AWS 白皮书

AWS 多区域基础知识

AWS 多区域基础知识: AWS 白皮书

Table of Contents

AWS 多区域基础知识

发布日期：2022 年 12 月 20 日 (文档修订)

摘要

这篇高级 300 级的 paper 面向有兴趣使用多区域架构来提高工作负载弹性的云架构师和高级领导AWS者。本 paper 假设对AWS基础设施和服务有基本了解。它概述了常见的多区域用例，分享了围绕设计、开发和部署的基本多区域概念和含义，并提供了规范性指导，以帮助您更好地确定多区域架构是否适合您的工作负载。

您使用 Well-Architected 了吗？

当您在云端构建系统时，AWS Well-Architected Framework 可助您了解所作决策的利弊。利用此框架的六个支柱，您可以了解到设计和运行可靠、安全、高效、经济有效且可持续的系统的架构最佳实践。您可以使用 AWS Management Console 免费提供的 AWS Well-Architected Tool，回答与每个支柱相关的一组问题，即可根据这些最佳实践检查自己的工作负载。

有关云架构的更多专家指导和最佳实践（参考架构部署、图表和白皮书），请参阅 AWS 架构中心。

简介

每个可用区都AWS 区域由一个地理区域内的多个独立且物理上独立的可用区组成。每个区域的软件服务之间保持了严格的逻辑隔离。这种有针对性的设计可确保一个地区的基础设施或服务故障不会导致另一个地区的相关故障。

大多数AWS客户可以使用多个可用区 (AZ) 或区域AWS服务在单个区域中实现其工作负载的弹性目标。但是，一部分客户之所以选择多区域架构，有三个原因。

- 他们认为单一区域无法满足的最高级别工作负载的高可用性和操作连续性要求。
- 它们需要满足数据主权要求（例如遵守当地法律、法规和合规性），这些要求工作负载在特定的司法管辖区内运行。
- 他们需要在离最终用户最近的地方运行工作负载，从而改善工作负载的性能和客户体验。

本 paper 侧重于操作的高可用性和连续性要求，并帮助您了解为工作负载采用多区域架构的注意事项。我们描述了适用于多区域工作负载设计、开发和部署的基本概念，以及一个规范性框架，可帮助您

确定多区域架构是否是特定工作负载的正确选择。您需要确保多区域架构是您的工作负载的正确选择，因为这些架构具有挑战性，而且如果操作不当，工作负载的整体可用性可能会降低。

设计和运营以提高单一地区的韧性

在深入探讨多区域概念之前，首先要确认您的工作负载已在单个区域中尽可能具有弹性。为实现这一目标，请根据Wel l-Architecte AWS d Framework的可靠性支柱和卓越运营支柱评估您的工作量，并进行任何必要的更改以采用推荐的最佳实践。Well-Architected 框架中涵盖AWS了以下概念：

- 基于域边界的工作负载分段
- 定义明确的服务合同
- 依赖关系管理和耦合
- 处理失败、重试和退缩策略
- 等能操作以及有状态与无状态事务
- 运营准备和变更管理
- 了解工作负载运行状况
- 回应事件

要进一步提高单区域弹性，请查看并应用高级多可用区弹性模式中讨论的概念来处理灰色故障。本白皮书提供了有关在每个可用区中使用副本来控制故障的最佳实践，并扩展了 W AWS ell Architected 中引入的多可用区概念。在完全应用了在单个区域实现最高弹性的推荐概念和最佳实践后，可以根据多区域架构的基础知识评估特定的工作负载，以确定是否可以使用多区域方法提高工作负载的弹性。

多区域基础知识 1：了解需求

如前所述，高可用性和操作连续性是采用多区域架构的常见原因。可用性指标衡量工作负载在定义的时间段内可供使用的时间百分比，而操作连续性指标则衡量大规模且持续时间通常较长的事件的恢复情况。

衡量可用性几乎是一个持续的过程。具体的衡量标准或指标可能有所不同，但通常围绕目标可用性进行汇总，通常被称为 9（例如 99.99% 的可用性）。就可用性目标而言，不能一刀切。需要在工作负载级别制定可用性目标，而不是在所有工作负载中应用单一目标，将非关键组件与关键组件分开。

为了保证操作的连续性，通常使用以下 point-in-time测量：

- 恢复时间目标 (RTO) - RTO 是服务中断和恢复服务之间可接受的最大延迟。此值决定了服务受损的可接受持续时间。
- 恢复点目标 (RPO)-RPO 是自上次数据恢复点以来的最大可接受时间量。这决定了在最新恢复点和服务中断之间哪些数据丢失被认为是可接受的。

与设置可用性目标类似，还应在工作负载级别定义 RTO 和 RPO。为了实现更严格的操作连续性或高可用性要求，需要增加投资。也就是说，并非每个应用程序都能要求或需要相同级别的弹性。创建分层机制可以帮助建立框架，使业务和 IT 所有者能够根据业务影响确定要求最苛刻的应用程序，并相应地对它们进行分层。分层的示例可在下表中找到。

表 1 — SLA 的弹性分层示例

可用性服务级别协议 (SLA)	弹性等级	可接受的停机时间/年
99.99%	铂	52.60 分钟
99.90%	黄金	8.77 小时
99.5%	银	1.83 天

表 2 — RTO 和 RPO 的弹性分层示例

套餐	最大 RTO	最大 RPO	标准	费用
铂	15 分钟	五分钟	任务关键型工作负载	$$$
黄金	15 分钟 — 六个小时	两个小时	重要，但不是任务关键型工作负载	$$
银	六个小时 — 几天	24 小时	非关键工作负载	$

在设计具有弹性的工作负载时，必须了解高可用性与操作连续性之间的关系。例如，如果工作负载需要 99.99% 的可用性，则每年的停机时间不能超过 53 分钟。检测故障可能至少需要五分钟，操作员还需要十分钟才能接触、决定恢复步骤并执行这些步骤。单个问题需要 30 到 45 分钟才能恢复，这种情况并不少见。在这种情况下，采用多区域策略来提供消除相关影响的孤立实例，可以在有限的时间内进行故障转移，同时对初始减值进行独立分类，从而实现持续运营。这就需要定义适当的 RTO 和 RPO。

对于具有极高的可用性需求（例如 99.99% 或更高的可用性）或严格的操作连续性要求且只能通过故障转移到另一个区域才能满足的任务关键型工作负载，多区域方法可能是合适的。但是，这些要求通常仅适用于企业工作负载组合中的一小部分，这些工作负载的恢复时间限制在几分钟或几小时内。除非应用程序需要几分钟或几小时的恢复时间，否则等待受影响地区内应用程序的区域中断得到修复可能是更好的方法，并且通常与较低级别的工作负载保持一致。

在实施多区域架构之前，业务决策者和技术团队应就成本影响（包括运营和基础设施成本驱动因素）进行协调。与单区域方法相比，典型的多区域架构的成本可能增加两倍。虽然业务连续性有几种多区域模式，例如在热待机、热待机和指示灯下运行，但实现恢复目标风险最低的模式将涉及运行热备用，并且会使您的工作负载成本增加一倍。

关键指导

- 运营目标（例如 RTO 和 RPO）的可用性和连续性应根据工作负载确定，并与业务和 IT 利益相关者保持一致。
- 大多数可用性和运营连续性目标都可以在单个区域内实现。对于单一区域无法实现的目标，应考虑多区域，同时明确成本、复杂性和收益之间的权衡。

多区域基础知识 2：了解数据

对于多区域架构来说，管理数据是一个不容忽视的问题。区域之间的地理距离会带来不可避免的延迟，这种延迟表现为跨区域复制数据所花费的时间。在可用性、数据一致性以及向使用多区域架构的工作负载引入更高数量级的延迟之间进行权衡是必要的。无论使用异步复制还是同步复制，您都需要修改应用程序以应对复制技术带来的行为变化。由于数据一致性和延迟方面的挑战，要采用专为单区域设计的现有应用程序并使其成为多区域非常困难。了解特定工作负载的数据一致性要求和数据访问模式对于权衡利弊至关重要。

2a：了解数据一致性要求

CAP 定理为推理数据一致性、可用性和网络分区之间的权衡提供了参考，对于工作负载，其中只有两个分区可以同时满足。根据定义，多区域包括区域之间的网络分区，因此您必须在可用性和一致性之间做出选择。

如果您选择跨区域的数据可用性，则在事务写入期间不会出现明显的延迟，因为在复制完成之前，需要对已提交的数据进行异步复制，从而降低各区域之间的一致性。对于异步复制，当主区域出现故障时，很有可能出现待从主区域复制的写入操作。这会导致一种情况，即在恢复复制之前，最新数据不可用，并且需要对账流程来处理未从经历中断的地区复制的正在进行的交易。

对于偏爱异步复制的工作负载，您可以使用提供异步跨区域复制的 Amazon Aurora 和 Amaz on DynamoDB 等服务。亚马逊 Aurora 全球数据库和亚马逊 DynamoDB 全局表都有默认的 CloudWatch 亚马逊指标，以帮助监控复制延迟。

设计工作负载以利用事件驱动架构对多区域策略来说是一个好处，因为这意味着工作负载可以包括数据的异步复制，并通过重播事件来实现状态重建。由于流媒体和消息服务在单个区域中缓冲消息有效载荷数据，因此区域故障转移/故障恢复流程必须包括一种机制，用于重定向客户端输入数据流，以及协调存储在经历中断的区域中的传输中和/或未交付的有效负载。

如果选择一致性，则由于在事务写入期间同步复制数据，将导致延迟很长。同步写入多个区域时，如果所有区域的写入操作均未成功，则可用性可能会降低，因为事务不会提交，需要重试。尝试同步向所有区域写入数据的重试将以每次尝试的延迟为代价。在某个时候，当重试次数用尽时，需要做出决定，要么使事务完全失败，从而降低可用性，要么仅将事务提交到可用区域，从而导致不一致。有些法定人数形成技术，例如 Paxos，可以帮助同步复制和提交数据，但需要开发人员的大量投资。

当写入涉及跨多个区域的同步复制以满足严格的一致性要求时，写入延迟会增加一个数量级。如果不进行重大更改，通常无法将较高的写入延迟改装到应用程序中。理想情况下，在首次设计应用程序时必须将其考虑在内。对于优先考虑同步复制的多区域工作负载，AWS合作伙伴解决方案可以提供帮助。

2b：了解数据访问模式

工作负载数据访问模式分为以下类型之一：读取密集型或写入密集型。了解特定工作负载的这一特性将指导选择合适的多区域架构。

对于读取密集型工作负载，例如完全只读的静态内容，可以在不显著复杂的情况下实现主动/主动多区域架构。使用内容分发网络 (CDN) 在边缘提供静态内容，通过缓存离最终用户最近的内容来确保可用性；在 Amazon 中使用诸如 Origin 故障转移之类的功能集 CloudFront可以帮助实现这一目标。另一种选择是在多个区域部署无状态计算，并使用 DNS 将用户路由到最近的区域以读取内容。可以使用@@带有地理定位路由策略的 Route 53 来实现这一目标。

对于读取比例大于写入百分比的读取密集型工作负载，可以使用本地读取、写入全局策略。这需要将所有写入操作都发送到特定区域的数据库，同时将数据异步复制到所有其他区域，并且可以在任何区域进行读取以实现此目的。这种方法需要工作负载才能实现最终一致性，因为跨区域写入复制的延迟会增加，因此本地读取可能会过时。

Aurora Global Databas e 可以帮助在只能在本地处理所有读取流量的备用区域中配置只读副本，并在特定区域配置单个主数据存储以处理写入。数据从主数据库异步复制到备用数据库（只读副本），如果您需要将操作故障转移到备用区域，则可以将备用数据库提升为主数据库。如果工作负载更适合非关系数据模型，则也可以在这种方法中使用 DynamoDB。同样，工作负载需要包含最终的一致性，如果不是从一开始就为此而设计的，则可能需要对其进行重写。

对于写入密集型工作负载，应选择主区域，并在工作负载中设计故障转移到备用区域的功能。与主动/主动方法相比，主/备用方法不那么复杂。这是因为对于主动/主动架构，需要重写工作负载，以处理到区域的智能路由、建立会话关联性、确保等效事务以及处理潜在的冲突。

大多数考虑多区域恢复能力的工作负载不需要主动/主动方法。分片策略可用于通过限制整个客户群中损伤的爆炸半径来提高弹性。如果您可以有效地对客户群进行分片，则可以为每个分片选择不同的主区域。例如，如果您可以对客户端进行分片，使一半的客户端与区域一对齐，一半的客户端与区域二对齐，将区域视为单元，则可以创建多区域单元方法，从而缩小工作负载的冲击半径。

分片方法可以与主/备用方法相结合，为分片提供故障转移功能。需要在工作负载中设计经过测试的故障转移流程，还需要设计数据协调流程，以确保故障转移后数据存储的事务一致性。这些 paper 稍后将详细介绍。

关键指导

- 出现故障时，待复制的写入操作很可能不会提交到备用区域。在恢复复制之前，数据将不可用（假设异步复制）。

- 作为故障转移的一部分，需要一个数据协调过程，以确保使用异步复制的数据存储保持事务一致的状态。
- 当需要强一致性时，需要修改工作负载以容忍同步复制的数据存储所需的延迟。

多区域基础知识 3：了解您的工作负载依赖关系

特定工作负载在一个区域中可能有多个依赖关系，例如使用的AWS服务、内部依赖关系、第三方依赖关系、网络依赖关系、证书、密钥、机密和参数。为了确保工作负载在故障情况下运行，主区域和备用区域之间不应存在任何依赖关系；每个区域都应能够相互独立运行。为此，必须仔细检查工作负载中的所有依赖关系，以确保它们在每个区域中都可用。这是必需的，因为主区域的故障不应影响备用区域。此外，当依赖关系处于降级状态或完全不可用时，必须了解工作负载是如何运行的，这样才能设计出适当的解决方案来处理这个问题。

3a：服务 AWS

在设计多区域架构时，必须了解将要使用的特定AWS服务。第一个方面是了解该服务必须具备哪些功能才能启用多区域，以及是否必须设计解决方案以实现多区域目标。例如，在 Amazon Aurora 和 Amazon DynamoDB 中，有一项功能可以将数据异步复制到备用区域。任何AWS服务依赖关系都需要在运行工作负载的所有区域中都可用。为确保要使用的服务在所需区域可用，请查看AWS 区域所有服务列表。

3b：内部依赖关系和第三方依赖关系

对于工作负载存在的任何内部依赖关系，请确保该工作负载将在其外运行的区域中可用。例如，如果工作负载由许多微服务组成，则应了解构成业务能力的所有微服务。然后，确保所有这些微服务都部署在工作负载将要运行的每个区域。

不建议在工作负载内的微服务之间进行跨区域调用，因此应保持区域隔离。 这是因为创建跨区域依赖关系会增加相关失败的风险，这抵消了你试图通过工作负载的孤立区域实现所获得的好处。本地依赖关系也可能是工作负载的一部分，因此，当务之急是了解如果主要区域发生变化，这些集成的特征会如何变化。例如，如果备用区域距离本地环境更远，则延迟的增加将产生负面影响。

了解软件即服务 (SaaS) 解决方案、软件开发套件 (SDK) 和其他第三方产品依赖关系，并能够演练这些依赖关系降级或不可用的场景，将使人们更深入地了解系统链在不同故障模式下的运行和行为。这些依赖关系可能存在于应用程序代码中，从如何使用 AWS Secrets Manager 或第三方保管库解决方案（例如 Hashicorp）在外部管理机密，到依赖于 IAM Identity Center 进行联合登录的身份验证系统。

在依赖关系方面拥有冗余可以帮助提高弹性。SaaS 解决方案或第三方依赖项也有可能使用与工作负载 AWS 区域相同的主服务器。如果是这种情况，您应该与供应商合作，确定他们的弹性状态是否符合工作负载的要求。

此外，请注意工作负载及其依赖关系（例如第三方应用程序）之间的共同命运。如果故障转移后在辅助区域（或来自辅助区域）的依赖关系不可用，则工作负载可能无法完全恢复。

3c：故障转移机制

域名系统 (DNS) 通常用作故障转移机制，用于将流量从主区域转移到备用区域。严格审查和仔细检查故障转移机制所需要的所有依赖关系。例如，如果您的工作负载使用的是 Amazon Route 53，则了解控制平面托管在 US-East-1 意味着您依赖该特定区域的控制平面。如果主区域也是 US-east-1，则不建议将其作为故障转移机制的一部分。如果使用另一种故障转移机制，则需要深入了解其无法按预期运行的任何情况。一旦达成了这种谅解，就要做好应急计划或在需要时制定新的机制。查看使用 Amazon Route 53 创建灾难恢复机制，了解可用于成功进行故障转移的方法。

如内部依赖关系部分所述，作为业务能力一部分的所有微服务都需要在部署工作负载的每个区域中可用。作为故障转移策略的一部分，业务功能需要一起进行故障转移，以消除跨区域调用的机会。或者，如果微服务独立进行故障转移，则可能会出现不良行为，即微服务可能会进行跨区域调用，这会带来延迟，并可能导致在客户端超时时时工作负载不可用。

3d：配置依赖关系

证书、密钥、密钥和参数是设计多区域时所需的依赖关系分析的一部分。只要有可能，最好在每个区域内对这些组件进行本地化，这样它们就不会因为这些依赖关系而在区域之间共享命运。对于证书，证书的到期时间应各不相同，如果可能，也应在每个区域中有所不同，以避免即将到期的证书（警报设置为提前通知）影响多个区域的情况。

加密密钥和机密也应该是特定于区域的。这样，如果密钥或密钥的轮换出现错误，则影响仅限于特定区域。

最后，所有工作负载参数都应存储在本地，以便在特定区域中检索工作负载。

关键指导

- 多区域架构受益于区域间的物理和逻辑分离。在应用层引入跨区域依赖关系会破坏这一好处。避免此类依赖。
- 故障转移控制应在不依赖于主区域的情况下起作用。
- 需要在业务能力上协调故障转移，以消除延迟增加和跨区域呼叫依赖性的可能性。

多区域基础知识 4：运营准备就绪

操作多区域工作负载是一项复杂的任务，会带来多区域特有的运营挑战。其中包括AWS 账户管理、重新调整部署流程、创建多区域可观察性策略、创建和测试故障转移和故障恢复运行手册，然后管理成本。运营准备情况评估 (ORR) 可以帮助团队为生产工作做好准备，无论是在单个区域还是在多个区域运行。

4a：管理 AWS 账户

要在跨区域部署工作负载AWS 区域，请确保账户内的所有AWS服务配额在不同区域之间保持平衡。首先，了解架构中的所有AWS服务，查看备用区域的计划使用情况，然后将其与当前使用情况进行比较。在某些情况下，如果以前未使用过备用区域，则可以参考默认服务配额来了解起点。然后，在将要使用的所有服务中，使用 S ervice Quotas 控制台（需要登录）或 API 申请增加配额。

AWS需要在每个区域配置 Ident@@ ity and Access Managem ent (IAM) 角色，以确保操作员、自动化工具和AWS服务对备用区域内的资源拥有适当的权限。区域隔离角色实现了我们对多区域架构所追求的区域隔离。在备用区域上线之前，请确保这些权限已到位。

4b：部署实践

使用多区域功能，将工作负载部署到多个区域可能很复杂。 AWS CloudFormation有助于将基础设施部署到单个或多个区域，并且可以根据您的需求进行定制。 AWS CodePipeline有助于提供近乎持续的集成/持续交付 (CI/CD) 管道，该管道具有跨区域操作，允许部署到与管道所在区域不同的区域。再加上蓝/绿等强大的部署策略，可以实现最少至零的停机时间部署。

但是，当应用程序或数据的状态未外部化到持久存储时，有状态功能的部署可能会更加复杂。在这些情况下，请仔细调整部署流程以满足您的需求。设计部署管道和流程，一次部署在一个区域，而不是同时在多个区域部署。这减少了区域之间出现相关故障的机会。要了解 Amazon 用于自动部署软件的技术，请阅读生成器库文章自动执行安全、无需动手的部署。

4c：可观察性

在设计多区域时，请考虑如何监控每个区域中所有组件的运行状况，以全面了解区域运行状况。这可能包括复制延迟的监控指标，这不是单个区域工作负载的考虑因素。

在构建多区域架构时，也要考虑从备用区域观察工作负载的性能。 这包括在备用区域运行运行状况检查和加那利群岛（综合测试），从而提供主区域健康状况的外部视图。此外，您还可以使用 Amazon CloudWatch Internet Monit or 从最终用户的角度了解外部网络的状态和工作负载的性能。同样，主区

域应具有相同的可观察性来监控备用区域。这些加那利群岛应该监控客户体验指标，以了解工作负载的整体运行状况。这是必需的，因为如果主区域出现问题，则主区域的可观察性可能会受到损害，从而影响评估工作负载运行状况的能力。

在这种情况下，在该区域之外进行观察可以提供见解。这些指标应汇总到每个区域可用的仪表板中，并在每个区域创建警报。由于 Amazon CloudWatch 是一项区域性服务，因此需要在两个区域都提供这些服务。此监控数据将用于调用从主区域到备用区域的故障转移。

4d：流程、程序和测试

回答“我应该何时进行故障转移？”这个问题的最佳时机 早在你需要之前。包括人员、流程和技术在内的业务连续性计划都应在问题出现之前尽早确定，并定期进行测试。确定恢复决策框架。如果有经过良好实践的恢复过程并且对恢复时间了如指掌，则可以选择启动通过故障转移达到 RTO 目标的恢复过程的时间点。这个时间点可能是在发现主区域的应用程序存在问题之后立即出现的，也可能是在该区域应用程序中的恢复选项已经用尽之后，现在应该开始故障转移以满足 RTO。

虽然故障转移操作本身应实现 100% 自动化，但激活故障转移的决定应由人力（通常是组织中少数预先确定的个人）做出。此外，决定故障转移的标准需要明确界定，并让组织全面了解这些标准。这些流程可以使用 Sy AWSstem Manager 运行手册来定义和完成，它允许完全 end-to-end 自动化，并确保测试和故障转移期间流程运行的一致性。

这些运行手册应在主区域和备用区域中可用，以启动故障转移或故障恢复过程。一旦实现了这种自动化，就应该定义并遵循定期的测试节奏。这样可以确保在发生实际事件时，响应是在组织有信心的明确、实践的流程上进行的。同样重要的是要记住数据协调过程的既定容差。确认建议的流程符合既定的 RPO/RTO 要求。

4e：成本和复杂性

多区域架构的成本影响是由更高的基础设施使用率、运营开销和资源时间造成的。如前所述，备用区域的基础设施成本与预配置时主区域的基础设施成本相似，因此成本是原来的两倍。配置容量使其足以满足日常运营，但仍保留足够的缓冲容量以容忍需求激增——并在每个区域配置相同的限制。

此外，如果您采用主动-主动架构，则可能需要进行应用程序级别的更改才能在多区域架构中成功运行，而主动-主动架构的设计和操作可能需要花费大量时间和资源。组织至少需要花时间了解每个地区的技术和业务依赖关系，并设计故障转移和故障恢复流程。

团队还应进行正常的故障转移和故障恢复练习，以便对活动期间使用的运行手册感到满意。尽管这些活动对于从多区域投资中获得预期结果极其重要和关键，但它们代表着机会成本，会占用其他活动的时间和资源。

关键指导

- AWS需要审查服务配额，并在工作量所在的所有地区保持平等。
- 部署过程应一次针对一个区域，而不是同时针对多个区域。
- 需要监控复制延迟等其他指标，这些指标特定于多区域场景。
- 将工作负载的监控范围扩展到主区域之外。应按区域监控客户体验指标，并从每个运行工作负载的区域之外进行衡量。
- 需要定期测试故障转移和故障恢复。确保实施在测试和直播活动期间使用的故障转移和故障恢复流程的单一操作手册。测试和直播活动的运行手册不能不同。

结论

本白皮书讨论了多区域的常见用例、如何实现多区域架构的基础知识以及这种方法的意义。这些基础知识可以应用于任何工作负载，并可用作框架，以帮助决策多区域架构是否适合特定业务。

贡献者

本文档的贡献者包括：

技术贡献者：

- John Formento, Jr.，AWS多区域团队首席解决方案架构师

编辑撰稿人：

- Lisi Lewis，产品营销高级经理

延伸阅读

如需了解其他信息，请参阅：

- 高级多可用区弹性模式（AWS白皮书）
- 可靠性支柱——Well-Arch AWS itected 框架
- 可用性及其他：了解和提高分布式系统的弹性 AWS（AWS白皮书）
- AWS故障隔离边界（AWS白皮书）

文档修订

如需获取有关本白皮书更新的通知，请订阅 RSS 源。

变更	说明	日期
文件已发布	首次出版。	2022 年 12 月 20 日

版权声明

客户有责任对本文档中的信息进行单独评测。本文档：(a) 仅供参考，(b) 代表当前的 AWS 产品和实践，如有更改，恕不另行通知，以及 (c) 不构成 AWS 及其附属公司、供应商或许可方的任何承诺或保证。AWS 产品或服务“按原样”提供，不附带任何明示或暗示的保证、陈述或条件。AWS 对其客户承担的责任和义务受 AWS 协议制约，本文档不是 AWS 与客户直接协议的一部分，也不构成对该协议的修改。

AWS 术语表

有关最新的 AWS 术语，请参阅《AWS 词汇表参考》中的 AWS 词汇表。

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。