



模块 2：云架构简介

AWS Academy Cloud Architecting

© 2023, Amazon Web Services, Inc. 或其联属公司。保留所有权利。

欢迎学习模块 2: 云架构简介。

模块概览

章节

1. 什么是云架构？
2. Amazon Web Services (AWS) Well-Architected Framework
3. 在 AWS 上构建解决方案的最佳实践
4. AWS 全球基础设施



知识考核



© 2023, Amazon Web Services, Inc. 或其联属公司。保留所有权利。

2

本模块包括以下章节：

1. 什么是云架构？
2. Amazon Web Services (AWS) Well-Architected Framework
3. 在 AWS 上构建解决方案的最佳实践
4. AWS 全球基础设施

在本模块结束时，您需要完成一个知识考核，以测试您对本模块中涵盖的关键概念的理解程度。

模块目标

学完本模块后，您应该能够：

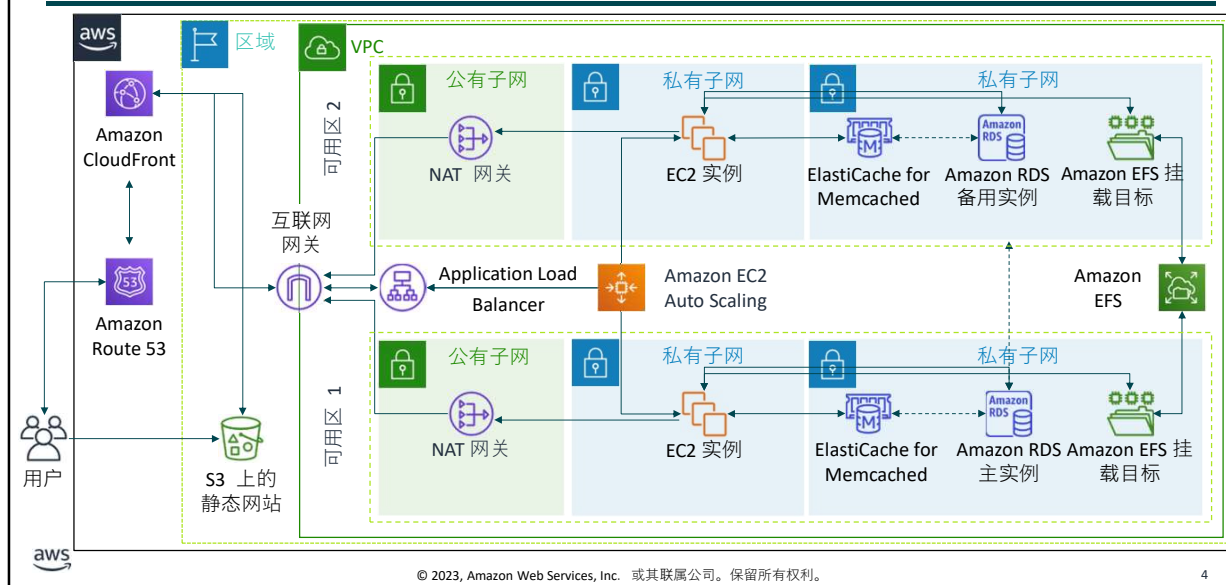
- 定义云架构
- 描述如何使用 AWS Well-Architected Framework 来设计和评估架构
- 解释在 AWS 上构建解决方案的最佳实践
- 描述如何就 AWS 资源的放置位置作出明智的决策



学完本模块后，您应该能够：

- 定义云架构
- 描述如何使用 AWS Well-Architected Framework 来设计和评估架构
- 解释在 AWS 上构建解决方案的最佳实践
- 描述如何就 AWS 资源的放置位置作出明智的决策

一个大型架构



本课程结束时，您将了解此架构图中的所有组件。您还能构建您自己的大型、稳健的解决方案架构，如同本示例中的一样。在课程大部分模块的开头，您都会重复看到本图。图中的新组件将随着课程中的介绍揭示。

第 1 节：什么是云架构？

模块 2：云架构简介



© 2023, Amazon Web Services, Inc. 或其联属公司。保留所有权利。

介绍第 1 节：什么是云架构？

架构需求

从 2000 年左右开始，Amazon 一直在努力让自己的新购物网站具有高可用性和可扩展性。



要了解什么是云架构以及它为何重要，首先要举例说明没有云架构时的软件开发情况。

在 2000 年左右，Amazon 试图创建一项电子商务服务，使第三方卖家能够在 Amazon 电子商务引擎之上建立自己的在线购物网站。该公司一直在努力让自己的新购物网站具有高可用性和可扩展性。

AWS 的起源

- 根据 AWS 首席执行官 Andy Jassy 的说法，当时 Amazon 电子商务工具是“一团糟”
- 应用程序和架构的构建没有经过正确的规划
- 很难将服务彼此分开
- 解决方案：Amazon 创建了一组记录完备的 API，并作为公司的服务开发标准



在[关于 AWS 起源的 TechCrunch 采访中](#)，AWS 首席执行官 (CEO) Andy Jassy 表示，一开始，Amazon 电子商务工具是“一团糟”。应用程序和架构的构建没有经过正确的规划。Jassy 还表示，“将各种服务分开，构建集中式开发平台是一个巨大的挑战。”

这个问题的解决方案是，创建一组记录完备的 Application Programming Interface (API) 来组织开发环境。

问题依然存在

- Amazon 仍然难以快速构建应用程序。
 - 数据库、计算和存储组件耗时 3 个月来构建。
 - 每个团队构建自己的资源，没有可扩展性或重用性方面的规划。
- 解决方案：Amazon 构建了内部服务，以便在基础设施上创建高度可用、可扩展并且可靠的架构。2006 年，Amazon 开始将这些服务作为 AWS 进行销售。

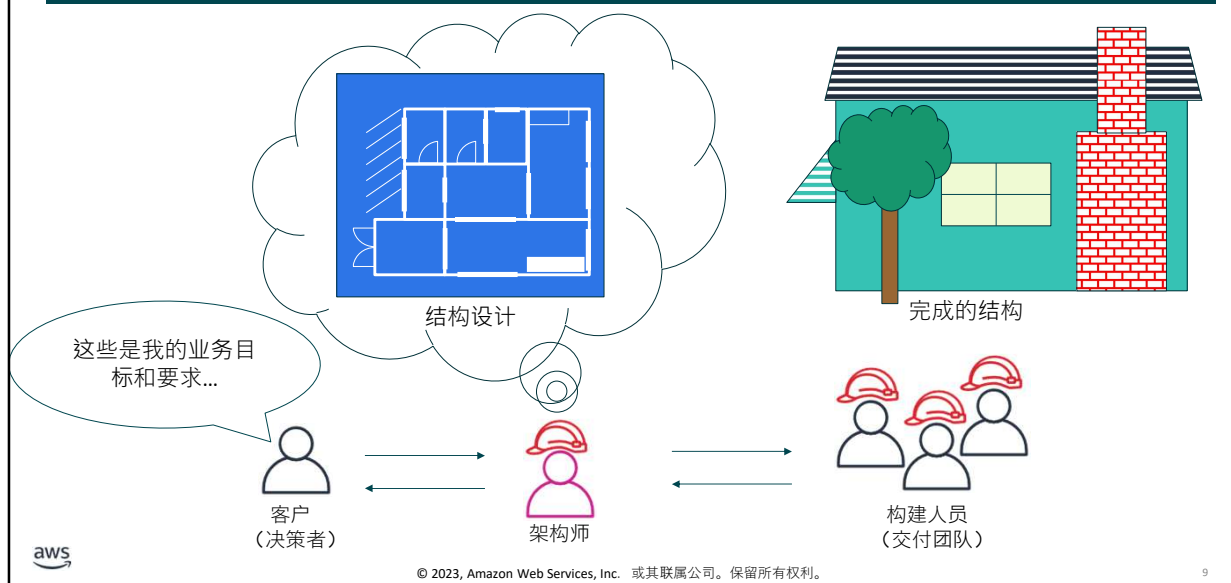


但是，随着公司的发展以及聘用了更多软件工程师，Amazon 仍然难以快速构建应用程序：

- 本来整个项目预计需要 3 个月的时间，但是构建数据库、计算和存储组件却花了 3 个月的时间。
- 每个团队构建自己的资源，而没有可扩展性或可重复使用性方面的规划。

解决方案是构建内部服务，以便在 Amazon 基础设施上创建高度可用、可扩展并且可靠的架构。2006 年，Amazon 开始将这些服务作为 AWS 进行销售。

云架构



那么，什么是云架构？云架构是将云特性应用于一个解决方案，使解决方案可以利用云服务和功能来满足企业技术需求和业务使用案例。解决方案类似于建筑物的蓝图。

软件系统需要架构师来管理其规模和复杂性。

云架构师：

- 与决策者交流，以确定业务目标和需要改进的能力。
- 确保解决方案的技术交付成果与业务目标保持一致。
- 与实施解决方案的交付团队合作，以确保技术功能适用。

拥有架构完善的系统会增加技术交付成果有助于实现业务目标的可能性。

第 1 节要点



- 云架构是将云特性应用于一个解决方案，使解决方案可以利用云服务和功能来满足企业技术需求和业务使用案例
- 您可以使用 **AWS** 服务创建高度可用、可扩展且可靠的架构

本模块中这节内容的要点包括：

- 云架构是将云特性应用于一个解决方案，使解决方案可以利用云服务和功能来满足企业技术需求和业务使用案例
- 您可以使用 **AWS** 服务创建高度可用、可扩展且可靠的架构

第 2 节：AWS Well-Architected Framework

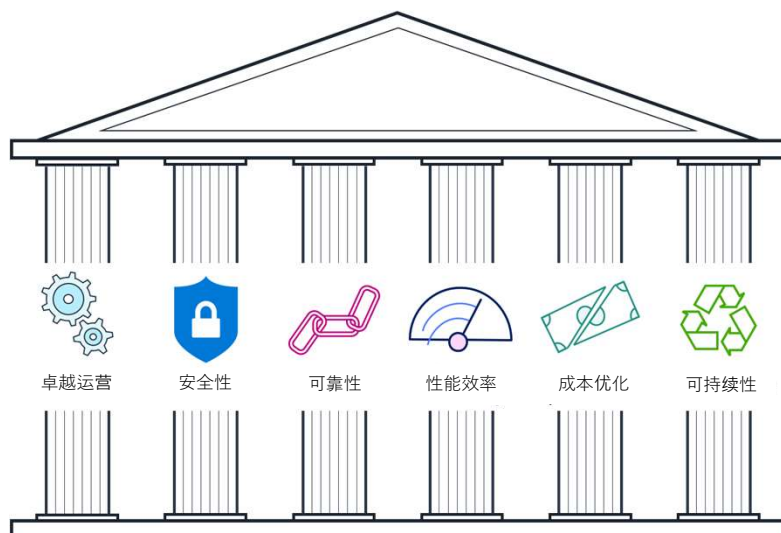
模块 2：云架构简介



© 2023, Amazon Web Services, Inc. 或其联属公司。保留所有权利。

介绍第 2 节：AWS Well-Architected Framework。

AWS Well-Architected Framework 的支柱



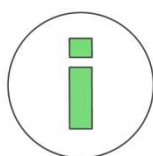
© 2023, Amazon Web Services, Inc. 或其联属公司。保留所有权利。

12

AWS Well-Architected Framework 是一份指南，提供了评估云架构的一致方法和有助于实施设计方案的指导。它记录了一系列基本问题和最佳实践，可让您了解某个特定架构是否高度符合云最佳实践。在审查了 AWS 上的数千个客户架构后，AWS 开发了此框架。

AWS Well-Architected Framework 分为六大支柱：卓越运营、安全性、可靠性、性能效率、成本优化和可持续性。自 2015 年该框架推出以来，前五大支柱一直是该框架的组成部分。可持续性支柱是 2021 年新增的第六个支柱，旨在帮助企业了解如何最大限度地减少运行云工作负载对环境的影响。

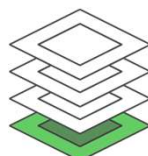
安全性支柱



身份机制



可跟踪性



在所有层确保安全性



风险评估与缓解策略



安全性支柱能够保护信息、系统和资产，同时通过风险评估和缓解策略实现商业价值。

只需采取一些措施，您的架构就会处于更好的安全状态。这些措施包括采用强大的身份机制、实现可跟踪性、在所有层确保安全性、自动应用安全性最佳实践以及保护传输中的数据和静态数据。

有关安全性最佳实践的更多信息，请参阅[安全性支柱白皮书](#)。

卓越运营支柱

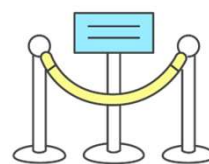
- 能够运行和监控各种系统
- 不断改善支持流程和程序



已部署



已更新



已运行



© 2023, Amazon Web Services, Inc. 或其联属公司。保留所有权利。

14

卓越运营支柱能够实现运行系统和深入了解其运营以实现商业价值。它还能实现持续改进支持流程和程序。

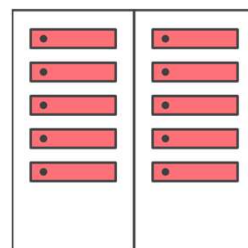
在设计运营工作负载时，您必须了解其如何部署、更新和运作。实施符合减少缺陷和快速安全修复的工程实践。通过日志记录、仪器以及业务和技术指标启用观察，以便您可以深入了解架构内发生的情况。

在 AWS 中，您可以将整个工作负载（应用程序、基础设施、策略、监管和操作）视为代码。工作负载可以在代码中定义，并使用代码来更新。这意味着您可以将用于应用程序代码的设计规范应用到堆栈中的每个元素。

有关卓越运营最佳实践的更多信息，请参阅[卓越运营支柱白皮书](#)。

可靠性支柱

- 迅速从基础设施或服务故障中恢复
- 动态获取计算资源以满足需求
- 减少中断，例如：
 - 配置错误
 - 暂时性网络问题



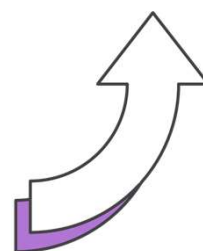
可靠性支柱能够实现系统从基础设施或服务中断中恢复，以及动态获取计算资源以满足需求。它还能够助力系统减少中断（例如配置错误或暂时性网络问题）。

在传统环境中确保可靠性可能会很困难。单点故障、缺乏自动化和缺乏弹性都会引起问题。通过应用可靠性支柱中概述的最佳实践，您可以避免许多此类问题。在高可用性、容错能力和整体冗余方面正确设计架构对您和您的客户都会有所帮助。

有关可靠性最佳实践的更多信息，请参阅[可靠性支柱白皮书](#)。

性能效率支柱

- 选择有效的资源并在需求变化时保持资源效率
- 普及先进技术
- 应用机械同感



在考虑性能时，您需要有效使用计算资源，从而最大限度地提高性能。随着需求的变化，您还希望保持这种效率。

普及先进技术也同样重要。在自己难以实施技术的情况下，请考虑使用供应商。通过为您实施技术，供应商利用所掌握的知识来处理复杂问题，让您的团队专注于附加价值更高的工作。

*机械同感*是在使用工具或系统时，了解其最佳运行方式。使用与您要实现的目标最为一致的技术方法。例如，在选择数据库或存储方法时考虑数据访问模式。

有关性能最佳实践的更多信息，请参阅[性能效率支柱白皮书](#)。

成本优化支柱

- 衡量效率
- 消除不必要的支出
- 考虑使用托管服务



成本优化是所有良好架构设计的长期要求。这一过程重复进行，应该在您的整个生产生命周期内完善和改进。了解当前的架构相对于目标的效率，可以消除不必要的费用。考虑使用托管服务，因为它们在云规模下运行，因此可以降低事务处理成本或服务成本。

要了解更多信息，请参阅[成本优化支柱白皮书](#)。

可持续性支柱

- 了解影响
- 制定可持续发展目标
- 最大限度地提高利用率
- 预测并采用更高效的新型软硬件产品
- 降低云工作负载对下游的影响



© 2023, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

18

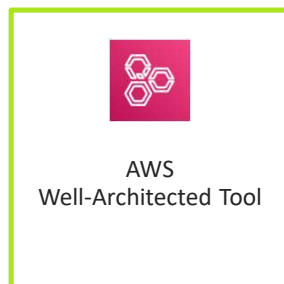
可持续性支柱涉及构建能最大限度提高效率和减少浪费的架构的能力。

云中的可持续性是一项持续的工作，主要集中在工作负载所有组件的节能和增效上，方法是从所预置的资源中获得最大效益，并最大限度地减少所需的资源总量。这项工作包括最初选择高效的编程语言、采用现代算法、使用高效的数据存储技术、部署正确规模和高效的计算基础设施，以及尽量减少对高功率终端用户硬件的要求。

有关云中可持续性设计原则的更多信息，请参阅 *可持续性支柱：AWS Well-Architected Framework* 白皮书，网址为

<https://docs.aws.amazon.com/wellarchitected/latest/sustainability-pillar/sustainability-pillar.html>。

AWS Well-Architected Tool



- 有助于您查看工作负载的状态，并将其与最新的 AWS 架构最佳实践进行对比
- 您可以根据需要，随时访问 AWS 架构师使用的知识和最佳实践
- 提供行动计划，并逐步指导如何为云构建更好的工作负载
- 提供一致的流程，供您查看和测评自己的云架构



如果您希望在设计架构完善的解决方案方面获得帮助，可以使用 AWS Well-Architected Tool。AWS Well-Architected Tool 是一种自助式工具，借助此工具，您可以根据需要随时访问当前的 AWS 最佳实践。这些最佳实践旨在帮助您在 AWS 上构建安全、高性能、具有弹性和高效的应用程序基础设施。

AWS Well-Architected Tool 有助于您查看工作负载的状态，并将其与最新的 AWS 架构最佳实践进行对比。它使您可以根据需要随时访问 AWS 架构师使用的知识和最佳实践。

此工具在 AWS 管理控制台中提供。您可以定义工作负载，并回答卓越运营、安全性、可靠性、性能效率和成本优化领域中的一系列问题。之后，AWS Well-Architected Tool 会提供一个行动计划，此计划逐步指导您针对云环境改进工作负载。

AWS Well-Architected Tool 提供一致的流程，供您查看和测评自己的云架构。您可以使用该工具提供的结果来确定改进的后续步骤、推动架构决策，并将架构注意事项纳入公司监管流程。

要了解有关 AWS Well-Architected Tool 的更多信息，请参阅 [AWS Well-Architected Tool 网站](#)。

第 2 节要点



- AWS Well-Architected Framework 提供了一致的方法来评估云架构，还提供了有助于实现设计方案的指南
- AWS Well-Architected Framework 分为六个支柱
- 每个支柱记录了一系列基本问题，可让您了解某个特定架构是否高度符合云最佳实践
- AWS Well-Architected Tool 有助于您查看工作负载的状态，并将其与最新的 AWS 架构最佳实践进行对比

本模块中这节内容的要点包括：

- AWS Well-Architected Framework 提供了一致的方法来评估云架构，还提供了有助于实现设计方案的指南。
- AWS Well-Architected Framework 分为六个支柱。
- 每个支柱记录了一系列基本问题，可让您了解某个特定架构是否高度符合云最佳实践。
- AWS Well-Architected Tool 有助于您查看工作负载的状态，并将其与最新的 AWS 架构最佳实践进行对比。

第 3 节：在 AWS 上构建解决方案的最佳实践

模块 2：云架构简介

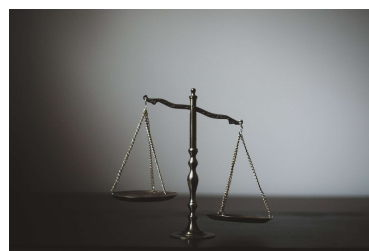


© 2023, Amazon Web Services, Inc. 或其联属公司。保留所有权利。

介绍第 3 节：在 AWS 上构建解决方案的最佳实践。

设计权衡

- 评估权衡，以便您可以选择最佳方法
- 权衡的示例包括：
 - 在一致性、持久性和空间方面进行妥协以减少用时和延迟，从而提供更高的性能
 - 优先考虑新功能的上市速度而不是成本
- 基于经验数据作出设计决策



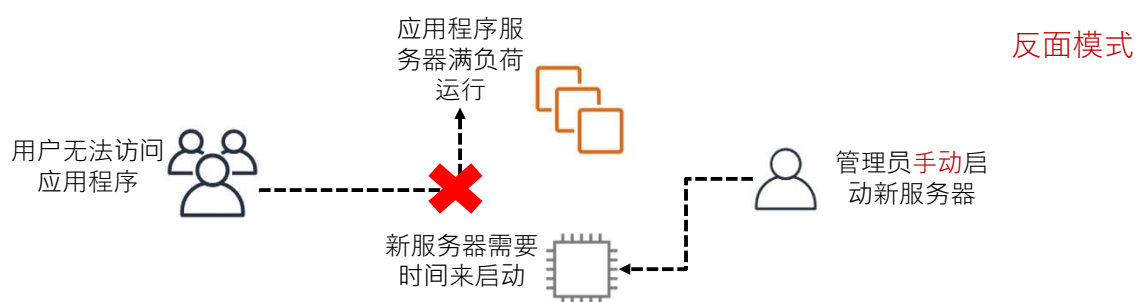
在设计解决方案时，请仔细考虑权衡，以便选择最佳方法。例如，您可以在一致性、持久性和空间方面进行妥协以减少用时和延迟，从而提供更高的性能。或者，您可能会优先考虑上市速度而不是成本。

权衡可能会增加架构的成本和复杂性，因此您的设计决策应该基于经验数据。例如，您可能需要执行负载测试以确保在性能方面获得可衡量的优势。或者，您可能需要执行基准测试，以实现随着时间推移最具成本优势的工作负载。在评估与性能相关的改进时，您还需要考虑架构设计选择将如何影响客户和工作负载效率。

在本节中，您将了解在 AWS 上设计解决方案的最佳实践。您还将了解应避免的反面模式（或错误的解决方案设计）。

1. 启用可扩展性 (1/2)

确保您的架构能够应对需求的变化。



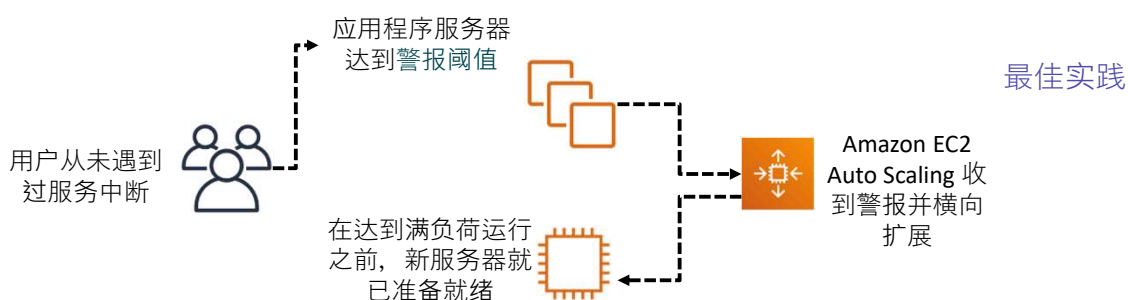
在 AWS 云上运行工作负载时，您可以快速、主动地扩展基础设施。确保在基础设施的每个层实施可扩展性。

要了解可扩展性的重要性，请考虑一下这种反面模式，即以被动和手动的方式进行扩展。

在这种情况下，当应用程序服务器满负荷运行时，用户将无法访问应用程序。然后，管理员手动启动一个或多个新实例来管理负载。遗憾的是，实例启动后需要几分钟的时间才能使用。因此，用户无法访问应用程序的时间增加。

1. 启用可扩展性 (2/2)

确保您的架构能够应对需求的变化。



© 2023, Amazon Web Services, Inc. 或其附属公司。保留所有权利。

24

通过启用可扩展性，您可以改进设计，以预测对更多容量的需求，并提前提供这些容量。

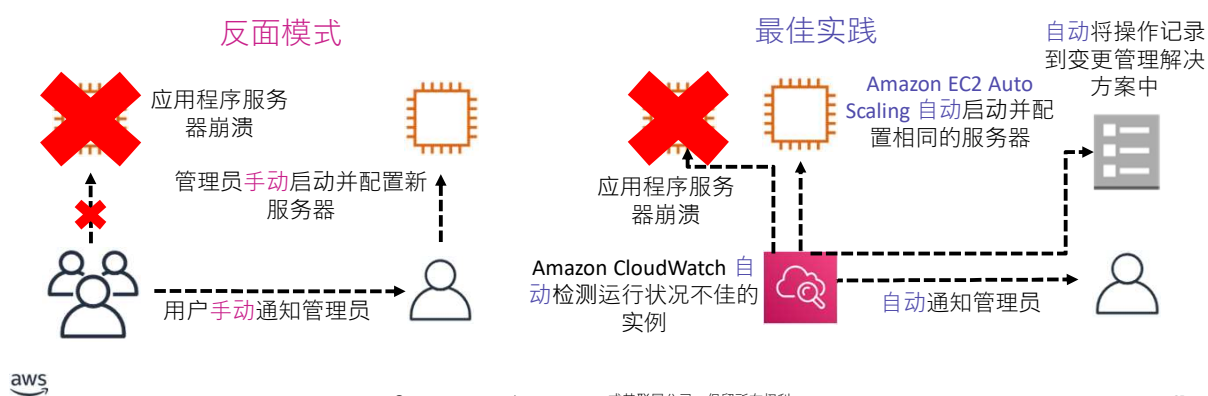
例如，您可以使用监控解决方案（如 Amazon CloudWatch）来检测服务器队列的总负载是否达到指定的阈值。您可以将该阈值定义为 *CPU 利用率持续高于 60% 超过 5 分钟*，或者与资源使用有关的任何内容。借助 CloudWatch，您还可以根据特定应用程序来设计自定义指标，从而触发所需的资源扩展。

触发警报后，Amazon EC2 Auto Scaling 会立即启动新实例。在达到容量之前，该实例便已准备就绪，从而为用户提供了无缝体验。

理想情况下，您还应该将此系统设计为能够在需求下降时 *缩减容量*，这样您就不会运行（和支付）不再需要的实例。

2. 让您的环境实现自动化

在可能的情况下，实现资源预置、终止和配置的自动化。



AWS 几乎在基础设施的每一层都提供内置的监控和自动化工具。利用这些工具，确保您的基础设施能够快速应对变化。

您可以使用诸如 CloudWatch 和 Amazon EC2 Auto Scaling 等工具来检测运行状况不佳的资源，并自动启动替换资源。当资源分配发生变化时，您还可以收到通知。

3. 将资源视为一次性资源

利用云计算的动态预置特性。

反面模式

- 随着时间的推移，不同的服务器最终具有不同的配置
- 资源在不需要时依然运行
- 硬编码的 IP 地址会妨碍灵活性
- 在正在使用的硬件上测试新的更新可能很困难或不方便

最佳实践

- 自动部署具有相同配置的新资源
- 终止未使用的资源
- 自动切换到新的 IP 地址
- 测试新资源的更新，然后用更新的资源替换旧资源



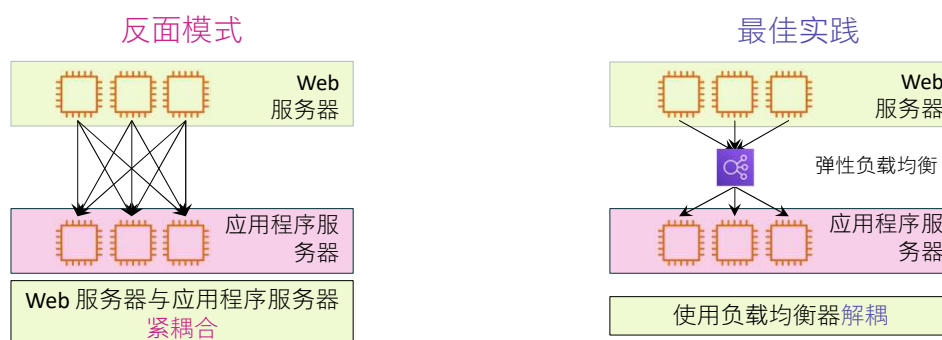
将资源视为一次性资源的最佳做法是指将基础设施视为软件而非硬件。

对于硬件，很容易购买比所需数量更多的特定组件，以便为使用高峰做好准备。这样做既昂贵又缺乏灵活性 - 由于沉没成本，升级难度更大。

相反，当您将资源视为一次性资源时，在实例或其他离散资源之间进行迁移就相当简单。您可以快速应对容量需求的变化、升级应用程序并管理底层软件。

4. 使用松耦合的组件

使用独立的组件设计架构。



传统的基础设施由紧密集成的服务器链组成，每个服务器都有特定的用途。问题是，一旦其中一个组件或层出现故障，对系统造成的破坏可能是致命的。还会妨碍扩展。如果在某一层添加或删除服务器，还必须连接每个连接层上的每个服务器。

左侧的示例显示了紧耦合的 Web 服务器和应用程序服务器的集合。如果一台应用程序服务器宕机，Web 服务器在尝试连接它时就会出错。

通过松耦合，您可以使用托管解决方案作为系统各层之间的中间件。通过这种设计，中间件可以自动处理故障以及组件或层的扩展。

右侧的示例显示了在 Web 服务器和应用服务器之间路由请求的负载均衡器（本例中为弹性负载均衡器）。如果一台应用程序服务器宕机，负载均衡器会自动开始将所有流量定向到两台运行正常的服务器。

用于解耦组件的两个主要解决方案是**负载均衡器**和**消息队列**。

5. 设计服务，而不是服务器

使用各种 AWS 服务。不要将基础设施局限于服务器。

反面模式

- 简单的应用程序在持久性服务器上运行
- 应用程序之间直接通信
- 静态 Web 资产本地存储在实例上
- 后端服务器处理用户身份验证和用户状态存储

最佳实践

- 适当时，考虑使用容器或无服务器解决方案
- 消息队列处理应用程序之间的通信
- 静态 Web 资产存储在外部，例如存储在 Amazon Simple Storage Service (Amazon S3) 上
- 用户身份验证和用户状态存储均由托管的 AWS 服务进行处理



下一个最佳实践是设计服务，而不是服务器。虽然 Amazon Elastic Compute Cloud (Amazon EC2) 为设计和设置解决方案提供了极大的灵活性，但它并非始终是满足每种需求的优先（或唯一的）解决方案。在某些情况下，容器或无服务器解决方案可能更合适。因此，重要的是要考虑您的需求是什么，以及哪种解决方案是合适的。

借助 AWS 无服务器解决方案和托管服务，您无需预置、配置和管理整个 Amazon EC2 实例。

托管解决方案具有更低配置文件和更高性能，可以更低成本替代基于服务器的解决方案。示例包括 AWS Lambda、Amazon Simple Queue Service (Amazon SQS)、Amazon DynamoDB、弹性负载均衡、Amazon Simple Email Service (Amazon SES) 和 Amazon Cognito。

6. 选择正确的数据库解决方案

*将技术与工作负载相匹配,
而不是反之。*

要考虑的事项：

- 读取和写入需求
- 总存储要求
- 典型的对象大小和这些对象的访问性质
- 持久性要求
- 延迟要求
- 支持的最大并发用户数
- 查询性质
- 完整性控制所需的强度

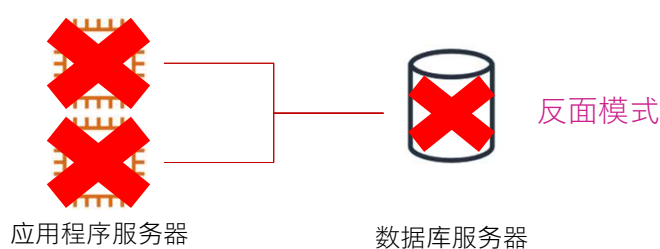


请务必选择正确的数据库解决方案。在传统的数据中心和本地部署环境中，可用硬件和许可证的限制会制约您对数据存储解决方案的选择。AWS 建议您根据对应用程序环境的需求选择数据存储。

7. 避免单点故障 (1/2)

假设一切都可能失败。
然后，逆向设计。

尽可能地使用冗余，以防止单点故障
导致整个系统崩溃。

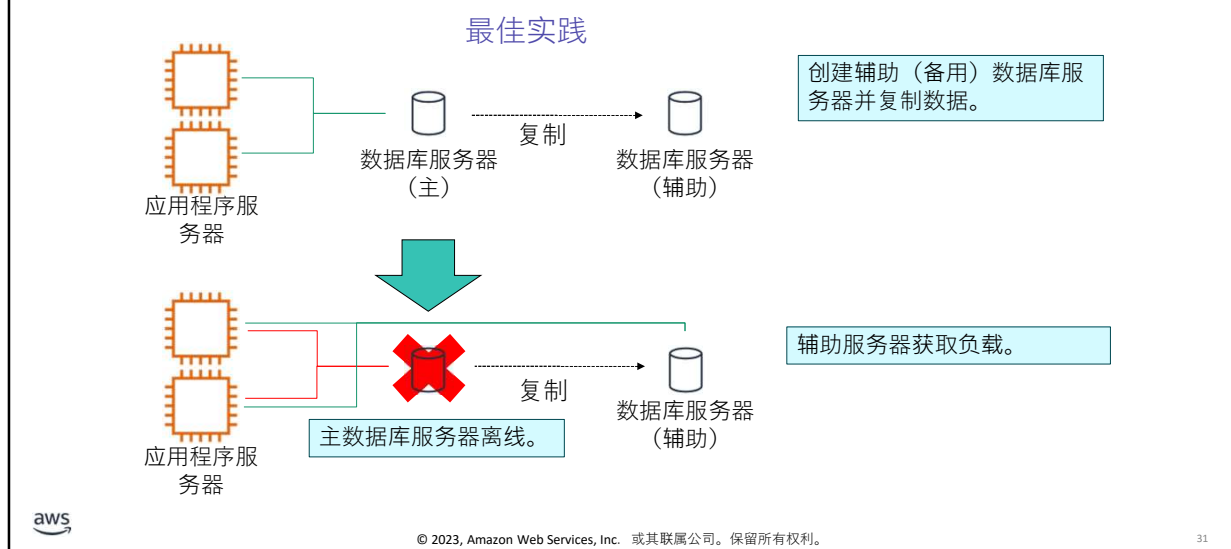


尽可能消除架构中的单点故障。这并不意味着您必须总是复制每个组件。根据停机时间服务级别协议（SLA），您可以使用仅在需要时启动组件的自动化解决方案。您还可以使用托管服务，在此服务中，AWS 会自动为您替换出现故障的底层硬件。

这个简单的系统显示了连接到单个数据库服务器的两个应用程序服务器。数据库服务器是单点故障，应避免使用。当它宕机时，应用服务器也会宕机。

即使底层物理硬件发生故障、被移除或更换，应用程序服务器也应继续正常运行。

7. 避免单点故障 (2/2)



避免单点故障的常见方法是，创建辅助（备用）数据库服务器并复制数据。这样，如果主数据库服务器离线，辅助服务器就能承担起负载。

在此示例中，当主数据库离线时，应用程序服务器会自动将请求发送到备用数据库。此示例也体现了最佳实践 3：将资源视为一次性资源，并设计应用程序以支持硬件变化。

8. 优化成本

利用 AWS 的灵活性来提高成本效益。

要考虑的事项：

- 我的资源的规模和类型是否适合任务需要？
- 应该监控哪些指标？
- 如何确保关闭未使用的资源？
- 我需要多久使用一次该资源？
- 是否可以用托管服务替代任何服务器？



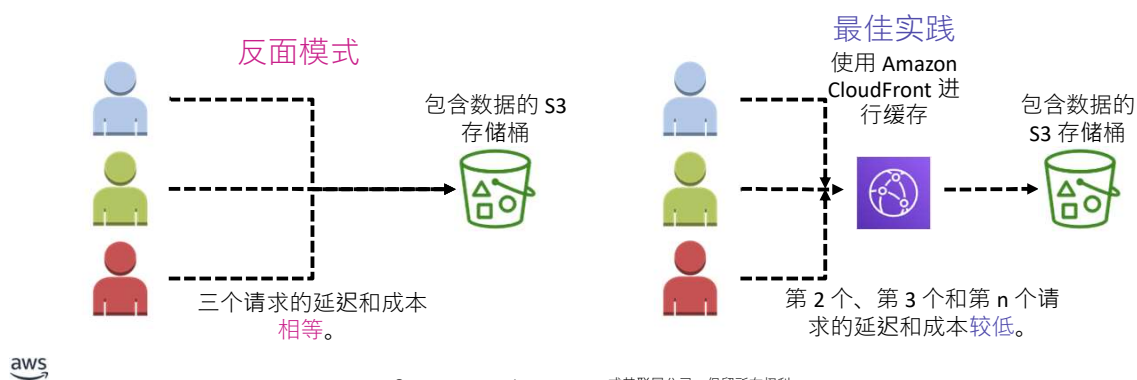
云计算让您可以用资本支出换取可变支出。*资本支出 (capex)* 是公司用于购置、升级和维护不动产、工业建筑或设备等实体资产的资金。在这种模式下，无论数据中心的服务器是否处于运行状态，您都要为其付费。

相比之下，AWS 服务采用的是 *可变支出* 成本模式，这意味着您只需根据服务的使用量，为所需的单项服务付费。在每项服务中，您都可以优化成本。许多服务提供不同的定价套餐、模型或配置。

请记住，要在云中复制本地部署中服务器全天候运行的数据中心设置，费用可能会非常高。因此，从成本角度来看，构建基础设施的最佳方法是仅预置所需的资源，并在不使用时停止服务。

9. 使用缓存

缓存最大限度地减少了冗余数据检索操作，
优化了性能和成本。



缓存是一种技术，通过在请求者和永久存储之间的中间位置临时存储数据，使未来的请求更快，并减少网络吞吐量。

在反面模式示例中，没有使用缓存服务。当任何人从 Amazon Simple Storage Service (Amazon S3) 的一个存储桶中请求文件时，每次请求都需要相同的时间来完成，每次请求的成本也相同。

在最佳实践模式示例中，基础设施在 Amazon S3 之前使用 Amazon CloudFront 提供缓存。在这种情况下，初始请求会检查 Amazon CloudFront 中的文件。如果找不到，CloudFront 会从 Amazon S3 请求文件。然后，CloudFront 将文件副本存储在靠近用户的边缘站点，并将副本发送给提出请求的用户。对文件的后续请求将从 CloudFront 中的（现在更接近的）边缘站点而不是 Amazon S3 中检索。

这样可以减少延迟和成本，因为在第一次请求之后，无需再为从 Amazon S3 传输的文件付费。

10. 保护整个基础设施

在基础设施的每一层构建安全性。

要考虑的事项：

- 隔离基础设施的各个部分
- 对传输中的数据和静态数据进行加密
- 使用最低权限原则精细地实施访问控制
- 使用多重身份验证 (MFA)
- 使用托管服务
- 记录资源的访问情况
- 自动执行部署以保持一致的安全性



安全性不仅仅是通过基础设施的外部边界。这还包括确保各个环境及其组件相互安全。

例如，在 Amazon EC2 中，您可以创建安全组来确定实例上哪些端口可以发送和接收流量。安全组还可以确定流量的来源或去向。

您可以使用安全组来降低一个实例上的安全威胁扩散到环境中其他实例的可能性。您应该对其他服务采取类似的预防措施。整个课程都会讨论实施这一最佳实践的具体方法。

第 3 节要点



- 在设计解决方案时，评估权衡并根据经验数据作出决策
- 在 AWS 上构建解决方案时遵循这些最佳实践 –
 - 启用可扩展性
 - 让您的环境实现自动化
 - 将资源视为一次性资源
 - 使用松耦合的组件
 - 设计服务，而不是服务器
 - 选择正确的数据库解决方案
 - 避免单点故障
 - 优化成本
 - 使用缓存
 - 保护整个基础设施

本模块中这节内容的要点包括：

- 在设计解决方案时，评估权衡并根据经验数据作出决策
- 在 AWS 上构建解决方案时遵循这些最佳实践 –
 - 启用可扩展性
 - 让您的环境实现自动化
 - 将资源视为一次性资源
 - 使用松耦合的组件
 - 设计服务，而不是服务器
 - 选择正确的数据库解决方案
 - 避免单点故障
 - 优化成本
 - 使用缓存
 - 保护整个基础设施

第 4 节：AWS 全球基础设施

模块 2：云架构简介



© 2023, Amazon Web Services, Inc. 或其联属公司。保留所有权利。

介绍第 4 节：AWS 全球基础设施。

AWS 区域

- AWS 区域是一个地理区域
- 每个 AWS 区域都由两个或更多可用区组成
- 区域之间的通信使用 AWS 骨干网络基础设施
- 您可以实现和控制跨区域数据复制



示例：伦敦区域



AWS 云基础设施是围绕区域构建的。AWS 在全球有 22 个区域。AWS 区域是一个实际的地理位置，拥有两个或多个可用区。可用区又由一个或多个数据中心组成。

AWS 区域连接到多个网络服务提供商 (ISP)。区域还与一个专用全球骨干网连接。与公共互联网相比，专用全球骨干网的费用更低，跨区域网络延迟更稳定。

2019 年 3 月 20 日之前推出的 AWS 区域默认处于启用状态。2019 年 3 月 20 日之后推出的区域（例如，亚太地区（香港）和中东（巴林））默认处于禁用状态。您必须先启用这些区域，然后才能使用它们。您可以使用 AWS 管理控制台来启用或禁用区域。

某些区域的访问受限。通过 AWS（中国）账户只能访问北京区域和宁夏区域。要了解有关 AWS 中国的更多信息，请参阅“[AWS 中国](#)”页面。隔离的 AWS GovCloud (US) 区域专门面向美国政府机构和客户，方便他们将敏感工作负载移至云中，从而满足其特定的法规和合规性要求。

为实现容错能力和稳定性，区域之间相互隔离。一个区域中的资源不会自动复制到其他区域。在特定区域存储数据时，数据不会复制到该区域之外。如果您的业务需要，您应自行负责在多个区域间复制数据。AWS 提供有关每个区域所在的国家/地区和省/市/自治区的信息（如适用）。您有责任根据合规性和网络延迟要求，选择存储数据的区域。

AWS 产品和服务按区域提供，因此各个区域提供的服务可能不尽相同。有关按区域提供的 AWS 服务的列表，请参阅[区域表](#)。

有关 AWS 全球云基础设施的更多信息，请参阅[全球基础设施网站](#)。有关 AWS 全球基础设施的最新互动式地图，请参阅[互动式 AWS 全球基础设施地图](#)。

AWS 可用区

- 每个可用区都具有以下特点 –
 - 由一个或多个数据中心组成
 - 专为故障隔离而设计
 - 使用高速专用链接与区域中的其他可用区互连
- 对于某些服务，您可以选择自己的可用区
- AWS 建议跨可用区进行复制，以获得弹性



每个 AWS 区域都有两个或多个相互隔离的位置，称为 *可用区*。每个可用区都包含一个或多个数据中心，某些可用区拥有多达六个数据中心。但是，每个数据中心只能属于一个可用区。

每个可用区都被设计为独立的故障区。这意味着可用区在典型的大都市区域内是物理隔绝的，并且位于风险较低的洪泛平原上（具体的洪泛区分类因区域而异）。除了具有分立的不间断电源和现场备用发电设施外，它们还分别通过独立公用事业公司的不同电网供电，以便进一步减少单点故障。可用区全部以冗余的方式连接至多家第 1 层传输提供商。

可用区是您可以为某些服务（例如 Amazon EC2）制定的最精细级别的规范。

您负责选择自己的系统所在的可用区。系统可以跨越多个可用区。您的系统应设计为，在灾难发生时，能够承受暂时或长期的可用区故障。将应用程序分布在多个可用区内，可使应用程序在大多数故障情况下（包括自然灾害或系统故障）保持弹性。

AWS Local Zones

- 使您能够在距离终端用户更近的特定地理位置，运行应用程序中对延迟敏感的部分和资源。
- 是 AWS 区域的扩展，您可以在靠近终端用户的地理位置上使用 AWS 服务
- 让您可以将 AWS 计算、存储、数据库和其他特定服务放在更靠近目前不存在 AWS 区域的大量人口聚居的位置，或者靠近行业 and IT 中心的位置
- 由 AWS 管理和支持
- 洛杉矶 (LA) AWS Local Zone 可应邀使用



AWS Local Zones 是 AWS 基础设施部署的一种新形式，可将 AWS 计算、存储、数据库和其他特定服务放在更靠近目前不存在 AWS 区域的大量人口聚居的位置，或者靠近行业 and IT 中心的位置。借助 AWS Local Zones，您能够在距离终端用户更近的特定地理位置，运行应用程序中对延迟敏感的部分和资源。您可以使用 AWS Local Zones 为媒体和娱乐内容创建、实时游戏、油藏模拟、Electronic Design Automation 和机器学习等使用案例提供个位数毫秒的延迟。

每个 AWS Local Zone 位置都是 AWS 区域的扩展。您可以在 AWS Local Zone 中运行对延迟敏感的应用程序，方法是在终端用户附近使用 AWS 服务，如 Amazon EC2、Amazon Virtual Private Cloud (Amazon VPC)、Amazon Elastic Block Store (Amazon EBS)、Amazon FSx 和弹性负载均衡。AWS Local Zones 在本地工作负载和在 AWS 区域运行的工作负载之间提供高带宽、安全的连接。因此，AWS Local Zones 使您能够无缝连接回 AWS 中运行的其他工作负载，并通过相同的 API 和工具集连接到区域内的所有服务。

AWS Local Zones 由 AWS 管理和支持，并为您提供云的所有弹性、可扩展性和安全性优势。借助 AWS Local Zones，您可以使用一套一致的 AWS 服务，在更靠近终端用户的地方构建和部署对延迟敏感的应用程序。您还可以扩大或缩小规模，而且只需为使用的资源付费。

目前，洛杉矶 AWS Local Zone 一般通过邀请即可使用，您可以期待更多 Local Zone 的出现。

要了解有关 AWS Local Zones 的更多信息，请参阅 [AWS Local Zones 页面](#)。

AWS 数据中心

- 数据中心是存放和处理数据的位置
- 一个数据中心通常包含上万台服务器
- 所有数据中心都是在线的，并为客户服务
- AWS 自定义网络设备 –
 - 来自多个 ODM
 - 拥有自定义网络协议堆栈



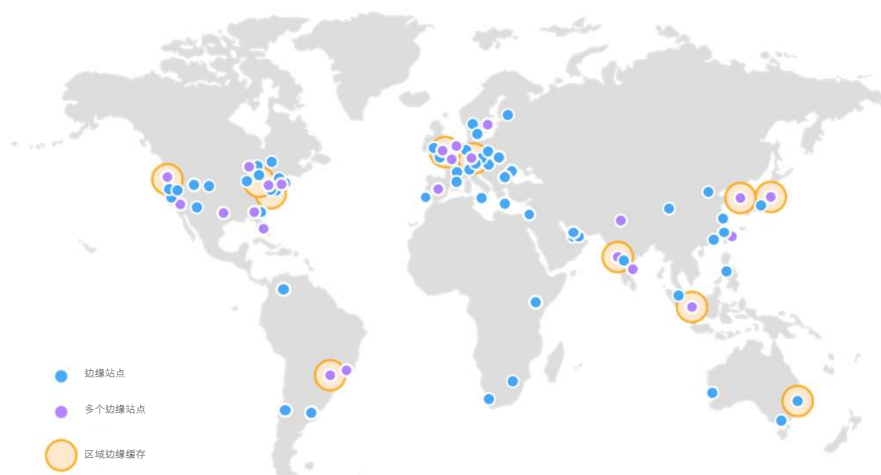
AWS 基础设施的基础是数据中心。您不指定用于部署资源的数据中心。数据中心则是实际数据的存放位置。Amazon 运行着具有高可用性的先进数据中心。尽管数据中心有时会发生一些故障，影响同一站点内实例的可用性，但是这种故障极少发生。如果您将所有实例都托管在受此类故障影响的同一个站点，一旦发生故障，您的所有实例都将不可用。

所有数据中心都是在线的，并为客户服务。如果出现故障，自动进程会将客户数据流量从受影响的区域移除。核心应用程序以 N+1 的配置进行部署，故在数据中心出现故障时，有足够的容量使流量能够均衡加载至其余的站点。

AWS 使用的自定义网络设备来自多个原始设备制造商 (ODM)。ODM 根据另外一家公司提供的规范来设计和制造产品。然后，这家公司在更换产品的品牌后进行销售。

有关 AWS 数据中心的更多信息，请参阅[了解我们如何通过设计来保护 AWS 数据中心](#)。

AWS 接入点



© 2023, Amazon Web Services, Inc. 或其联属公司。保留所有权利。

41

为了以较低的延迟向终端用户交付内容，Amazon CloudFront 使用了包含 200 多个接入点的全球网络，这些接入点由边缘站点和区域边缘缓存组成。

边缘站点位于北美、欧洲、亚洲、澳大利亚、南美洲、中东、非洲和中国。边缘站点支持 Amazon Route 53 和 Amazon CloudFront 等 AWS 服务。

默认情况下，区域边缘缓存与 Amazon CloudFront 一起使用。当您的内容访问频率不够高，无法保留在边缘站点时，可以使用区域边缘缓存。区域边缘缓存会保留这些内容，在用户必须从源服务器获取内容时提供一份替代方案。

有关 Amazon CloudFront 基础设施的更多信息，请参阅 **Amazon CloudFront 基础设施**，网址为 https://aws.amazon.com/cloudfront/features/?whats-new-cloudfront.sort-by=item.additionalFields.postDateTime&whats-new-cloudfront.sort-order=desc#Amazon_CloudFront_Infrastructure

第 4 节要点

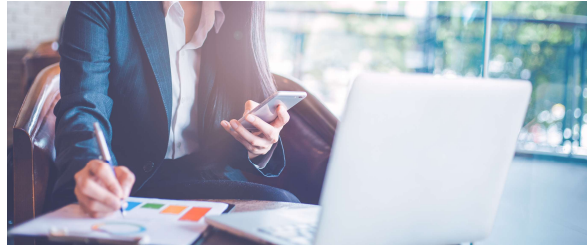


- AWS 全球基础设施由区域、可用区和边缘站点组成
- 您通常按照合规性要求或以减少延迟为原则来确定自己的首选区域
- 每个可用区都在物理上与其他可用区分开并具有冗余电源、网络 and 连接
- 边缘站点和区域边缘缓存通过将内容缓存到离用户更近的位置来提高性能

本模块中这节内容的要点包括：

- AWS 全球基础设施由区域和可用区组成
- 您通常按照合规性要求或以减少延迟为原则来确定自己的首选区域
- 每个可用区都在物理上与其他可用区分开，并具有冗余电源、网络 and 连接
- 边缘站点和区域边缘缓存通过将内容缓存到离用户更近的位置来提高性能

Course capstone project



The capstone project provides you with the opportunity to apply the skills and knowledge you develop in the course to a real-world scenario. Details about the project are included in the Bridging to certification module.

模块总结

模块 2：云架构简介



© 2023, Amazon Web Services, Inc. 或其联属公司。保留所有权利。

现在，我们来回顾和总结一下本模块，然后进行知识考核。

模块总结

总的来说，在本模块中，您学习了如何：

- 定义云架构
- 描述如何使用 AWS Well-Architected Framework 来设计和评估架构
- 解释在 AWS 上构建解决方案的最佳实践
- 描述如何就 AWS 资源的放置位置作出明智的决策



总的来说，在本模块中，您学习了如何：

- 定义云架构
- 描述如何使用 AWS Well-Architected Framework 来设计和评估架构
- 解释在 AWS 上构建解决方案的最佳实践
- 描述如何就 AWS 资源的放置位置作出明智的决策

完成知识考核



现在该完成本模块的知识考核了。

其他资源

- [AWS 全球基础设施页面](#)
- [互动式 AWS 全球基础设施地图](#)
- [AWS Well-Architected Framework 白皮书](#)
- [安全性支柱白皮书](#)
- [卓越运营支柱白皮书](#)
- [可靠性支柱白皮书](#)
- [性能效率支柱白皮书](#)
- [成本优化支柱白皮书](#)
- [可持续性支柱白皮书](#)



如果您想进一步了解本模块中涵盖的主题，以下额外资源可能会对您有所帮助：

- [AWS 全球基础设施页面](#)
- [互动式 AWS 全球基础设施地图](#)
- [AWS Well-Architected Framework 白皮书](#)
- [安全性支柱白皮书](#)
- [卓越运营支柱白皮书](#)
- [可靠性支柱白皮书](#)
- [性能效率支柱白皮书](#)
- [成本优化支柱白皮书](#)
- [可持续性支柱白皮书](#)



感谢您完成本模块的学习。