



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 © 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  |
| 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  |
| All-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 |
| Resources ( 资源 ) .....   | 13 |
| 文档历史记录 .....             | 14 |
| .....                    | xv |

# 在迁移 AWS 期间切断网络流量的最佳实践

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正在实施或领导向云迁移的任何其他角色 AWS。

## 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：

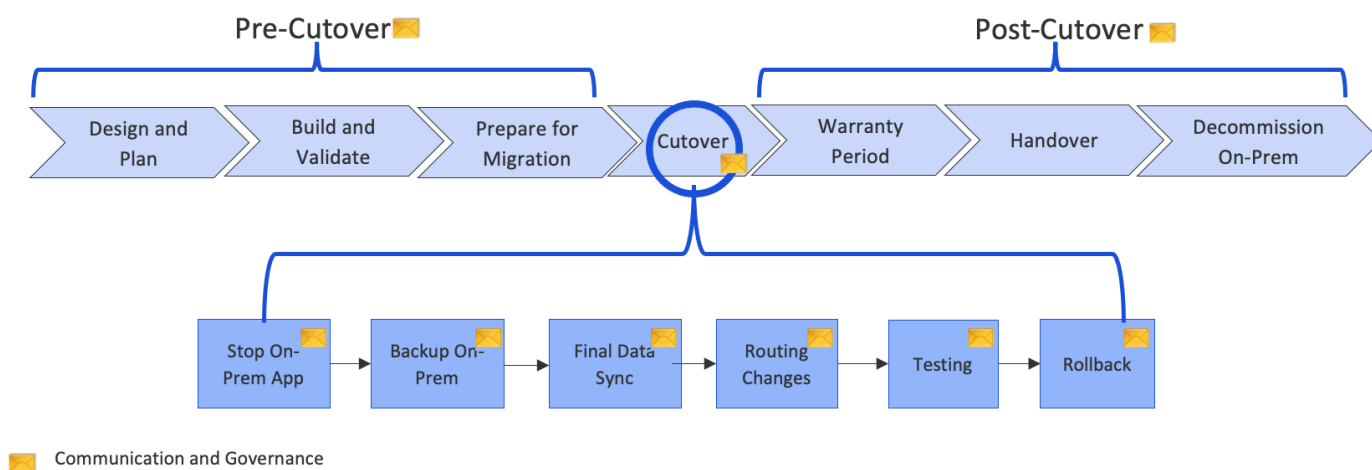
- 制定最佳割接流程
- 制定有关割接的沟通与治理策略
- 了解不同的割接方案，根据您的要求选择最有效的方案

## 割接阶段概述

在迁移过程的切换阶段，您可以将网络流量从现有端点转移到云中新部署的 AWS 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 或微软 Teams 聊天室 | 所有负责发布活动最新信息的利益相关者<br>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，与构建全新的堆栈 AWS 相比，迁移 VMware 环境的直接转换选项有所不同。

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
- 通过重复使用久经考验的机制将应用程序迁移到云端，从而提高工作 AWS 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

有两种割接方法可以考虑：一次性割接或分阶段割接。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。

### All-at-once 方法

如果采用一次性割接，只需按下开关就能割接整个解决方案。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。我们建议您的一次性割接计划应包括广泛的性能测试和回归测试（如适用），以便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段内逐步割接。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来源和应用程序服务器之间的延迟增加，这就清楚地表明需要采用一次性割接。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，然后切换到其 AWS 其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？

在回滚功能方面，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，这比一次性方法具有关键优势。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。

究竟是一次性割接一个应用程序或一堆相关应用程序，还是采用渐进的方式分阶段割接服务器和服务，要具体情况具体分析。我们通常看到客户采用以下方法：

- 能够容忍一定程度的停机时间的开发和测试环境将受益于简单、低工作量的一次性割接方法。
- 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。有关行动准备状态的更多信息，请参阅 AWS 文档[AWS Well-Architected](#)中的 AWS Well-Architected Tool 框架卓越运营支柱。

## 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：

- Fail-forward 方法 — 由于迁移后的数据库成为主数据库，因此您的本地 AWS 数据库在转换后可能会过时。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，仪表板会监控运营所需的关键服务级别协议 (SLA) 指标和服务级别指标 (SLIs)。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。

## Resources ( 资源 )

- [AWS WA 工具](#)
- [使用 AWS DMS 从迁移中回滚](#)
- [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 |

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。