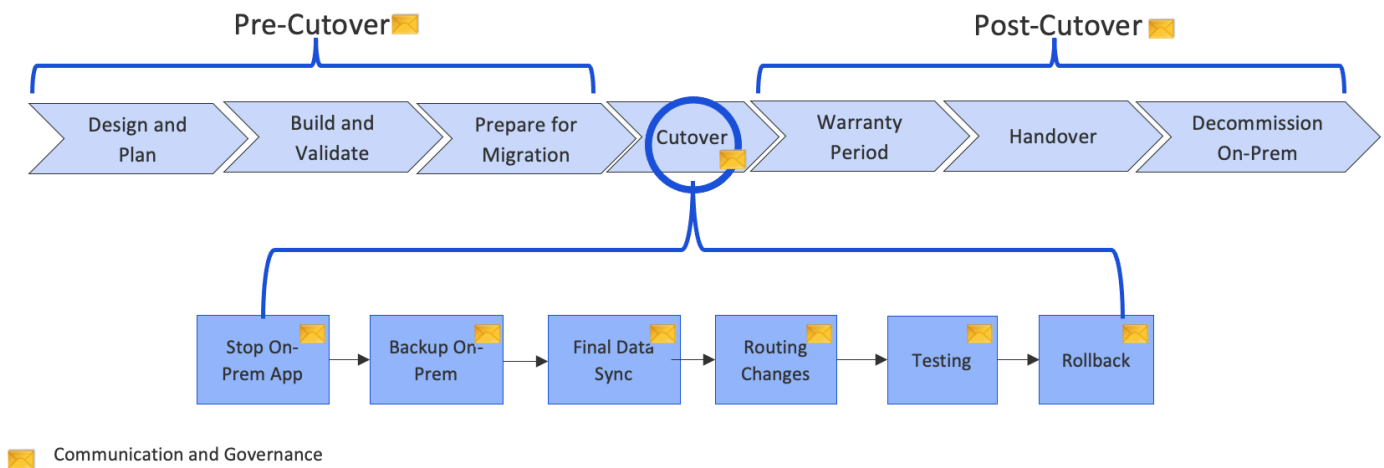
- 制定最佳割接流程
- 制定有关割接的沟通与治理策略
- 了解不同的割接方案，根据您的要求选择最有效的方案

## 割接阶段概述

在迁移过程的割接阶段，您需要将网络流量从现有端点转移到 AWS 云中新部署的资源。例如，您可以更新域名系统（DNS）记录，将其添加到新预置的 Elastic Load Balancer 的端点。对于大多数应用程序，在迁移的这一阶段，您可能必须割接多个集成点（例如，割接应用程序和数据库层以迁移两层应用程序）。

与割接相关的时间、复杂性和风险取决于您的业务需求、技术堆栈、风险偏好、应用程序架构和其他因素。但是，一般来说，随着割接时间窗口的缩短，割接的复杂性也会增加。割接所需的停机时间窗口与迁移的应用程序的重要性之间有很强的相关性。通常，业务关键型应用程序的风险会增加，因为它们需要的停机时间更短。风险的增加反过来又导致需要进行更广泛的规划，以管理割接的复杂性并降低风险。

割接阶段是[迁移过程](#)的最后阶段。下图显示了割接阶段在迁移过程中的位置。如图所示，沟通与治理（将在本指南下一节中介绍）在割接阶段起着重要作用。



# 沟通与治理计划

## 沟通与治理策略

您的沟通内容和沟通方式可能是割接成功的关键。为了确保成功，我们建议您制定并实施一项沟通与治理策略，提供有关沟通工具的明确指导，定义所有权角色，并为每项割接任务指定沟通渠道。我们还建议迁移团队对沟通策略进行记录和管理，并作为项目管理的一部分，将其与迁移升级路径集成。

## 沟通与治理渠道

建立沟通渠道时，务必定义目标受众，确定哪种沟通方式最有效。例如，如果组织中的大多数人不主动监控电子邮件，那么即时消息可能是首选方法。我们建议您在所有相关利益相关者之间建立一个渠道。理想情况下，在割接前、割接中和割接后，渠道都是开放的，并且您的利益相关者对您设置的沟通协议负责。

在某些情况下，可能需要对沟通渠道进行分组（例如，为业务部门或密切相关的应用程序创建 Slack 组）。我们建议您将管理沟通和消息的责任分配给沟通负责人或迁移经理。

### 注意

在进行任何割接之前，最好是获得迁移团队、频波所有者和应用程序所有者的支持。务必充分告知所有应用程序利益相关者割接情况。

下表列出了各种沟通渠道的高级沟通协议示例。

渠道	方法	所有者	频率
高级利益相关者（内部）	电子邮件	项目经理或割接经理	重大事件（如开始和结束日期）或确认的 P1 问题
项目/技术团队	Slack、Chime 或 Microsoft Teams 聊天室	所有负责发布活动最新信息的利益相关者 项目经理或割接经理	所有任务的开始和结束日期及延迟

渠道	方法	所有者	频率
数据中心监控团队/L1 运维	简报会和电子邮件	项目经理或割接经理	开始参与 ( 四周后 ) ，每个已完成阶段的更新
客户的高级利益相关者	简报会和电子邮件	项目经理或割接经理	开始参与 ( 四周后 ) ，每个已完成阶段的更新
最终客户	电子邮件	项目经理或应用程序所有者	割接开始和结束



## 割接前阶段

您的割接选项取决于多种因素，包括您使用的迁移模式，以及源环境和目标环境的工具、技术和配置。例如，将 VMware 环境迁移到 AWS 与构建全新堆栈的割接选项是不同的。

从根本上说，所有割接都是为了实现同样的结果 - 将工作负载转移到另一个位置。了解服务器或应用程序移至何处非常重要，这样您就可以配置上游和下游依赖关系，以便连接到新的端点。

根据服务的性质，现有数据库及其与迁移到新位置的其他系统的连接可能非常复杂。同步数据库以确保两个位置的数据一致性可能会增加复杂性。此外，多项服务要求您要么调整网络配置，要么同时割接多项服务。

## 制定割接计划

为了降低延迟、返工、意外中断、数据丢失、性能问题和用户体验不佳的风险，您可以制定割接计划，创建割接工作手册，进行割接演练。我们建议您的割接计划包含应急计划和风险缓解策略，以应对割接失败的情况。确保记录回滚程序，作为割接计划的一部分。

我们建议您在割接前计划流程中分析和评测以下关键项目：

- 所需的停机时间，包括：
  - 超过分配的停机时间窗口对业务的影响（例如对收入或信任的影响）
  - 在发生不可预见的事件时应对“向前修正”活动的应急措施
  - 发生故障时的回滚时间
- 用户连接
- 应用程序和系统依赖关系
- 基础设施变更
- 运营变更
- 测试计划，包括：
  - 功能测试
  - 通过基线测试和割接后测试进行性能测试
  - 相关应用程序、系统和服务的集成和连接测试
  - 割接和保证期内的策略测试
- 工具和策略相关的部署变更

最后，考虑将以下内容纳入您的计划：

- 任务定义和持续时间
- 任务所有者
- 阶段和任务顺序

## 创建割接运行手册

我们建议您创建割接运行手册，供迁移团队用来跟踪活动，包括各自计划的开始和结束时间、顺序和所有者。您可以使用 RACI 矩阵将责任分配给相应的团队成员。割接运行手册具有以下好处：

- 通过主动协调不同的利益相关者来降低风险
- 通过重复使用将应用程序迁移到 AWS 云的成熟机制来提高工作效率
- 通过使用结构化模板进行割接规划，加快应用程序迁移速度
- 提供一个框架，以便遵循必要的流程
- 提供一种机制，将流程的变更告知所有团队

# 割接阶段

迁移存储数据的组件时，您需要考虑数据一致性是否是关键要求。如果是，则可能需要在开始割接过程之前锁定源环境（如数据库锁）。锁定数据库可确保源环境中没有新事务。但是，锁定可能需要更长的停机时间。

割接通常包括以下几个阶段：

- 摄取冻结 - 将本地应用程序和数据摄取冻结到数据库。这样可以确保应用程序的本地版本在割接期间不会收到任何新的事务或数据。
- 备份 - 对本地系统进行最终备份。如有必要，您可以在紧急情况下使用该备份进行回滚。
- 数据同步 - 完成本地和目标（云）环境之间的最终数据同步。
- 路由更改 - 将用户路由到云环境（例如，更新 DNS 记录或更改负载均衡器目标）。
- 测试 - 在将迁移标记为完成之前，测试并验证一切是否正常。

## 割接方法

有两种切换方法需要考虑：all-at-once 一种方法或一种分阶段的方法。选择最佳割接方法的关键是了解您的业务需求和技术限制。本部分概述了两种方法。

### 一种 all-at-once 方法

如果你采用 all-at-once 这种方法，那么你只需轻按一下开关即可切断整个解决方案。例如，可以通过更新 DNS 或更换负载均衡器来实现这一点。然后，所有用户和实时流量将立即使用新系统。如果由于潜在的主机名冲突、许可证问题或域身份验证限制而无法让新系统上线，这种方法就非常有用。因为时间至关重要，所以重点在于何时或由谁来请求故障恢复。我们建议您的 all-at-once 方法计划包括广泛的性能测试，并在适用的情况下包括回归测试，以便您可以验证应用程序的功能和非功能特性。

### 分阶段割接（金丝雀部署）

分阶段割接涉及在规定的时间内逐步割接。这种方法包括持续监控和检查，以验证当前系统能否承受负载，以及每个系统组件是否按预期运行。分阶段割接有助于降低潜在割接问题的风险，因为您可以根据反馈调整系统性能。如果发现关键问题，还可以轻松回滚任何更改。

## 选择正确的方法

要选择正确的方法，请确定以下几点：

- 属于同一移动组的相关应用程序和服务
- 可在本地和迁移应用程序之间使用的通用数据来源
- 可将部分负载重定向到不同端点的应用程序和基础设施

如果您的应用程序无法容忍数据源和应用程序服务器之间延迟的增加，则这清楚地表明需要一种 all-at-once 方法。在这种情况下，您可以将所有应用程序资源（服务器和数据库）割接在一起，以免影响性能。

在分阶段转换中，您可以将构成应用程序的服务器和服务的一定比例从整体上拆分出来，然后切换到剩 AWS 余的服务器和服务保留在本地。然后，根据负载均衡或 DNS 策略将客户端流量路由到这两个环境。分阶段割接可帮助您最大限度地减少对用户的影响，使受割接影响的用户数量最少。如果能确定影响，就可以相应地调整服务器和服务的百分比。但是，分阶段割接方法依赖于底层应用程序支持该方法的能力。我们建议您问自己以下问题：

- 应用程序是否有多个层（前端、后端、数据库），由可拆分的弹性服务器对或服务器组组成？
- 应用程序是否通过负载均衡器进行访问？负载均衡器是否支持将流量路由到 AWS 网络和本地网络？
- 应用程序服务器能否迁移以 AWS 容忍数据库或其他后端依赖项的延迟。如果将数据库切换到 AWS，则留在本地的服务器或服务能否继续按要求运行和执行？

与回滚功能相比，能够将一定比例的用户发送到新迁移的服务器，AWS 同时保持现有的本地容量，这具有关键优势。all-at-once 由于迁移服务器和现有服务器相混合，服务于应用程序，且负载分散在不同服务器之间，因此一旦出现问题，恢复起来既快又简单。在大多数情况下，只需要更改负载均衡器、DNS 规则或策略即可。分阶段转换方法还允许您逐步增加负载 AWS，这使应用程序团队能够评估应用程序的性能，并在转移满负荷之前进行所需的更新或更改。

选择是最好一次切换一个应用程序或一堆依赖的应用程序，还是使用分阶段切换服务器和服务的增量方法，都不太可能成为一个 one-size-fits-all 决定。我们通常看到客户采用以下方法：

- 能够容忍一段停机的开发和测试环境将受益于该 all-at-once 方法的简单性和较低的工作量。
- 相比之下，分阶段方法更复杂、更耗时，但通常对停机时间的要求更低，回滚选项也更快。因此，分阶段方法最常用于业务关键型生产工作负载。

我们建议您在割接更改窗口前花点时间了解您的源系统。通过在早期规划阶段投入更多时间，您可以支持各种流程，如割接准备和迁移后验证。迁移到服务器时，客户可以更改其服务器的 IP 地址 AWS。在这种情况下，要避免的关键因素是在应用程序中使用硬编码的 IP 地址。我们建议您全面审视您的源环境，因为它可能同时具有上游或下游依赖项。例如，您有可能对连接到您迁移的服务的其他系统造成

问题。值得考虑的是，在开始割接之前，将所有连接转为使用完全限定域名 ( FQDN ) 或 DNS 记录是否有价值。

## 何时执行割接

一般来说，用户最少的时候是进行割接的最佳时机，因为对业务的影响最小。但是，这需要与割接窗口期间支持团队的可用性保持均衡。您需要支持团队，帮助其排查和解决潜在问题。同样重要的是，要考虑割接的日期和时间以及利益相关者的准备情况。如果您的任何利益相关者没有做好准备，没有在预定日期和时间在场，那么割接就可能会面临延误的风险。

## 割接前可以测试的内容

如果迁移方法允许，最佳做法是在割接窗口之前执行功能和非功能测试。例如，您可以利用负载测试工具来验证新环境是否在割接窗口前进行了适当配置。通常，此阶段的测试不会造成中断，因为 AWS 环境不提供实时流量。

## 割接前无法测试的内容

由于与其他应用程序的依赖关系，可能无法测试生产中发生的所有情况。在这种情况下，应记录潜在风险、计划如何识别风险以及测试失败时团队将采取的相应缓解措施。

## 运营准备情况审查

在将应用程序移交给之前 AWS，我们建议您进行操作准备情况审查。在此阶段，您需要评估测试的完整性，验证团队监控和获取警报的能力，并确认利益相关者了解如何支持和维护工作负载。这可能需要与业务和技术团队合作。有关运营准备情况的更多信息，请参阅文档中 Well-Architect [AWS ed 的 AWS Well-Architected Tool 框架](#) 卓越运营支柱。AWS

## 回滚

在某些条件下，可能需要进行迁移回滚。为了应对可能出现的回滚，我们建议您制定一个包括以下内容的回滚计划：

- 定义的检查点，如果满足某些预定义的标准，则在割接期间启动回滚
- 管理回滚和处理数据的回滚策略
- 一个联络点，由其决定是向前修正还是向后回滚迁移

## 在割接过程中或没有新数据的情况下回滚

如果您和您的利益相关者决定在不更改任何数据的情况下执行回滚，则回滚方法会很简单，例如，恢复本地实例，然后通过修改负载均衡器或 DNS 配置来重定向流量。

### 使用更改后的数据回滚

如果在成功割接后启动了回滚，并且您的应用程序已收到新的事务和数据，则您可能需要将数据从云环境恢复到本地环境。在这种情况下，我们建议您考虑以下方法：

- 故障转发方法- 由于迁移后的数据库成为主数据库，因此您的本地数据库在切换后可能会过时。  
AWS 您可以使用 [AWS Database Migration Service \(AWS DMS\)](#) 来设置故障转发数据库，该数据库会将数据复制到新的本地数据库。如果出现任何问题，AWS DMS 会将您的应用程序回滚到指定的故障转发数据库，而不是陈旧的本地数据库。
- 双写策略 - 在这种情况下，您的应用程序逻辑必须同时允许写入新旧数据库。
- 本机备份和还原 - 要评估还原所需的时间，可在割接前阶段使用下层环境（即非生产环境）执行备份和还原测试。

## 割接后阶段

### 设置监控控制面板

在将应用程序迁移到云后，我们建议您与利益相关者合作，确定应监控哪些信息或参数并将其发布到监控控制面板。通常，控制面板会监控运营所需的关键服务水平协议 ( SLA ) 指标和服务级别指标 ( SLI )。您可以监控可达性、可用性、安全性、流量模式和其他相关数据点。监控这些数据不仅可以验证您的应用程序是否按预期运行，还将显示您的应用程序是否达到预期质量。

### 了解和沟通成功

在成功完成工作负载迁移后，我们建议将迁移成功告知相关利益相关者。您的利益相关者可能包括单线程负责人、赞助商、最终用户、工作负载应用程序所有者以及任何其他相关的应用程序所有者。

此外，任何有助于改进未来应用程序迁移过程的经验教训都值得考虑。您可以使用迁移记分卡来衡量迁移成功率，并在各个应用和业务部门沟通整体迁移进度。这项工作可以促进团队寻找机会相互学习，最大限度地降低迁移风险，并采用最佳实践。

### 定义保证期

迁移项目通常都有一个保证期，在保证期内，如果在预定义的时间窗口 ( 通常为一天到一周 ) 内出现问题，迁移团队将提供支持。务必根据要迁移的应用程序商定一个合适的时间范围。例如，如果您的应用程序按季度批量计划运行，一周的保证期可能不够。此外，我们建议您的迁移团队利用保证期来验证业务连续性和灾难恢复元素是否按预期配置和运行。

## 结论

割接是一项复杂的活动，涉及的不仅仅是将流量重新路由到新系统。这种复杂性取决于许多因素，包括业务关键性、上游和下游系统的数量、用户数量等。您的团队必须了解割接阶段有关人员、流程和技术的所有复杂性。基于这些复杂性，您可以为迁移的割接前、割接中和割接后的活动确定、制定和采用沟通与治理计划。最后，我们建议您利用现有的本地系统知识来估算割接的持续时间。从割接的预期结果倒推，这可能是一项有用的工作。



## 资源

- [AWS WA 工具](#)
- [使用 AWS DMS 从迁移回滚](#)
- [AWS Database Migration Service](#)

# 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
<a href="#">初次发布</a>	—	2022 年 10 月 31 日

# AWS Prescriptive Guidance 术语表

以下是 AWS Prescriptive Guidance 提供的策略、指南和模式中常用的术语。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：

- **重构/重新架构** - 充分利用云原生功能来提高敏捷性、性能和可扩展性，以迁移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：将本地 Oracle 数据库迁移到 Amazon Aurora PostgreSQL 兼容版。
- **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 云中的 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：将客户关系管理 ( CRM ) 系统迁移到 Salesforce.com。
- **更换主机 ( 直接迁移 )** - 将应用程序迁移到云中，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 云中 EC2 实例上的 Oracle。
- **重新定位 ( 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。此迁移场景特定于 VMware Cloud on AWS，它支持本地环境和 AWS 之间的虚拟机兼容性和工作负载可移植性。在将基础设施迁移到 VMware Cloud on AWS 时，您可以在本地数据中心使用 VMware Cloud Foundation 技术。示例：将托管 Oracle 数据库的管理程序重新定位到 VMware Cloud on 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

### ABAC

请参阅[基于属性的访问控制](#)。

## 抽象服务

参见[托管服务](#)。

## 酸

参见[原子性、一致性、隔离性、耐久性](#)。

## 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。与[主动-被动迁移](#)相比，它更灵活，但需要更多的工作。

## 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。

## 聚合函数

一个 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

参见[人工智能](#)。

## AIOps

参见[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 ( IAM ) 文档中的[有关 AWS 的 ABAC](#)。

## 权威数据源

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

一个 AWS 区域 中的不同位置，用于与其他可用区的故障隔离，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 )

AWS 的指导原则和最佳实践框架，旨在帮助组织制定高效且有效的计划来成功迁移到云。AWSCAF 将指导原则分为六个重点领域（角度）：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，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 Workload Qualification Framework ( AWS WQF )

一种评估数据库迁移工作负载、推荐迁移策略并提供工作量估算的工具。AWSWQF 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

### BCP

参见[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。另请参见[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。

### 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）。

### 破碎的玻璃通道

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 Well [-Architected 指南中的“实施破碎玻璃程序”](#) 指示 AWS 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

参见 [AWS 云采用框架](#)。

## CCoE

参见 [云卓越中心](#)。

## CDC

参见 [变更数据捕获](#)。

## 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弹性。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

查看 [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。

## 客户端加密

在目标 AWS 服务接收数据之前，在本地对数据进行加密。

## 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 云企业战略博客上的 [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。云计算通常与[边缘计算](#)技术相关。

## 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 云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
- 重塑 - 优化产品和服务，在云中创新

Stephen Orban 在 AWS 云企业战略博客上发表的博客文章[云优先之旅和采用阶段](#)中对这些阶段进行了定义。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

参见[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 AWS CodeCommit。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

机器用来在图像中以比肩或高于人类水平的精度辨识人、地点和事物的 AI 领域。它往往用深度学习模型构建，使得可自动从单幅图像或一系列图像提取、分析、分类和理解有用信息。

## 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 规则和修复操作，您可以将其组合起来以自定义合规性和安全性检查。您可以使用 YAML 模板，将合规性包作为单个实体部署到 AWS 账户区域中，或者跨组织部署。有关更多信息，请参阅 AWS Config 文档中的[合规性包](#)。

## 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、提高工作效率、改善代码质量并加快交付速度。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](#)。

# 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 Framework 中安全支柱的一个组成部分。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。

## 数据最少化

仅收集并处理绝对必要数据的原则。在 AWS Cloud 中践行数据最少化可以降低隐私风险、成本和您的分析碳足迹。

## 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

参见[数据库定义语言](#)。

## 深度融合

组合多个深度学习模型进行预测。您可以使用深度融合来获得更准确的预测或估算预测中的不确定性。

## 深度学习

一个 ML 子字段使用多层人工神经网络来识别输入数据和感兴趣的目标变量之间的映射。

## defense-in-depth

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。当您在 AWS 上采用此策略时，您可以在 AWS Organizations 结构的不同层添加多种控制措施，来保护资源。例如，一种 defense-in-depth 方法可以结合多因素身份验证、网络分段和加密。

## 委托管理员

在 AWS Organizations 中，兼容服务可以注册 AWS 成员账户来管理组织的账户，并管理该服务的权限。此账户被称为该服务的委托管理员。有关更多信息和兼容服务列表，请参阅 AWS Organizations 文档中[使用 AWS Organizations 的服务](#)。

## 部署

使应用程序、新功能或代码修复在目标环境中可用的过程。部署涉及在代码库中实现更改，然后在应用程序的环境中构建和运行该代码库。

## 开发环境

参见[环境](#)。

## 侦测性控制

一种安全控制，在事件发生后进行检测、记录日志和发出警报。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

在[星型架构](#)中，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)

您用来最大限度地减少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 [AWS 上工作负载的灾难恢复：AWS Well-Architected Framework](#) 中的云中恢复。

## DML

参见 [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。Eric Evans 在其著作领域驱动设计：软件核心复杂性应对之道 ( Boston: Addison-Wesley Professional, 2003 ) 中介绍了这一概念。有关如何将领域驱动设计与 strangler fig 模式结合使用的信息，请参阅 [使用容器和 Amazon API Gateway 逐步将原有的 Microsoft ASP.NET \( ASMX \) Web 服务现代化](#)。

## DR

参见 [灾难恢复](#)。

### 漂移检测

跟踪与基线配置的偏差。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

参见 [开发价值流映射](#)。

## E

### EDA

参见 [探索性数据分析](#)。

### 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](#)相比，边缘计算可以减少通信延迟并缩短响应时间。

## 加密

一种将人类可读的纯文本数据转换为密文的计算过程。

### 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。大端序系统先存储最高有效字节。小端序系统先存储最低有效字节。

### 端点

参见[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 AWS PrivateLink 创建端点服务，并将权限授予其他 AWS 账户 或 AWS Identity and Access Management ( IAM ) 主体。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](#)。

### 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](#)。

### environment

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 security epics 包括身份和访问管理、侦测性控制、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](#)。

## 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 and 创建数据可视化得以执行。

## F

### 事实表

[星形架构](#)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。

### 失败得很快

一种使用频繁和增量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

参见[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。

## FGAC

请参阅[精细的访问控制](#)。

### 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，它使用连续的数据复制，通过[更改数据捕获](#)在尽可能短的时间内迁移数据，而不是使用分阶段的方法。目标是将停机时间降至最低。

## G

### 地理封锁

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，而[基于主干的工作流程](#)是现代的首选方法。

### 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。侦测性防护机制会检测策略违规和合规性问题，并生成警报以进行修复。它们通过使用 AWS Config、Amazon、AWS Security Hub GuardDuty AWS Trusted Advisor、Amazon Inspector 和自定义 AWS Lambda 支票来实现。

# H

## HA

参见[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。

### 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。

### hypercure 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 hypercure 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。



|

## laC

参见[基础架构即代码](#)。

### 基于身份的策略

附加到一个或多个 IAM 主体的策略，用于定义它们在 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

参见[工业物联网](#)。

### 不可变的基础架构

一种为生产工作负载部署新基础架构，而不是更新、修补或修改现有基础架构的模型。[不可变基础架构本质上比可变基础架构更一致、更可靠、更可预测](#)。有关更多信息，请参阅 Well-Architected Framework 中的[使用不可变基础架构AWS部署最佳实践](#)。

### 入站 ( 入口 ) VPC

在 AWS 多账户架构中，一种用于接受、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。

### 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置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，一种用于管理 VPC ( 相同或不同的 AWS 区域 )、互联网和本地网络之间的网络流量检查的集中式 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。

## 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[使用 AWS 实现机器学习模型的可解释性](#)。

## IoT

参见[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

登录区是一个架构完善、可扩展且安全的多账户 AWS 环境。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

见 [7 R](#)。

### 小端序系统

一个先存储最低有效字节的系统。另请参见[字节顺序](#)。

### 下层环境

参见[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

参见[分支](#)。

## 托管服务

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。亚马逊简单存储服务 (Amazon S3) Service 和 Amazon DynamoDB 就是托管服务的示例。这些服务也称为抽象服务。

## MAP

参见[迁移加速计划](#)。

## 机制

一个完整的过程，在此过程中，您可以创建工具，推动工具的采用，然后检查结果以进行调整。机制是一种在运行过程中自我增强和改进的循环。有关更多信息，请参阅在 Well-Architect AWS ed 框架中[构建机制](#)。

## 成员账户

除管理账户外，属于 AWS Organizations 中的组织的所有 AWS 账户。一个账户一次只能是一个组织的成员。

## 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 AWS 上实现微服务](#)。

## 迁移加速计划 ( MAP )

一项提供咨询支持、培训和服务的 AWS 计划，旨在帮助组织为迁移到云奠定坚实的运营基础，并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

跨职能团队，通过自动化、敏捷的方法简化工作负载迁移。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 Application Migration Service 将主机迁移到 Amazon EC2。

### 迁移组合评测 (MPA)

一种在线工具，提供了用于验证迁移到 AWS 云的业务用例的信息。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）。[MPA 工具](#)（需要登录）向所有 AWS 顾问和 APN 合作伙伴顾问免费提供。

### 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，找出优势和劣势，并制定行动计划来弥补发现的差距。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 云中的方法。有关更多信息，请参阅此词汇表中的 [7 R](#) 条目和[动员组织以加快大规模迁移](#)。

### ML

参见[机器学习](#)。

### MPA

参见[迁移组合评估](#)。

## 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。有关详细信息，请参阅[在 AWS 云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。有关详细信息，请参阅[在 AWS 云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](#)。

### 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？”

### 可变基础架构

一种用于更新和修改现有生产工作负载基础架构的模型。为了提高一致性、可靠性和可预测性，Well-Architect AWS ed Framework 建议使用[不可变基础设施](#)作为最佳实践。

## O

### OAC

请参阅[源站访问控制](#)。

### OAI

参见[源访问身份](#)。

### OCM

参见[组织变更管理](#)。

## 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

参见[运营集成](#)。

## OLA

参见[运营层协议](#)。

## 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。

## 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)

一份问题清单和相关的最佳实践，可帮助您理解、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 Well-Architecte AWS d Frame [work 中的运营准备情况评估 \(ORR\)](#)。

## 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 创建的跟踪，用于记录 AWS Organizations 中的组织的所有 AWS 账户 事件。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，这个框架称为人员加速，因为云采用项目需要快速的变革。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 支持所有 AWS 区域中的 S3 存储桶、使用 AWS KMS 的服务器端加密 ( SSE-KMS ) 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。

## 或者

参见[运营准备情况审查](#)。

## 出站 ( 出口 ) VPC

在 AWS 多账户架构中，一种用于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

查看[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。



## 策略

一个对象，可以在中定义权限（参见[基于身份的策略](#)）、指定访问条件（参见[基于资源的策略](#)）或定义组织中所有账户的最大权限AWS Organizations（参见[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。有关更多信息，请参阅[在微服务中实现数据持久性](#)。

## 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或的查询条件false，通常位于子WHERE句中。

## 谓词下推

一种数据库查询优化技术，可在传输前筛选查询中的数据。这减少了必须从关系数据库检索和处理的数据量，并提高了查询性能。

## 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

AWS 中可执行操作并访问资源的实体。该实体通常是 AWS 账户、IAM 角色或用户的根用户。有关更多信息，请参阅 IAM 文档中[角色术语和概念](#)中的主体。

## 隐私设计

一种贯穿整个工程化过程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](#)措施，旨在防止部署不合规的资源。这些控件会在资源置备之前对其进行扫描。如果资源与控件不兼容，则不会对其进行配置。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。

## 生产环境

参见[环境](#)。

## 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。

## Q

### 查询计划

一系列步骤，例如指令，用于访问 SQL 关系数据库系统中的数据。

### 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

参见“[负责任、负责、咨询、知情](#)” ([RACI](#))。

### 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

参见“[负责任、负责、咨询、知情](#)” ([RACI](#))。

### RCAC

请参阅[行和列访问控制](#)。

## 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。

## 重新设计架构

见 [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

见 [7 R](#)。

## 区域

地理区域中的 AWS 资源集合。每个 AWS 区域是孤立的，独立于其他的区域，以提供容错能力、稳定性和弹性。有关更多信息，请参阅在 AWS 一般参考中[管理 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

见 [7 R](#)。

## 版本

在部署过程中，推动生产环境变更的行为。

## 搬迁

见 [7 R](#)。

## 更换平台

见 [7 R](#)。

## 回购

见 [7 R](#)。

## 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

见 [7 R](#)。

## 退休

见 [7 R](#)。

## 旋转

定期更新 [密钥](#) 以使攻击者更难访问凭据的过程。

## 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

参见 [恢复点目标](#)。

## RTO

参见 [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。此功能可实现联合单点登录 (SSO)，因此用户可以登录 AWS Management Console 或调用 AWS API 操作，而无需在 IAM 中为组织中的每个人都创建用户。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](#)。

## SCP

参见[服务控制政策](#)。

## secret

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 文档中的密钥](#)。

## 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。安全控制主要有四种类型：[预防性](#)、[侦测](#)、[响应式](#)和[主动式](#)。

## 安全加固

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

一种预定义和编程的操作，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 实例或轮换证书。

## 服务器端加密

由接收数据的 AWS 服务 在目的地对数据进行加密。

## 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

AWS 服务的入口点的 URL。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。

## 服务级别指示器 (SLI)

对服务性能方面的衡量，例如其错误率、可用性或吞吐量。

## 服务级别目标 (SLO)

代表服务运行状况的目标指标，由服务[级别指标](#)衡量。

## 责任共担模式

一种描述您在云安全性和合规性方面与 AWS 共担的责任模型。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](#)。

## 暹粒

参见[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

参见[服务级别协议](#)。

## SLI

参见[服务级别指标](#)。

## SLO

参见[服务级别目标](#)。

## split-and-seed 模型

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[中的分阶段实现应用程序现代化的方法。AWS Cloud](#)

## 恶作剧

参见[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交易数据或测量数据，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 逐步将原有的 Microsoft ASP.NET \( ASMX \) Web 服务现代化](#)。

## 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。

## 对称加密

一种加密算法，它使用相同的密钥来加密和解密数据。

## 综合测试

以模拟用户交互的方式测试系统，以检测潜在问题或监控性能。你可以使用 [Amazon S CloudWatch ynthetic](#) 来创建这些测试。

# T

## tags

充当元数据的键值对，用于组织 AWS 资源。标签可帮助您管理、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

参见[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。

## 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[什么是中转网关](#)。

## 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

为您指定的服务授予权限，让其代表您在 AWS Organizations 的组织中及其账户中执行任务。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[将 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。有关更多信息，请参阅[量化深度学习系统中的不确定性指南](#)。

### 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

参见[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

一个 SQL 函数，用于对一组以某种方式与当前记录相关的行进行计算。窗口函数对于处理任务很有用，例如计算移动平均线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。

## 蠕虫

参见 [一次写入，多读](#)。

## WQF

请参阅 [AWS 工作负载资格框架](#)。

## 一次写入，多次读取 (WORM)

一种存储模型，它可以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，但他们无法对其进行更改。这种数据存储基础架构被认为是 [不可变的](#)。

## Z

### 零日漏洞利用

一种利用未修补 [漏洞](#) 的攻击，通常是恶意软件。

### 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。

### 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。