欢迎学习模块 12：构建解耦架构。

模块概览

aws academy

小节目录

1. 架构需求
2. 解耦架构
3. 使用 Amazon Simple Queue Service (Amazon SQS) 进行解耦
4. 使用 Amazon Simple Notification Service (Amazon SNS) 进行解耦
5. 使用 Amazon MQ 在云应用程序和本地之间发送消息

知识测验

本模块包含以下章节：

1. 架构需求
2. 解耦架构
3. 使用 Amazon Simple Queue Service (Amazon SQS) 进行解耦
4. 使用 Amazon Simple Notification Service (Amazon SNS) 进行解耦
5. 使用 Amazon MQ 在云应用程序和本地之间发送消息

在本模块结束时，您需要完成一个知识测验，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：

- 区分紧耦合架构和松耦合架构
- 确定 Amazon SQS 的工作原理以及使用时间
- 确定 Amazon SNS 的工作原理以及使用时间
- 描述 Amazon MQ

学完本模块后，您应该能够：

- 区分紧耦合架构和松耦合架构
- 确定 Amazon SQS 的工作原理以及使用时间
- 确定 Amazon SNS 的工作原理以及使用时间
- 描述 Amazon MQ

模块 12：构建解耦架构

第 1 节：架构需求

介绍第 1 节：架构需求。

咖啡馆业务要求

咖啡馆的架构现在支持成千上万的用户。但是，很难对应用程序的一个层进行更改而不影响其他层。

咖啡馆的架构现在支持成千上万的用户。但是，咖啡馆的系统耦合过于紧密。很难对应用程序的一个层进行更改而不影响其他层。例如，每日订购报告是从同一个 Web 服务器生成的，该服务器也为客户提供咖啡馆的网站。

此外，Frank 提到，他没有收到周五 17:00 的定期报告。经过一系列调查，Sofía 和 Nikhil 发现，计划维护时段与报告系统尝试生成报告的时间相吻合。

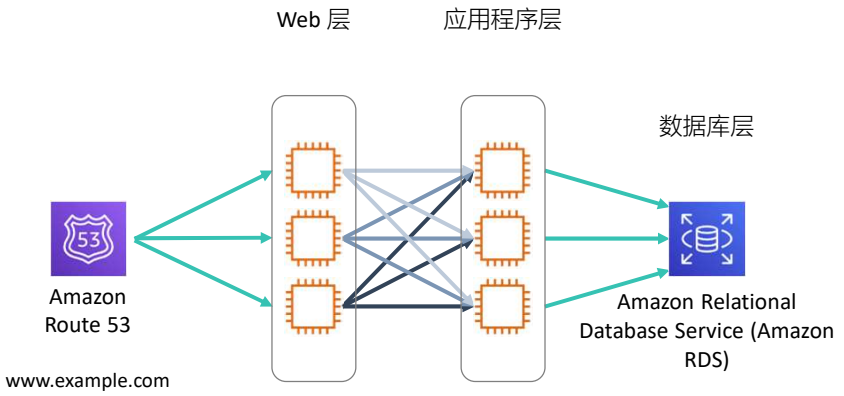
他们与 Olivia 交谈，Olivia 建议他们解耦架构。通过将报告流程移动到另一个系统，即使 Web 服务器暂时不可用，报告数据也不会丢失。此外，生成报告的请求将排队并进行处理。

模块 12：构建解耦架构

第 2 节：解耦架构

介绍第 2 节：解耦架构。

紧耦合的架构

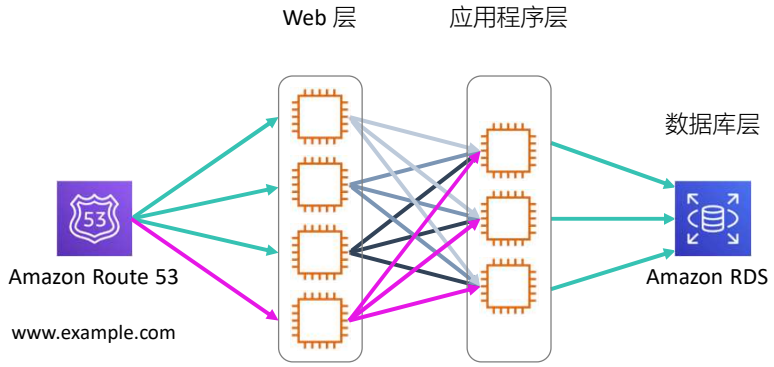


组件彼此之间紧密相连。

传统基础设施具有紧密集成的组件链。每个组件都有特定的目的。当一个组件出现故障时，系统中断可能是致命的。

考虑这个处理客户订单的 Web 应用程序的三层架构示例。Web 层中的每个实例都与应用程序层中的每个实例进行通信。应用程序层中的每个实例都将数据保存到后端数据库中。Web 或应用程序层中的实例故障还会导致某些客户订单数据的持久性失败。

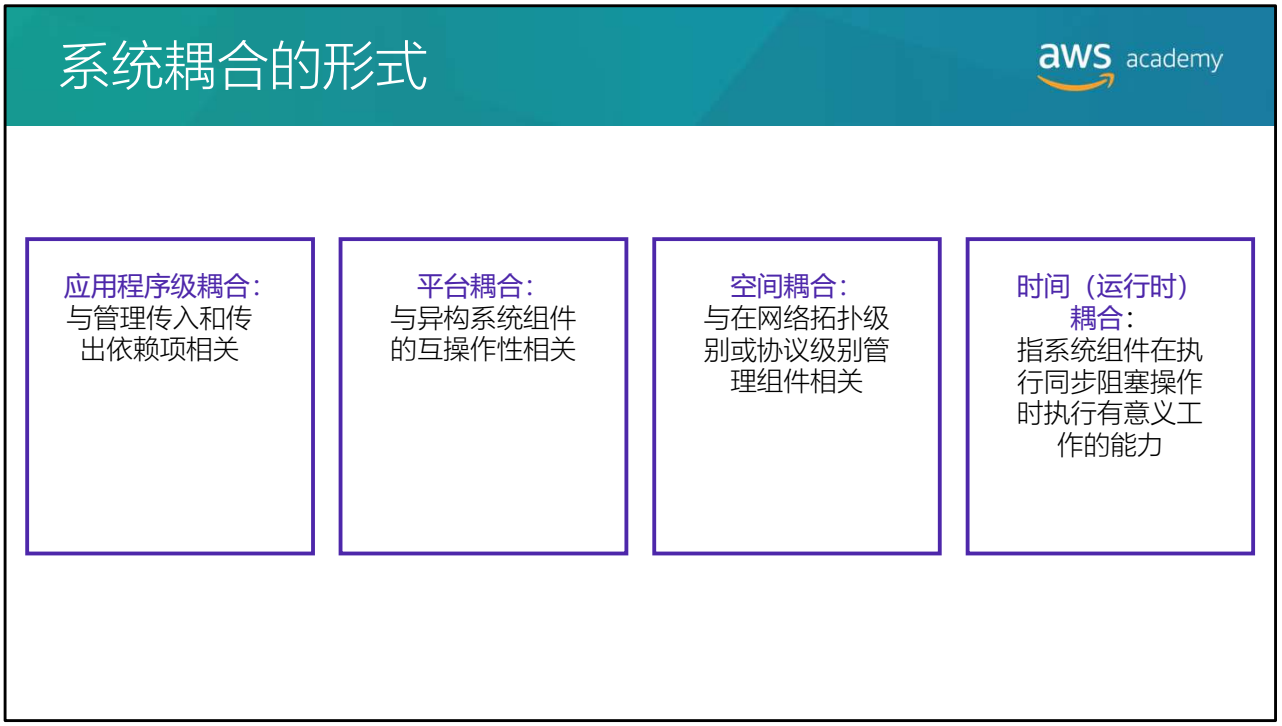
紧耦合的架构阻碍扩展



添加资源会增加复杂性并阻碍扩展。

在紧耦合的系统中，扩展也会受到阻碍：如果在一个层添加服务器，则还必须将其连接到每个连接层中的服务器。继续三层架构示例，添加到 Web 层的实例必须连接到应用程序层中的每个实例。

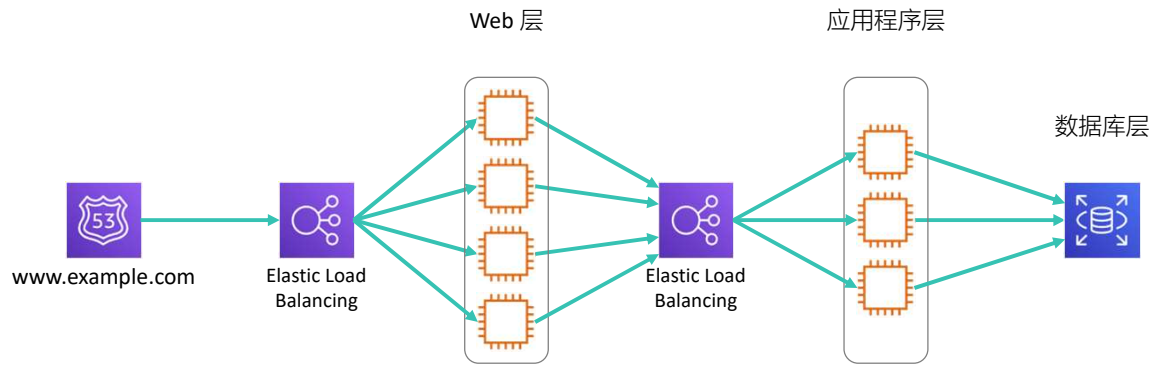
系统耦合的形式



系统可以通过多种方式进行耦合，在云中构建分布式应用程序时，应考虑这些因素：

- *应用程序级*耦合与管理传入和传出依赖项相关
- *平台*耦合与异构系统组件的互操作性相关
- *空间*耦合与在网络拓扑级别或协议级别管理组件相关
- *时间*（或运行时）耦合指系统组件在执行同步阻塞操作时执行有意义工作的能力

松耦合架构

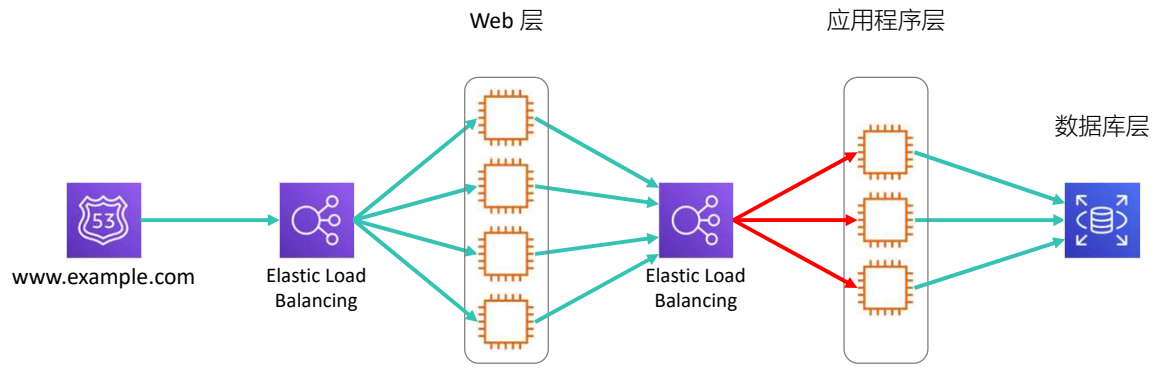


您可以使用托管解决方案作为各层之间的中间件。

为了帮助确保应用程序随着负载的增加而扩展，并确保系统不存在瓶颈或单点故障，请实施松耦合。通过松耦合，您可以通过将托管解决方案作为系统各层之间的中间件来减少系统中的依赖项。这样一来，中间件会自动处理组件或层发生的故障和扩展。

再次考虑三层 Web 应用程序架构。您可以通过在 Web 和应用程序层前添加负载均衡器来实现松耦合，以便在每层分配流量。如果一台服务器出现故障，负载均衡器将自动将流量定向到运行良好的实例。

松耦合架构中的注意事项



在订单处理工作流的业务使用案例中，一个潜在的漏洞点是将订单数据保存到数据库。如果业务要求订单数据必须保留在数据库中，那么各种情况（例如潜在的死锁、竞争条件或网络问题）都可能会导致订单持久性失败。在这种情况下，订单将丢失，您无法恢复。

第 2 节要点

- 紧耦合的系统具有紧密集成的组件链，会阻碍扩展
- 您可以使用托管解决方案（例如 Elastic Load Balancing）各层之间的中间件，在系统中实施松耦合

本模块中这节内容的要点包括：

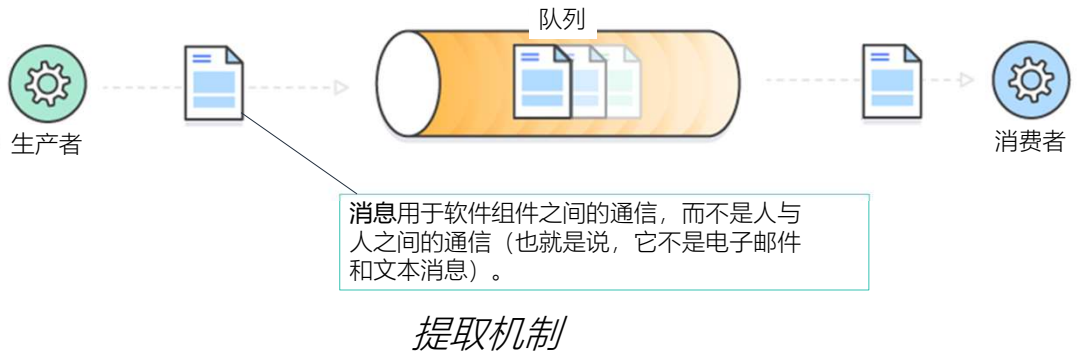
- 紧耦合的系统具有紧密集成的组件链，会阻碍扩展
- 您可以使用托管解决方案（例如 Elastic Load Balancing）各层之间的中间件，在系统中实施松耦合

模块 12：构建解耦架构

第 3 节：使用 Amazon SQS 进行解耦

介绍第 3 节：使用 Amazon SQS 进行解耦。

解耦架构的消息队列



提取机制

消息队列是帮助您实现解耦架构的另一个组件。消息队列可为这些分布式应用程序提供通信和协调。

消息队列是等待处理的消息的临时储存库。这些消息通常较小，可以是请求、回复、错误消息或明文信息等。消息示例包括客户记录、产品订单、发票、患者记录等。

要发送消息时，一个名为 *“生产者”* 的组件会将消息添加到队列。消息将存储在队列中，直至名为 *“消费者”* 的另一个组件检索并处理该消息。

Amazon SQS

- 完全托管的消息队列服务
- 使用提取机制
- 加密和存储消息，直至消息得到处理或被删除
- 用作生产者和消费者之间的缓冲区

Amazon Simple Queue Service (Amazon SQS) 是一项完全托管的消息队列服务，使您能够解耦应用程序组件，以便它们独立运行。它可以让 Web 服务应用程序对一个应用程序组件生成、由另一个组件使用的消息进行排队。

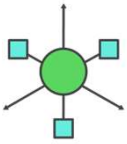
队列*是等待处理的消息的临时储存库。*存储消息，直至消息得到处理或被删除（从 1 到 14 天；默认为 4 天）。消息可包含最多 256KB 的任何格式的文本。Amazon SQS 可以大规模处理工作，每天处理数十亿条消息。它将所有消息队列和消息存储在具有多个冗余可用区的单个高度可用的 AWS 区域中。任何一台计算机、网络或可用区出现故障都不会使消息无法访问。可以同时发送和读取消息。

您可以安全地以匿名方式或与特定 AWS 账户共享 Amazon SQS 队列。您也可以通过 IP 地址和时间限制队列共享。SQS 队列中的消息通过 AWS Key Management Service (AWS KMS) 中托管的密钥使用服务器端加密 (SSE) 进行加密。Amazon SQS 仅在将消息发送给授权消费者时才会对消息进行解密。

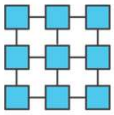
Amazon SQS 支持多个生产者和消费者与同一队列进行交互。Amazon SQS 可与多种 AWS 服务一起使用，包括：Amazon Elastic Compute Cloud (Amazon EC2)、Amazon Simple Storage Service (Amazon S3)、Amazon Elastic Container Service (Amazon ECS)、AWS Lambda 和 Amazon DynamoDB。

使用 Amazon SQS 实现松耦合

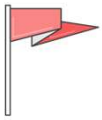
使用 Amazon SQS，您可以：



使用异步处理快速获取每个步骤的响应



通过增加作业实例的数量来处理性能和服务要求



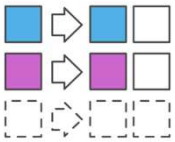
轻松从失败的步骤中恢复（因为消息将保留在队列中）

通过 Amazon SQS，您可以在架构中实现松耦合。

- 它执行异步处理，以便您可以快速获得每个步骤的响应
- 它可以通过增加作业实例的数量来应对性能和服务要求
- 您的应用程序能轻松从失败的步骤中恢复（因为消息将保留在队列中）

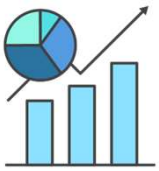
Amazon SQS 一般使用案例

工作队列



缓冲批处理操作

请求卸载

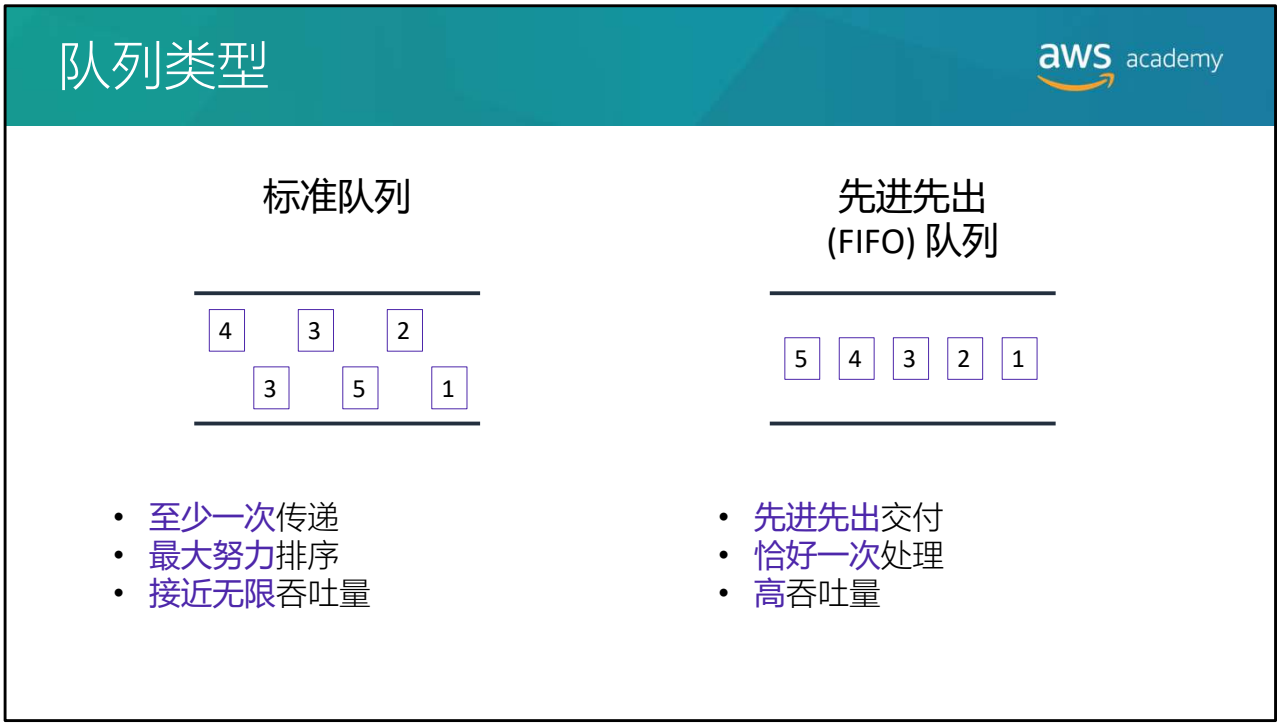


触发 Amazon EC2 Auto Scaling

Amazon SQS 有以下多种使用方式：

- *工作队列* – 对可能无法同时全部处理相同工作量的分布式应用程序的组件进行解耦。
- *缓冲批处理操作* – 为您的架构添加可扩展性和可靠性，并可消除临时卷峰值，而不会丢失消息或增加延迟。
- *请求卸载* – 将请求排入队列，从交互式请求路径中移出缓慢操作。
- *触发 Amazon EC2 Auto Scaling* – 使用 SQS 队列帮助确定应用程序的负载。当它们与 Amazon EC2 Auto Scaling 结合使用时，您可以根据流量规模向外或向内扩展 EC2 实例的数量。

队列类型



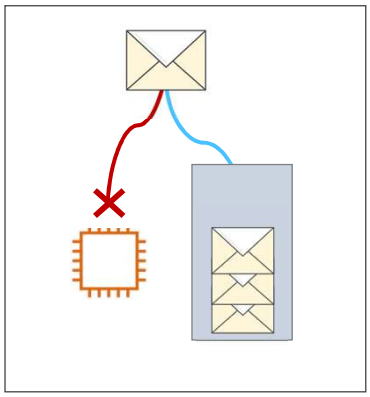
有两种 SQS 队列类型：

标准队列提供：
- 至少一次传递 – 消息至少传送一次，但偶尔会传送消息的多个副本。
- 最大努力排序 – 消息偶尔可能按不同于其发送时的顺序传递。
- 接近无限吞吐量 – 标准队列支持每个 API 操作几乎无限数量的每秒事务数 (TPS)。

先进先出 (FIFO) 队列：
- 旨在确保按照消息的发送和接收顺序对消息进行严格一次处理。
- 提供高吞吐量 – FIFO 队列每秒支持最多 300 条消息（每秒 300 次发送、接收或删除操作）。如果您每次操作批量处理 10 条消息（最多），那么 FIFO 队列每秒最多可支持 3000 条消息。

Amazon SQS 功能（第 1 页，共 3 页）

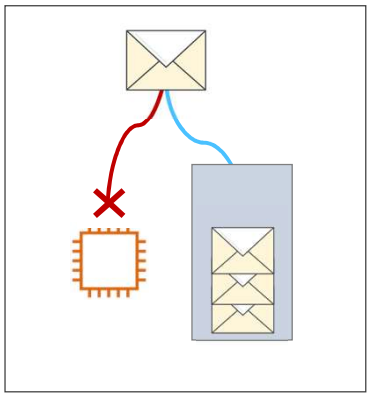
死信队列支持



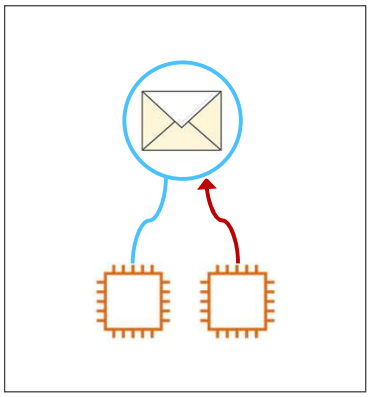
Amazon SQS 提供死信队列支持。*死信队列(DLQ) 指的是无法处理的消息队列。*它会在处理尝试次数达到最大值后接收消息。DLQ 跟任何其他 SQS 队列一样：可向其发送消息和从其中接收消息。您可以使用 Amazon SQS API 和控制台创建 DLQ。

Amazon SQS 功能（第 2 页，共 3 页）

死信队列支持



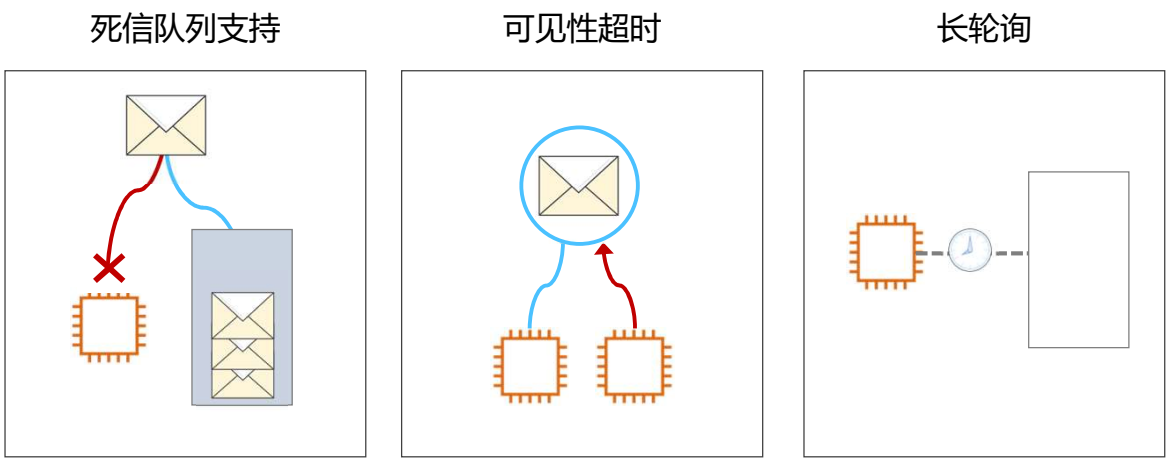
可见性超时



Amazon SQS 的另一个功能是可见性超时。*可见性超时*是 Amazon SQS 阻止其他消费者接收并处理相同消息的一段时间。超时有助于确保作业不会多次处理而导致重复。在可见性超时期间，收到消息的组件会先处理消息，然后将其从队列中删除。消息的默认可见性超时为 30 秒，最长为 12 小时。

如果消费者处理失败，没有在可见性超时到期之前删除消息，则该消息将对其他消费者可见，并且可以进行再次处理。通常情况下，您应将可见性超时设置为应用程序处理消息并将其从队列中删除所花费的最长时间。

Amazon SQS 功能（第 3 页，共 3 页）

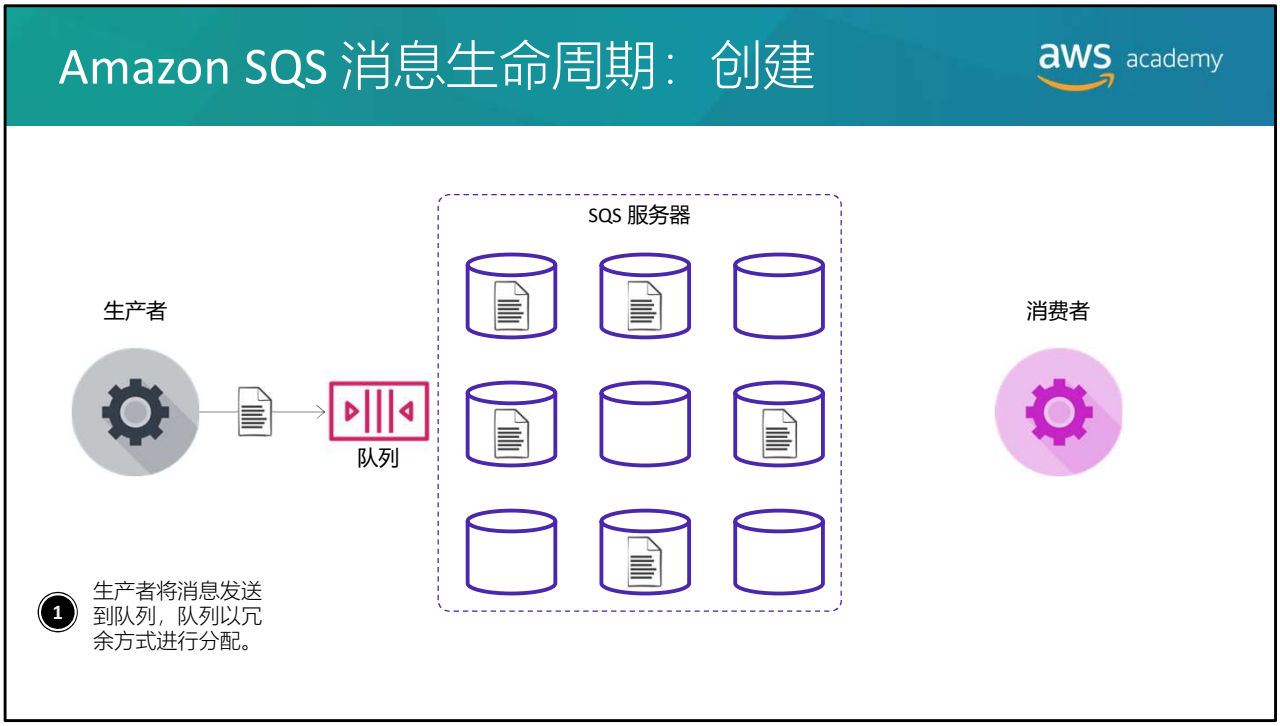


最后，Amazon SQS 支持短轮询和长轮询以从 SQS 队列检索消息。默认情况下，队列使用短轮询。

短轮询查询只能在响应中包含用于查找消息的服务器的子集（基于加权随机分布）。即使查询未发现消息，Amazon SQS 也会立即发送响应。

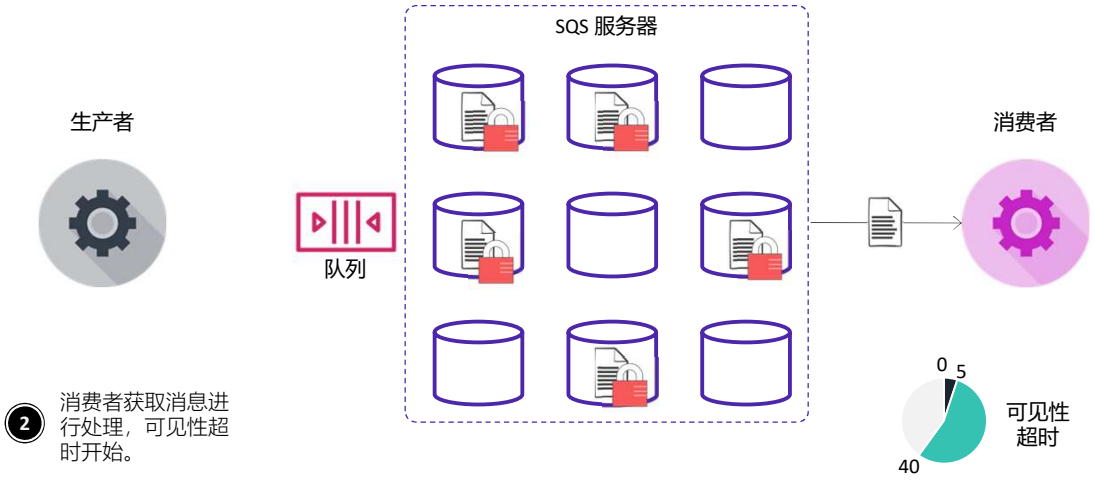
相比之下，长轮询会在所有服务器中查询消息。Amazon SQS 会在达到收集响应的最大消息数或轮询等待时间到期后发送响应。

如果要在消息可用后立即从 SQS 队列中检索消息，那么长轮询是一种成本较低的方式。因为您可以减少接收空消息的次数，所以长轮询可以降低 Amazon SQS 的使用成本。



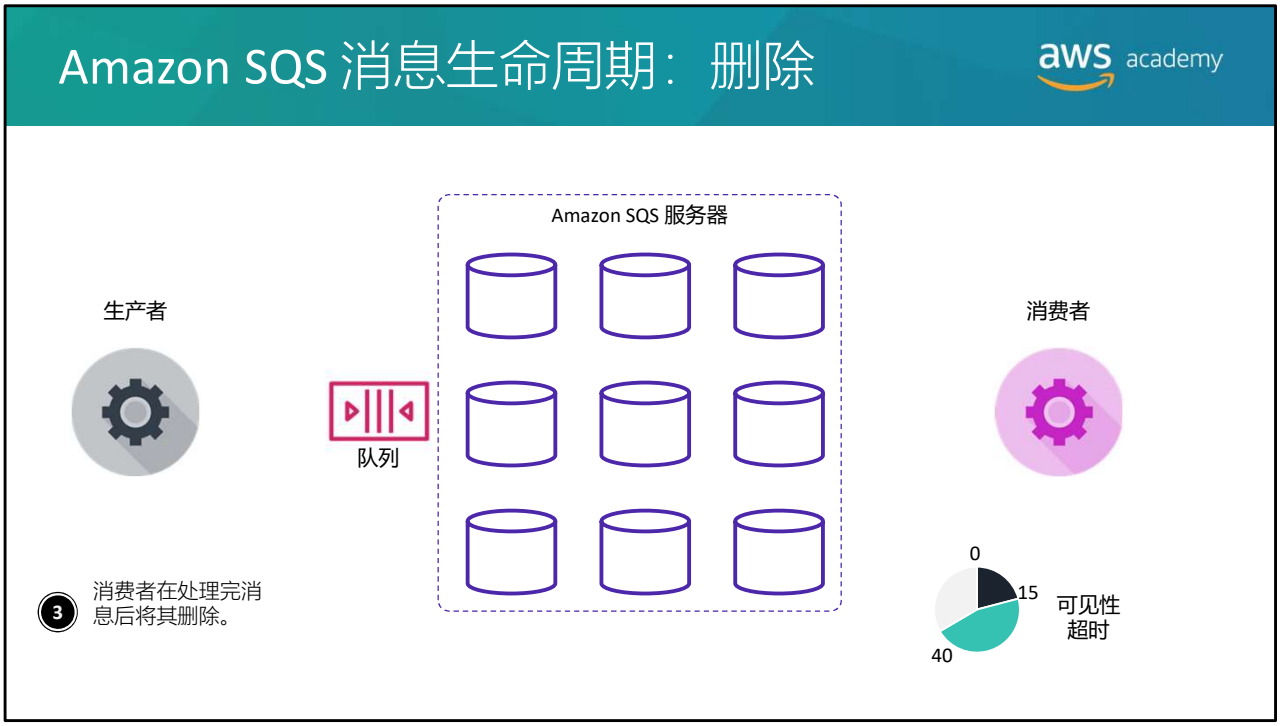
以下场景可以说明 SQS 队列中消息的生命周期。首先，有一个生产者向队列发送一条消息，消息以冗余方式跨 SQS 服务器分布。

Amazon SQS 消息生命周期：流程



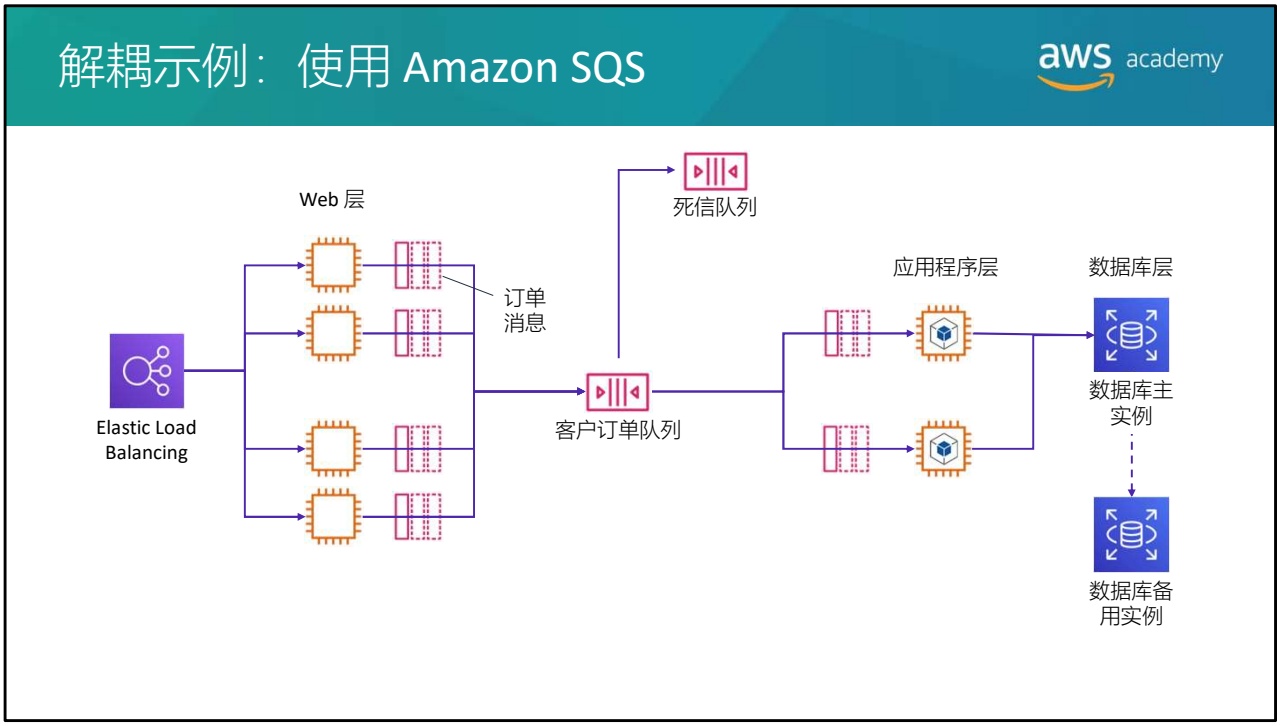
当消费者准备好处理消息时，会从队列中检索消息。消息接受处理期间仍将保留在队列中。

在可见性超时期间，其他消费者无法处理该消息。在此示例中，可见性超时为 *40 秒*。



处理消息后，消费者将从队列中删除该消息。此操作可阻止消息在可见性超时过期后被再次接收和处理。

Amazon SQS 不会自动删除消息。由于 Amazon SQS 是分布式系统，因此无法保证消费者实际收到消息（例如，由于连接问题或消费者应用程序中的问题而导致实际没有收到消息）。因此，消费者在接收和处理消息后必须将其从队列中删除。



再次考虑您在上一节中学到的订购处理应用程序。您可以通过引入 SQS 队列来解耦此应用程序的架构。您可以使用队列将处理逻辑隔离到其组件中，以便在一个独立于 Web 应用程序的进程中运行它。

这种设计使系统能够更好地应对流量激增。此外，它还使工作只能在管理成本所需的速度下进行。

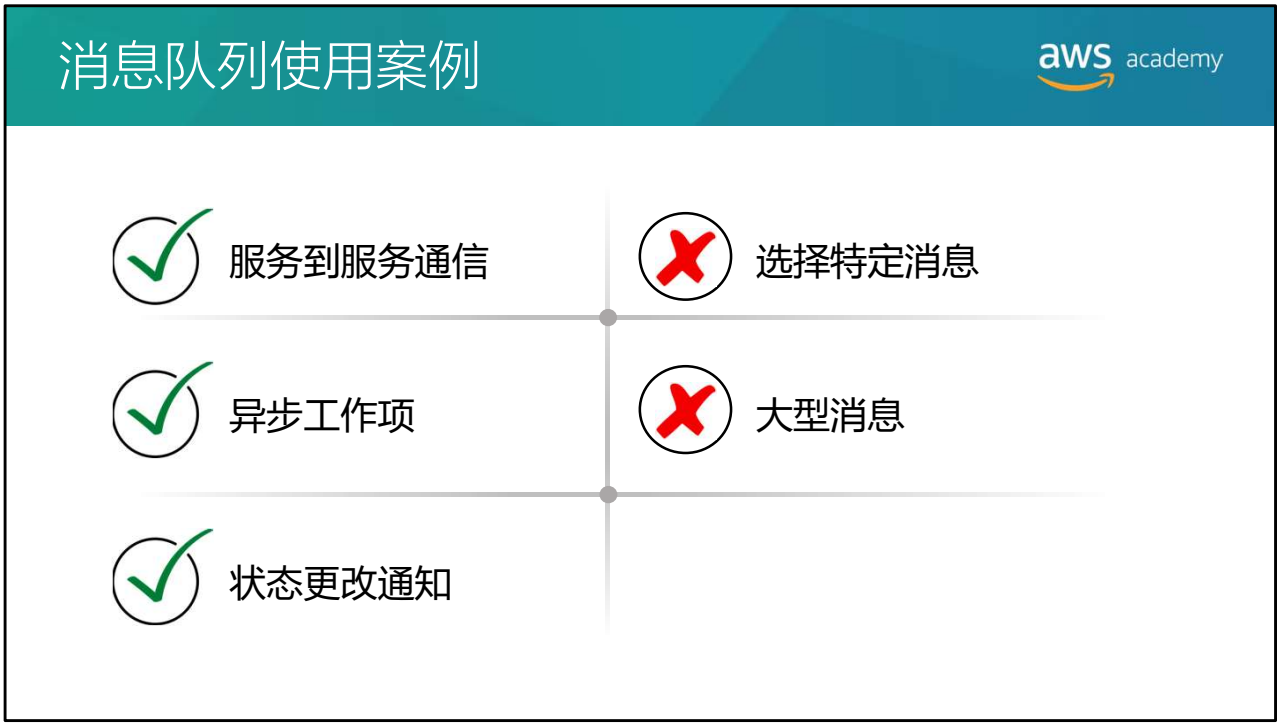
此外，您现在有了将订单作为消息持久存储的机制（队列充当临时数据库）。您还将数据库中的事务范围进一步向下移动到堆栈。如果发生应用程序异常或事务失败，此设计有助于确保将订单处理重试或重定向到死信队列，以便日后进行重新处理。

有关此使用案例的更多信息，请参阅此 AWS 计算博客文章。

消息队列使用案例

这些常见使用案例演示了消息队列最适合的场景：

- *服务到服务通信* – 例如，假设前端网站必须在后端客户关系管理 (CRM) 服务中更新客户的传递地址。您可以让前端网站的代码向 SQS 队列发送消息，然后让后端 CRM 服务使用这些消息。
- *异步工作项目* – 例如，假设酒店预订系统必须取消预留，这是一个需要很长时间的过程。您可以向 SQS 队列发送消息，让该酒店预订系统使用这些消息并执行异步取消。
- *状态变更通知* – 例如，假设您有管理某些资源的服务。您希望其他服务能够接收有关该资源更改的更新。库存系统可能会在某个物品库存量较低且需要订购时发布通知。



此外，了解特定技术何时*不*适合您的使用案例，也很重要。消息收发有其自己的一组常见反模式。

- *选择特定消息* – 您可能希望有选择地接收来自与特定属性集或与临时逻辑查询匹配的队列中的消息。例如，服务请求了一条具有特定属性的消息，因为它包含对服务发出的另一消息的响应。在这种情况下，队列中可能存在没有人轮询且没有人使用的消息。
- *大型消息* – 大多数消息收发协议和实施都最适用于合理大小的消息（数十或数百 KB）。随着消息大小的增加，您最好使用专用存储系统（例如 Amazon S3），并在消息本身中传递对该存储中对象的引用。

第 3 节要点

- Amazon SQS 是一项完全托管的消息队列服务，使您能够解耦应用程序组件，以便它们独立运行。
- Amazon SQS 支持标准队列和 FIFO 队列。
- 生产者向队列发送消息。在可见性超时期间，消费者可以处理和删除消息。
- 无法处理的消息可以发送到死信队列。
- 长轮询是从 SQS 队列中检索大量消息的方法。

本模块中这节内容的要点包括：

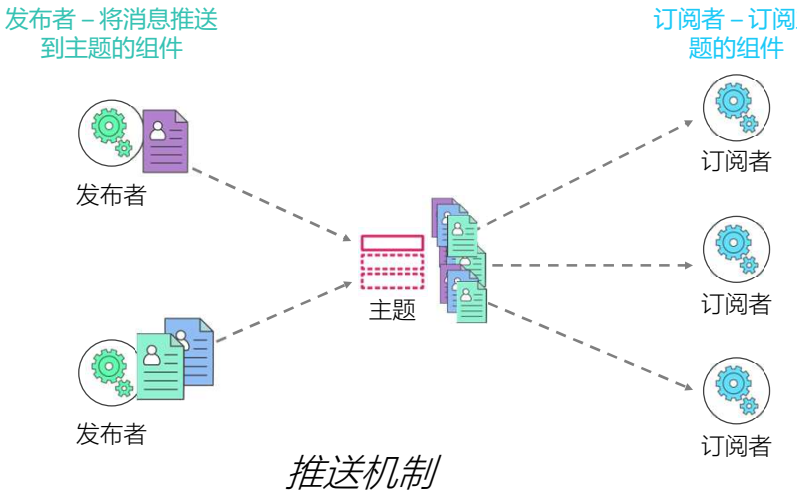
- Amazon SQS 是一项完全托管的消息队列服务，使您能够解耦应用程序组件，以便它们独立运行。
- Amazon SQS 支持标准队列和 FIFO 队列。
- 生产者向队列发送消息。在可见性超时期间，消费者可以处理和删除消息。
- 无法处理的消息可以发送到死信队列。
- 长轮询是从 SQS 队列中检索大量消息的方法。

模块 12：构建解耦架构

第 4 节：使用 Amazon SNS 进行解耦

介绍第 4 节：使用 Amazon SNS 进行解耦。

发布/订阅消息收发



在现代云架构中，应用程序被解耦为多个规模较小且更易于开发、部署和维护的独立构建块。发布/订阅 (pub/sub) 消息收发可以为这些分布式应用程序提供即时事件通知。

发布/订阅模式让消息能够异步广播到系统中的不同部分。消息*主题*提供了广播异步事件通知的轻量级机制。它还提供了软件组件能够连接到主题的终端节点，以便它们能够发送和接收这些消息。

在广播消息时，一个叫做 *“发布者”* 的组件会将消息推送到主题。与在消息被检索前批量处理消息的消息队列不同的是，消息主题无需或使用极少消息队列即可传输消息，并将消息立即推送给所有*订阅者*。除非设置了消息筛选策略，否则订阅者将收到广播的每条消息。订阅者的示例包括 Web 服务器、电子邮件地址、Amazon SQS 队列和 AWS Lambda 函数。

消息主题的订阅者通常执行不同的函数，并可以同时对消息执行不同的操作。发布者无需知道谁在使用广播的信息，而订阅者也无需知道消息来自哪里。这种消息收发模式与消息队列稍有不同，在消息队列中，发送消息的组件通常知道发送的目的地。

在“发布/订阅”(pub-sub) 消息收发范式中，使用*推送机制*将通知传输到客户端，无需定期检查或轮询新信息和更新。

Amazon SNS

- 是具有高可用性、持久性、安全性和完全托管的发布/订阅消息收发服务
- 使用推送机制
- 支持使用客户主密钥 (CMK) 加密主题

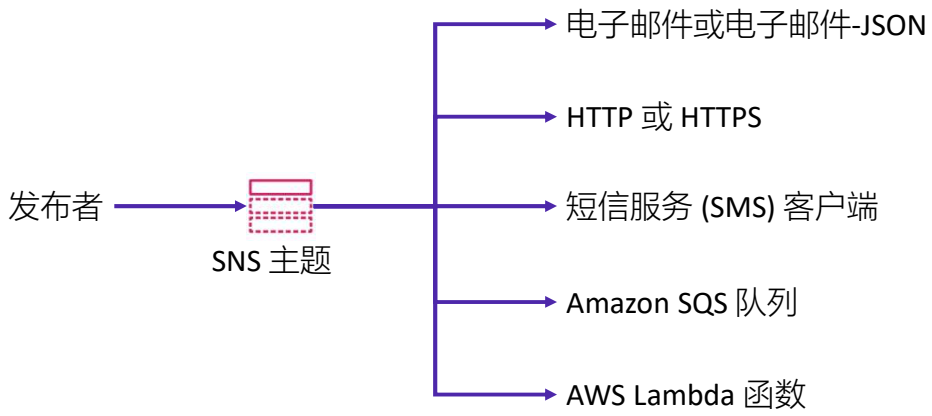
Amazon Simple Notification Service (Amazon SNS) 是一种 Web 服务，您可以轻松地在云中设置、操作和发送通知。该服务遵循“发布/订阅”(*pub-sub*) 消息收发范例，使用“推送”机制将通知传递给客户端。Amazon SNS 旨在满足最大型和要求最严格的应用程序的需求，使应用程序可以随时发布无限量的消息。

使用 Amazon SNS 时，您可以创建*主题*并设置策略来限制谁可以发布或订阅该主题。发布者会发送消息至他们创建的主题或他们有权发布的主题。然后，Amazon SNS 会将主题与该主题的订阅者列表进行匹配，并将消息传递给每个订阅者。每个主题都有一个唯一的名称，为发布者和订阅者标识 Amazon SNS 终端节点，以便他们发布消息和订阅注册通知。订阅者会收到发布至他们所订阅主题的所有消息，且一个主题的所有订阅者收到的消息都相同。

Amazon SNS 支持加密主题。当您将消息发布到加密主题后，Amazon SNS 会使用客户主密钥 (CMK) 来加密您的消息。CMK 是 AWS KMS 中的主要资源。Amazon SNS 支持客户托管的 CMK，也支持 AWS 管理的 CMK。

在 Amazon SNS 收到您的消息后，它们会使用 256 位高级加密标准伽罗瓦/计数器模式 (AES-GCM) 算法进行加密。加密后的消息会以冗余方式存储在多个服务器和数据中心之间，并跨多个可用区以实现持久性。消息在传送到已订阅的终端节点之前会进行解密。如需了解有关加密发布到 Amazon SNS 的消息的信息，请阅读此 AWS 计算博客文章。

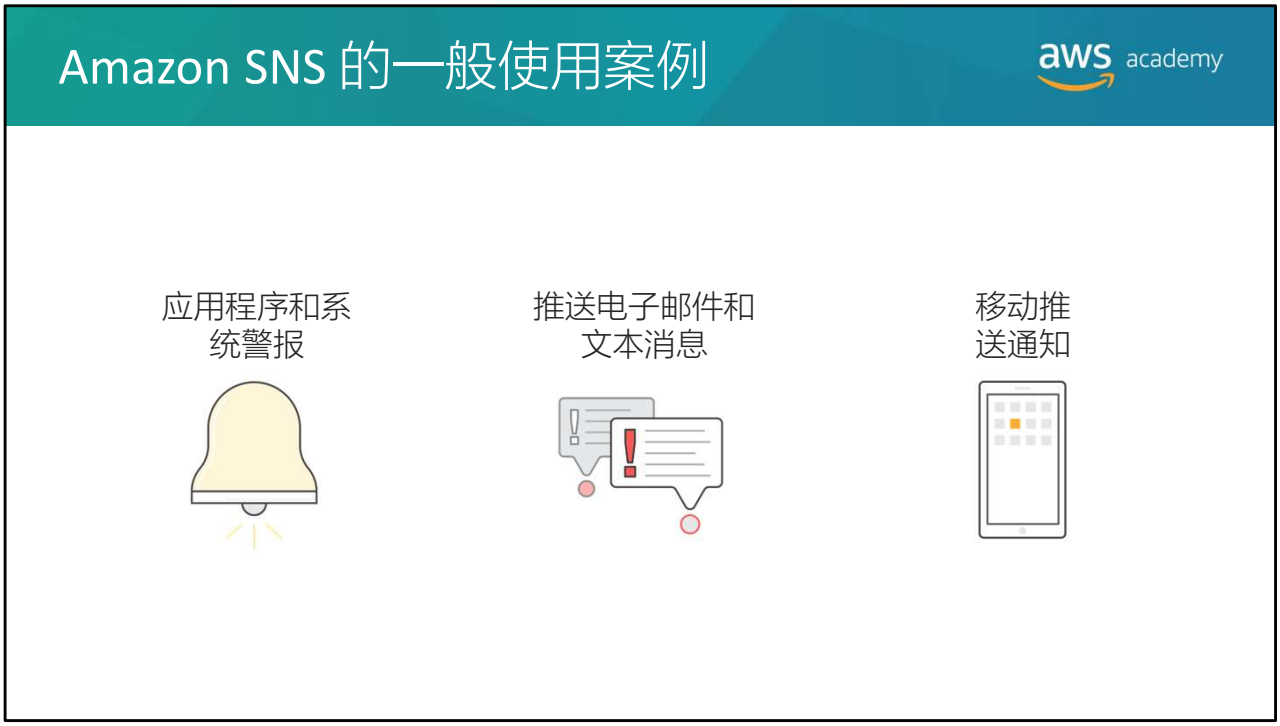
支持的传输协议



Amazon SNS 支持以下传输协议来传送消息：

- *电子邮件或电子邮件-JSON* – 消息以电子邮件的形式发送到注册地址。电子邮件 JSON 以 JavaScript 对象表示法 (JSON) 对象的形式发送通知，电子邮件则发送基于文本的电子邮件。
- *超文本传输协议 (HTTP) 或安全 HTTP (HTTPS)* – 在订阅注册期间，订阅者将指定 URL。消息通过 HTTP POST 请求传送到指定的 URL。
- *短信服务 (SMS)* – 消息以 SMS 文本消息的形式发送到注册的电话号码。
- *Amazon SQS 队列* – 用户将 SQS 标准队列指定为终端节点。Amazon SNS 会将通知消息排队到指定队列。FIFO 队列目前不受支持。
- *AWS Lambda 函数* – 消息传递到 AWS Lambda 函数，用于处理消息自定义项、维持消息持久性或与其他 AWS 服务进行通信。

Amazon SNS 的一般使用案例



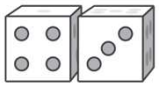
有多种方法可以使用 Amazon SNS：

- *应用程序和系统警报* – 您可以使用 Amazon SNS 在事件发生时接收即时通知，例如对 Auto Scaling 组进行更改。
- *推送电子邮件和文本消息* – 您可以使用 Amazon SNS 通过电子邮件或 SMS 将目标新闻标题推送给订阅者。
- *移动推送通知* – 您可以使用 Amazon SNS 向应用程序发送通知，表明有可用的更新。通知消息可以包含下载和安装更新的链接。

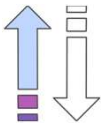
Amazon SNS 考虑因素

单个发布消息

没有召回选项



无法保证顺序和传递

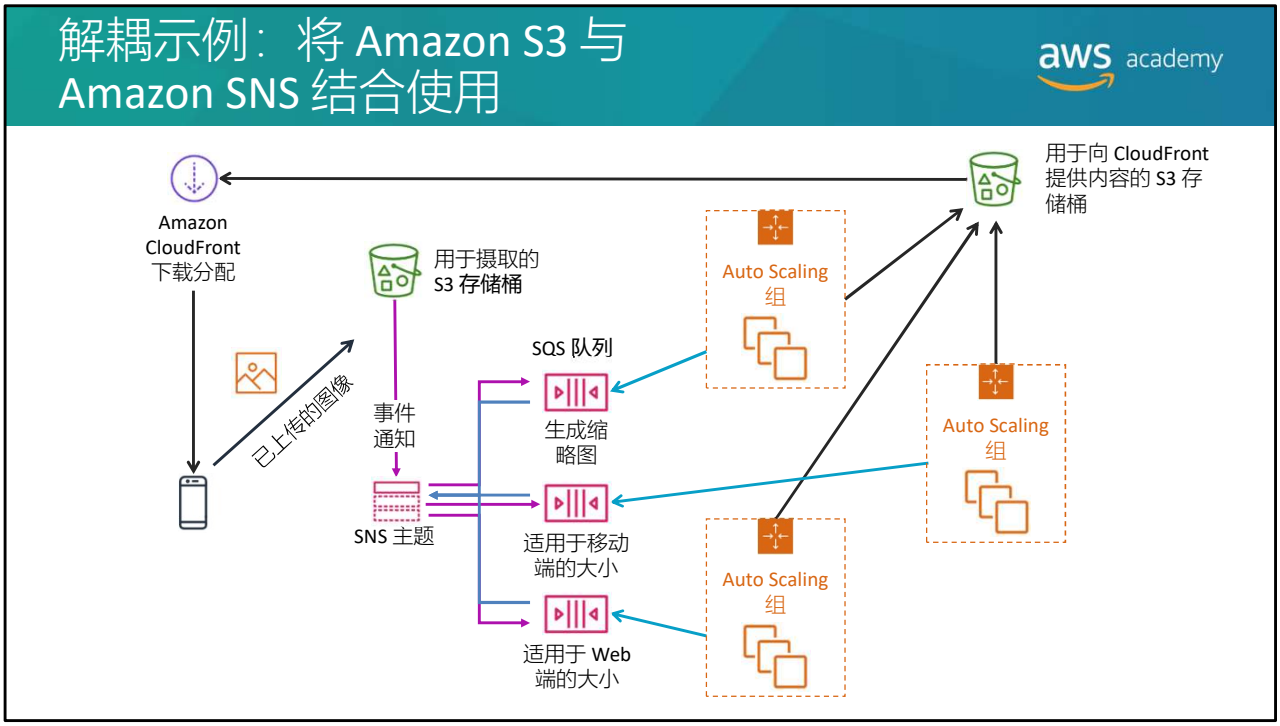


为每个传输协议重试策略

如果您计划使用 Amazon SNS，请考虑以下几点：

- 每个通知消息都包含一条已发布的消息。
- 消息成功传递后便无法再重新调用。
- Amazon SNS 会尝试按发布到主题中的顺序来传递发布者的消息。但是，网络问题可能会导致订阅者端的消息顺序错乱。
- Amazon SNS 为每个传输协议定义了一个传输策略。传输策略定义了在发生服务器端错误时（即当托管已订阅终端节点的系统变得不可用时），Amazon SNS 如何重试消息传输。如果第一次尝试无法成功传递消息，则 Amazon SNS 将使用四阶段重试策略：
    1. 在两次尝试之间没有延迟地重试；
    2. 在两次尝试之间进行延迟最小的重试；
    3. 根据退避模型进行重试；
    4. 在两次尝试之间进行延迟最大的重试；当消息传递重试策略用尽时，Amazon SNS 可以将消息移动至 DLQ。

解耦示例：将 Amazon S3 与 Amazon SNS 结合使用



通过 Amazon SNS，您可以使用主题将消息发布者与订阅者解耦，将消息同时群发给多个收件人，并消除应用程序中的轮询。

您可以使用 Amazon SNS 在单个账户中发送消息或向不同账户中的资源发送消息。

AWS 服务（例如 Amazon EC2、Amazon S3 和 Amazon CloudWatch）可以将消息发布到您的 SNS 主题，以触发事件驱动型计算和工作流。在此示例中，将映像上传到 S3 存储桶后，Amazon S3 会触发事件通知，此操作会自动将消息发送到 SNS 主题。然后，Amazon SNS 将 S3 事件通知传递给 SQS 队列订阅者。EC2 实例的 Auto Scaling 组处理 SQS 队列中的消息，然后将处理后的映像发布到 S3 存储桶，该存储桶将内容提供给 Amazon CloudFront。

Amazon SNS 与 Amazon SQS 的对比

特征	**Amazon SNS**（发布者/订阅者）	**Amazon SQS**（生产者/消费者）
（生产者/消费者）	发布/订阅	发送/接收
交付机制	推送（被动）	轮询（主动）
分配模型	多对多	一对一
消息持久性	否	是

- Amazon SNS 使用发布/订阅消息收发范式支持应用程序通过推送机制向多个订阅者发送对时间有严格要求的消息。
- Amazon SQS 使用发布/订阅消息收发范式并通过轮询模式交换消息 – 发送组件和接收组件是解耦的。
- Amazon SQS 为应用程序的分布式组件提供了灵活性 – 无需每个组件同时可用即可发送和接收消息。

第 4 节要点

- Amazon SNS 是一种 Web 服务，以便您设置、运行及从云中发送通知
- Amazon SNS 遵循发布/订阅消息收发模式
- 使用 Amazon SNS 时，您可以创建主题并设置策略来限制谁可以发布或订阅该主题
- 您可以使用主题将消息发布者与订阅者解耦，将消息同时群发给多个收件人，并消除应用程序中的轮询
- AWS 服务可以将消息发布到您的 SNS 主题，以触发事件驱动型计算和工作流

本模块中这节内容的要点包括：

- Amazon SNS 是一种 Web 服务，以便您设置、运行及从云中发送通知
- Amazon SNS 遵循发布/订阅消息收发模式
- 使用 Amazon SNS 时，您可以创建主题并设置策略来限制谁可以发布或订阅该主题
- 您可以使用主题将消息发布者与订阅者解耦，将消息同时群发给多个收件人，并消除应用程序中的轮询
- AWS 服务可以将消息发布到您的 SNS 主题，以触发事件驱动型计算和工作流

模块 12：构建解耦架构

第 5 节：使用 Amazon MQ 在云应用程序和本地之间发送消息

aws academy

介绍第 5 节：使用 Amazon MQ 在云应用程序和本地之间发送消息。

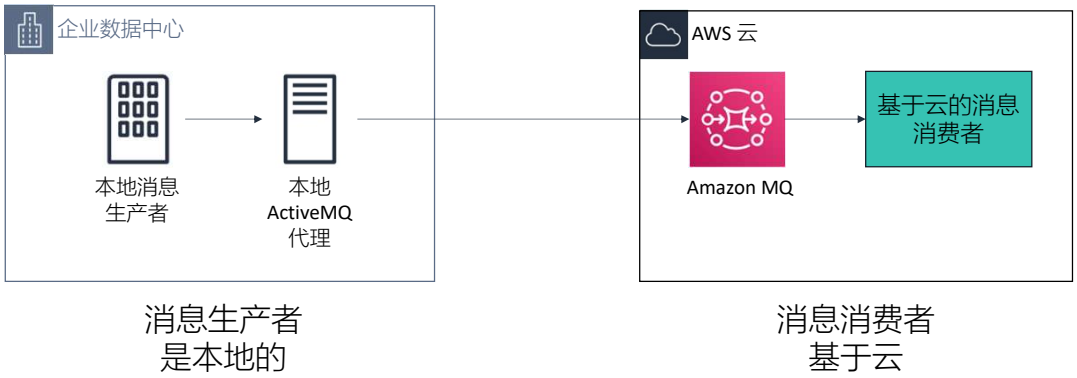
Amazon MQ

- 是 Apache ActiveMQ 的托管消息代理服务
- 管理 ActiveMQ 的预置、设置和维护
- 简化消息到云的迁移
- 与各种开放标准 API 和协议兼容
  - JMS、NMS、AMQP、STOMP、MQTT 和 WebSocket

Amazon MQ 是一种适用于 Apache ActiveMQ 的托管消息代理服务，让您能够在云中设置和操作消息代理。利用消息代理，不同的软件系统 （通常使用不同编程语言且在不同平台上运行）能够进行通信并交换信息。Amazon MQ 可以管理常用开源消息代理 ActiveMQ 的预置、设置和维护，从而减轻您的运营负荷。

通过 Amazon MQ，您可以轻松将消息收发迁移至云，同时保留应用程序间的现有连接。它支持用于消息收发的开放标准 API 和协议，包括 Java 消息服务 (JMS)、.NET 消息服务 (NMS)、高级消息队列协议 (AMQP)、面向流文本的消息传输协议 (STOMP)、消息队列遥测传输 (MQTT) 和 WebSocket。您可以轻松从使用这些标准的任意消息代理迁移至 Amazon MQ，通常无需重新编写任何消息收发代码。大多数情况下，您可以更新应用程序的终端节点来连接到 Amazon MQ，然后就可以开始发送消息。

Amazon MQ 使用案例：混合云环境



许多组织，尤其是企业，都依赖消息代理来连接和协调不同的系统。消息代理使分布式应用程序可以相互通信。他们是 IT 环境以及最终业务服务的技术支柱。应用程序依赖消息收发正常运行。

在许多情况下，这些组织已开始构建新的原生云应用程序或将应用程序迁移到 AWS。有些应用程序（例如大型机系统）的迁移成本太高。在这些情况下，本地应用程序仍必须与基于云的组件进行交互。

借助 Amazon MQ，组织能够在云中的应用程序与本地应用程序之间发送消息，以实现混合环境和应用程序现代化。例如，您可以从由 Amazon MQ 代理托管的队列和主题中调用 AWS Lambda，以将传统系统与无服务器架构进行集成。

该示例说明了您可以借助 ActiveMQ 的代理功能网络使用 Amazon MQ 将本地和云环境集成在一起。该图显示了从本地生产者到本地代理的消息生命周期，该消息生命周期遍历本地代理与 Amazon MQ 之间的混合连接。最后，消息迁移到 AWS 云内使用。

有关如何使用 Amazon MQ 集成本地和云环境的更多信息，请阅读此 AWS 计算博客文章。

Amazon MQ 与 Amazon SQS 和 Amazon SNS 的对比

Amazon MQ	Amazon SQS 和 SNS
对于应用程序迁移	对于诞生于云中的应用程序
协议：JMS、NMS、AMQP、STOMP、MQTT 和 WebSocket	协议：HTTPS
丰富功能	接近无限吞吐量
按小时付费和按 GB 付费	按请求付费
可以执行发布/订阅	无法在 Amazon SQS 中执行发布/订阅，但是您可以在 Amazon SNS 中执行发布/订阅

Amazon MQ 是一项托管消息代理服务，可兼容许多常见消息代理。Amazon SQS 和 Amazon SNS 都是高度可扩展、易于使用且不需要您设置消息代理的队列和主题服务。

- 如果您使用现有应用程序中的消息收发功能，并且想要将消息收发功能移至云中，AWS 建议您使用 Amazon MQ。它支持开放标准 API 和协议。您可以从基于标准的任何消息代理切换到 Amazon MQ，无需重新编写应用程序中的消息收发代码。
- 如果您要在云中构建新的应用程序，AWS 建议您使用 Amazon SQS 和 Amazon SNS。

第 5 节要点

- Amazon MQ 是一种适用于 Apache ActiveMQ 的托管消息代理服务，让您能够在云中设置和操作消息代理
- Amazon MQ 可以管理 ActiveMQ 的预置、设置和维护，这种是常用的开源消息代理
- Amazon MQ 与多种开放标准 API 和协议兼容（即 JMS、NMS、AMQP、STOMP、MQTT 和 WebSocket）
- 您可以借助 ActiveMQ 的代理功能网络使用 Amazon MQ 将本地和云环境集成在一起

本模块中这节内容的要点包括：

- Amazon MQ 是一种适用于 Apache ActiveMQ 的托管消息代理服务，让您能够在云中设置和操作消息代理
- Amazon MQ 可以管理 ActiveMQ 的预置、设置和维护，这种是常用的开源消息代理
- Amazon MQ 与多种开放标准 API 和协议兼容（即 JMS、NMS、AMQP、STOMP、MQTT 和 WebSocket）
- 您可以借助 ActiveMQ 的代理功能网络使用 Amazon MQ 将本地和云环境集成在一起

模块 12：构建解耦架构

模块总结

现在来回顾下本模块，并对知识测验和对实践认证考试问题的讨论进行总结。

模块总结

总体来说，您在本模块中学习了如何：

- 区分紧耦合架构和松耦合架构
- 确定 Amazon SQS 的工作原理以及使用时间
- 确定 Amazon SNS 的工作原理以及使用时间
- 描述 Amazon MQ

总体来说，您在本模块中学习了如何：

- 区分紧耦合架构和松耦合架构
- 确定 Amazon SQS 的工作原理以及使用时间
- 确定 Amazon SNS 的工作原理以及使用时间
- 描述 Amazon MQ

完成知识测验

现在可以完成本模块的知识测验。

样题

一家公司必须执行异步处理，并已将 Amazon Simple Queue Service (Amazon SQS) 作为解耦架构的一部分实施。该公司希望确保将轮询请求中的空响应数量保持在最低水平。

解决方案架构师应该做些什么来确保减少空响应数量呢？

A. 增加队列的最大消息保留期
B. 增加队列的重新驱动策略的最大接收量
C. 增加队列的默认可见性超时
D. 增加队列的长轮询等待时间

请查看答案选项，并根据之前突出显示的关键字排除错误选项。

正确答案是 D： *“增加队列的轮询等待时间。”* 当队列的 ReceiveMessageWaitTimeSeconds 属性设置为大于零的值时，长轮询将生效。长轮询允许 Amazon SQS 等待消息可用后再向 ReceiveMessage 请求发送响应，从而减少空响应的数量。

其他资源

- 使用 Amazon SQS 和 Amazon SNS 构建松耦合且可扩展的 C# 应用程序
- Amazon SQS 资源
- Amazon SNS 资源
- Amazon MQ 资源

如果您想了解有关本模块所涵盖主题的更多信息，下面这些其他资源可能会有所帮助：

- 使用 Amazon SQS 和 Amazon SNS 构建松耦合且可扩展的 C# 应用程序
- Amazon SQS 资源
- Amazon SNS 资源
- Amazon MQ 资源

感谢您完成本模块的学习。