亚马逊云科技

开发人员指南

Amazon Key Management Service

Amazon Key Management Service: 开发人员指南

Amazon Web Services 文档中描述的 Amazon Web Services 服务或功能可能因区域而异。要查看适用于中国区域的差异，请参阅 中国的 Amazon Web Services 服务入门 (PDF)。

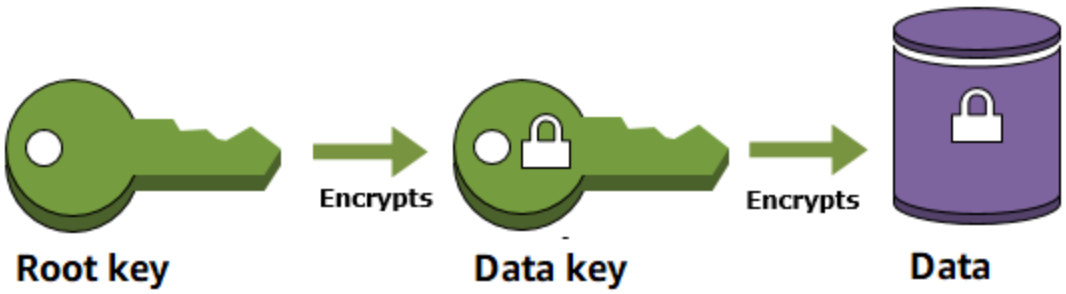
Table of Contents

Amazon Key Management Service

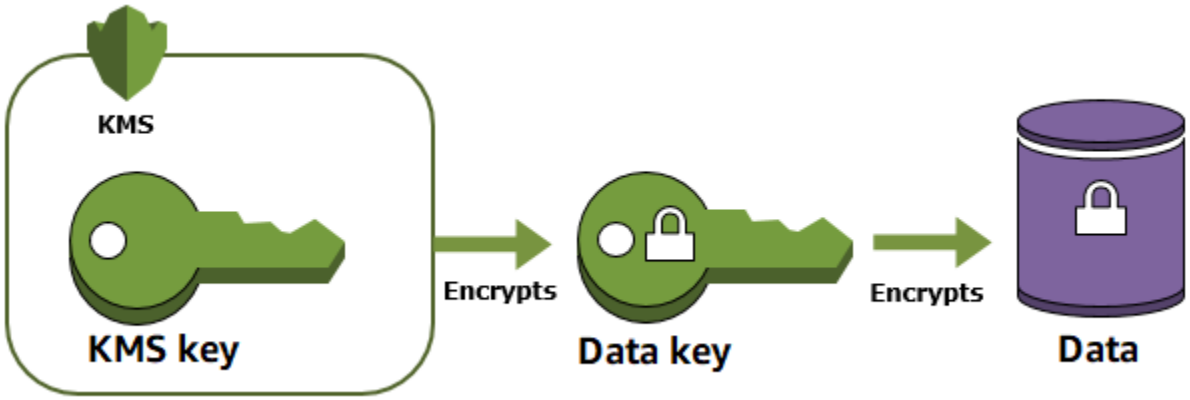
Amazon Key Management Service (Amazon KMS) 是一项 Amazon 托管服务，可让您轻松创建和控制用于加密和签名数据的密钥。您在中创建的 Amazon KMS 受 Amazon KMS keys 到 FIPS 140-3 安全等级 3 验证的硬件安全模块 (HSM) 的保护。他们永远不会以 Amazon KMS 未加密的方式离开。要使用或管理您的 KMS 密钥，您需要与进行交互 Amazon KMS。

为什么要使用 Amazon KMS？

在加密数据时，您需要保护加密密钥。如果加密您的密钥，您需要保护加密密钥。最终，您必须保护数据保护层次结构中最高级别的加密密钥（称为根密钥）。这就是用 Amazon KMS 武之地。



Amazon KMS 保护您的根密钥。KMS 密钥完全是在其中创建、管理、使用和删除的 Amazon KMS。其绝不会使服务处于未加密状态。要使用或管理您的 KMS 密钥，请致电 Amazon KMS。



此外，您还可以在中创建和管理密钥策略 Amazon KMS，确保只有受信任的用户才能访问 KMS 密钥。

Amazon KMS in Amazon Web Services 区域

Amazon Key Management Service 终端节点和配额中列出了支持的内容。 Amazon Web Services 区域 Amazon KMS 如果支持的 Amazon KMS 功能不 Amazon KMS 支持 Amazon Web Services 区域 ，则在有关该功能的主题中描述了区域差异。

Amazon KMS 定价

与其他 Amazon 产品一样，使用 Amazon KMS 不需要合同或最低购买量。有关 Amazon KMS 定价的更多信息，请参阅Amazon Key Management Service 定价。

Amazon KMS 服务等级协议

Amazon Key Management Service 以服务等级协议为后盾，该协议定义了我们的服务可用性政策。

正在访问 Amazon Key Management Service

您可以通过以下 Amazon KMS 方式使用：

主题

Amazon Web Services 管理控制台

控制台是一个基于 Web 的用户界面，用于 Amazon KMS 管理和 Amazon 资源。如果您已经注册了 Amazon Web Services 账户，则可以通过登录 Amazon Web Services 管理控制台 并 Amazon KMS 从 Amazon Web Services 管理控制台 主页上进行选择来访问 Amazon KMS 控制台。

使用 Amazon KMS 控制台所需的权限

要使用 Amazon KMS 控制台，用户必须拥有一组允许他们使用控制台中的 Amazon KMS 资源的最低权限 Amazon Web Services 账户。除这些 Amazon KMS 权限以外，用户还必须拥有列出 IAM 用户和 IAM 角色的权限。如果创建比必需的最低权限更为严格的 IAM policy，对于附加了该 IAM policy 的用户， Amazon KMS 控制台将无法按预期正常运行。

有关允许用户对 Amazon KMS 控制台进行只读访问所需的最低权限，请参阅允许用户在 Amazon KMS 控制台中查看 KMS 密钥。

要允许用户使用 Amazon KMS 控制台创建和管理 KMS 密钥，请将AWSKeyManagementServicePowerUser托管策略附加到用户，如中所述Amazon 的托管策略 Amazon Key Management Service。

您无需为通过Amazon SDKs、Amazon Command Line Interface或使用 Amazon KMS API 的用户提供最低控制台权限Amazon Tools for PowerShell。但是，您的确需要授予以下用户使用 API 的权限。有关更多信息，请参阅 权限参考。

Amazon Command Line Interface

您可以使用这些 Amazon CLI 工具在系统的命令行中发出命令或生成脚本来执行 Amazon （ 包括 Amazon KMS ）任务。

有关 Amazon KMS 通过使用的更多信息 Amazon CLI，请参阅《Amazon CLI 命令参考》

Amazon KMS REST API

的架构设计 Amazon KMS 为与编程语言无关。REST API是一个 HTTP 接口 Amazon KMS。使用 REST API 时，您可通过标准的 HTTP 请求来创建、获取和删除密钥。

有关使用的更多信息 Amazon KMS REST API，请参阅 Amazon Key Management Service API 参考

Amazon SDKs

Amazon 提供 SDKs （ 软件开发套件 ），其中包括常用编程语言和平台（ Java、、C JavaScript、Python 等 ）的库和示例代码。 Amazon SDKs 提供了一种便捷的方法来创建对 Amazon KMS 和的编程访问 Amazon。 Amazon KMS 是一项REST服务。您可以 Amazon KMS 使用 Amazon SDK 库向发送请求，这些库可以封装底层 Amazon KMS REST API并简化您的编程任务。有关信息（ Amazon SDKs包括如何下载和安装它们 ），请参阅构建工具 Amazon。

为 Amazon KMS 通过使用使用的代码示 Amazon KMS 例 Amazon SDKs提供了一个很好的起点 Amazon SDKs。

将此服务与 Amazon SDK 配合使用

Amazon 软件开发套件 (SDKs) 可用于许多流行的编程语言。每个软件开发工具包都提供 API、代码示例和文档，使开发人员能够更轻松地以其首选语言构建应用程序。

SDK 文档
Amazon CLI
适用于 Java 的 Amazon SDK

SDK 文档
适用于 JavaScript 的 Amazon SDK
适用于 .NET 的 Amazon SDK
适用于 PHP 的 Amazon SDK
Amazon Tools for PowerShell
适用于 Python (Boto3) 的 Amazon SDK
适用于 Ruby 的 Amazon SDK
适用于 SAP ABAP 的 Amazon SDK

Amazon Encryption SDK

Amazon Encryption SDK 是在应用程序中实现客户端加密的工具。它不提供对 KMS 的完全访问权限，而是与独立的 SDK 集成 Amazon KMS，或者可以在不引用 KMS 密钥的情况下用作独立 SDK。库可用于 Java、 JavaScript、C、Python 和其他编程语言。

有关更多信息，请参见Amazon Encryption SDK 开发人员指南。

Amazon KMS key 策略和 IAM 策略

Amazon KMS 最终一致性

由于系统的分布式特性， Amazon KMS API 遵循最终一致性模型。因此，您运行的后续命令可能无法立即看到对 Amazon KMS 资源的更改。

当您执行 Amazon KMS API 调用时，可能会有一段短暂的延迟，直到更改始终可用 Amazon KMS。更改通常需要不到几秒钟的时间即可在整个系统中传播，但在某些情况下，可能需要几分钟。在这段时间内，您可能会遇到意外错误，例如 `NotFoundException` 或 `InvalidStateException`。例如，`NotFoundException`如果您在呼叫后`GetParametersForImport`立即拨打电话，则 Amazon KMS 可能会返回`CreateKey`。

我们建议您在 Amazon KMS 客户端上配置重试策略，以便在短暂的等待时间后自动重试操作。有关更多信息，请参阅《工具参考指南》 Amazon SDKs 和《工具参考指南》中的重试行为。

对于与授权相关的 API 调用，您可以使用授权令牌来避免任何潜在的延迟，并立即在授权中使用权限。有关更多信息，请参阅最终一致性（用于授权）。

将混合后量子 TLS 与 Amazon KMS

Amazon Key Management Service (Amazon KMS) 支持传输层安全 (TLS) 网络加密协议的混合后量子密钥交换选项。当您连接到 Amazon KMS API 终端节点时，可以使用此 TLS 选项。这些混合后量子密钥交换功能是可选的，至少与我们目前使用的 TLS 加密一样安全，并且有可能会提供额外的长期安全优势。不过，与目前使用的传统密钥交换协议相比，它们会影响延迟和吞吐量性能。

您发送到 Amazon Key Management Service (Amazon KMS) 的数据在传输过程中受到传输层安全 (TLS) 连接提供的加密保护。借助 Amazon KMS 支持对 TLS 会话使用的传统密码套件，使得对密钥交换机制进行的暴力攻击在现有技术下是不可行的。不过，如果大规模量子计算在未来得到广泛应用，那么 TLS 密钥交换机制中使用的传统密码套件将会容易受到这些攻击。如果您开发的应用程序依赖于通过 TLS 连接传输的数据的长期机密性，则应考虑在大规模量子计算机可供使用之前迁移到后量子密码学的计划。 Amazon 正在努力为这个未来做准备，我们也希望你也做好充分的准备。

为了保护当今加密的数据免受未来潜在的攻击， Amazon 正在与密码学界一起开发抗量子算法或后量子算法。我们已经实现了混合后量子密钥交换密码套件 Amazon KMS ，该套件结合了经典和后量子元素，以确保您的TLS连接至少与经典密码套件一样强大。

在大多数情况下，这些混合密码套件可用于您的生产工作负载。 Amazon Web Services 区域但是，由于混合密码套件的性能特征和带宽要求与经典密钥交换机制不同，因此我们建议您在不同的条件下在 Amazon KMS API 调用中对其进行测试。

反馈

与以前一样，我们希望您能够提供反馈并参与我们的开源存储库。我们尤其希望了解您的基础设施如何与新形式的 TLS 流量进行交互。

- 要提供有关本主题的反馈，请使用此页面右上角的反馈链接。
- 我们正在的s2n-tls存储库中以开源方式开发这些混合密码套件。 GitHub要提供有关密码套件可用性方面的反馈，或者共享全新的测试条件或结果，请在 s2n-tls 存储库中创建问题。
- 我们正在编写代码示例，用于在aws-kms-pq-tls-example GitHub 存储库 Amazon KMS 中使用混合后量子 TLS。要提出问题或分享有关将 HTTP 客户端或 Amazon KMS 客户端配置为使用混合密码套件的想法，请在存储库中aws-kms-pq-tls-example创建问题。

支持 Amazon Web Services 区域

后量子 TLS Amazon KMS 适用于所有 Amazon Web Services 区域 Amazon KMS 支持的版本。

有关每个 Amazon KMS 终端节点的列表 Amazon Web Services 区域，请参阅中的Amazon Key Management Service 终端节点和配额Amazon Web Services 一般参考。有关 FIPS 端点的信息，请参阅《Amazon Web Services 一般参考》中的 FIPS endpoints。

关于 TLS 中的混合后量子密钥交换

Amazon KMS 支持混合后量子密钥交换密码套件。您可以在 Linux 系统上使用 Amazon SDK for Java 2.x 和 Amazon 通用运行时来配置使用这些密码套件的 HTTP 客户端。然后，每当您使用 HTTP 客户端连接到 Amazon KMS 终端节点时，都会使用混合密码套件。

此 HTTP 客户端使用 s2n-tls，后者是 TLS 协议的开源实现。s2n-tls 使用的混合密码套件只能用于密钥交换，而不能用于直接数据加密。在密钥交换过程中，客户端和服务器会计算将用于加密和解密传输中的数据的密钥。

s2n-tls 使用一种将椭圆曲线 Diffie-Hellman 算法（ECDH）与基于模格的密钥封装机制（ML-KEM）进行组合的混合算法。ECDH 算法是目前 TLS 中使用的传统密钥交换算法，基于模格的密钥封装机制是美国国家标准和技术研究院（NIST）指定为第一个标准后量子密钥协商算法的公有密钥加密和密钥生成算法。此混合算法单独使用每个算法来生成密钥。然后，以加密方式结合使用这两个密钥。使用 s2n-tls 时，您可以 配置 HTTP 客户端 来优先使用后量子 TLS，后者会将 ECDH 与 ML-KEM 放在优先列表的首位。传统密钥交换算法包含在优先列表中以确保兼容性，但它们的优先顺序较低。

将混合后量子 TLS 与 Amazon KMS

您可以使用混合后量子 TLS 来调用。 Amazon KMS在设置 HTTP 客户端测试环境时，请注意以下信息：

传输中加密

s2n-tls 中的混合密码套件仅用于传输中加密。当您的数据从您的客户端传输到 Amazon KMS 终端时，它们会保护您的数据。 Amazon KMS 不使用这些密码套件对下的数据进行加密。 Amazon KMS keys

取而代之的是，在 KMS 密钥下 Amazon KMS 加密您的数据时，它使用带有 256 位密钥的对称加密和伽罗瓦计数器模式下的高级加密标准 (AES-GCM) 算法，该算法已经具有量子抗性。理论上，在未来针对使用 256 位 AES-GCM 密钥创建的密文的大规模量子计算攻击中，密钥的有效安全性将会降至 128 位。这个安全级别足以使对 Amazon KMS 密文进行暴力攻击变得不可行。

支持的系统

目前，仅在 Linux 系统上支持使用 s2n-tls 中的混合密码套件。此外，只有支持 Amazon 公共运行时的密码套件 SDKs 才支持这些密码套件，例如。 Amazon SDK for Java 2.x有关示例，请参阅配置混合后量子 TLS。

Amazon KMS 端点

Amazon KMS 在所有端点上支持混合后量子 TLS，包括 FIPS 140-3 验证的端点。

配置混合后量子 TLS

在此过程中，为 Amazon 通用运行时 HTTP 客户端添加 Maven 依赖关系。然后配置一个优先使用后量子 TLS 的 HTTP 客户端。然后，创建一个使用 HTTP Amazon KMS 客户端的客户端。

要查看演示混合后量子 TLS 与 Amazon KMS结合使用的配置过程以及具体使用方法的完整工作示例，请参阅 aws-kms-pq-tls-example 存储库。

> **Note**
>
> C Amazon ommon Runtime HTTP 客户端已作为预览版提供，已于 2023 年 2 月正式上市。在正式发行版中，`tlsCipherPreference` 类和 `tlsCipherPreference()` 方法参数已替换为 `postQuantumTlsEnabled()` 方法参数替。如果您在预览期间使用此示例，则需要更新您的代码。

1. 将 Amazon 通用运行时客户端添加到你的 Maven 依赖项中。我们建议您使用最新可用版本。

   例如，此语句将 Amazon 通用运行时客户端`2.30.22`的版本添加到您的 Maven 依赖项中。

   ```
   <dependency>
       <groupId>software.amazon.awssdk</groupId>
       <artifactId>aws-crt-client</artifactId>
       <version>2.30.22</version>
   </dependency>
   ```

2. 要启用混合后量子密码套件，请将其添加到您的项目中并对其 Amazon SDK for Java 2.x 进行初始化。然后按照以下示例所示，在您的 HTTP 客户端上启用混合后量子密码套件。

   此代码使用 `postQuantumTlsEnabled()` 方法参数来配置 Amazon 通用运行时 HTTP 客户端，从而优先使用推荐的混合后量子密码套件（即 ECDH 与 ML-KEM 的组合）。然后，它使用配置的 HTTP 客户端来构建 Amazon KMS 异步客户端的实例`KmsAsyncClient`。此代码完成后，`KmsAsyncClient` 实例上的所有 Amazon KMS API 请求都将使用混合后量子 TLS。

```
// Configure HTTP client
SdkAsyncHttpClient awsCrtHttpClient = AwsCrtAsyncHttpClient.builder()
          .postQuantumTlsEnabled(true)
          .build();

// Create the Amazon KMS async client
KmsAsyncClient kmsAsync = KmsAsyncClient.builder()
         .httpClient(awsCrtHttpClient)
         .build();
```

3. 使用混合后量子 TLS 测试您的 Amazon KMS 通话。

   当您在配置的 Amazon KMS 客户端上调用 Amazon KMS API 操作时，您的呼叫将使用混合后量子 TLS 传输到 Amazon KMS 端点。要测试您的配置，请调用 Amazon KMS API，例如`ListKeys`。

   ```
   ListKeysReponse keys = kmsAsync.listKeys().get();
   ```

测试您的混合后量子 TLS 配置

考虑在调用的应用程序上使用混合密码套件运行以下测试。 Amazon KMS

- 查看 CloudTrail 日志条目中有关您的应用程序进行 Amazon KMS 的 API 调用的`tlsDetails`部分。该`keyExchange`字段应提及混合算法，例如`X25519MLKEM768`。有关示例，请参阅通过后量子 TLS 连接使用标准对称加密密钥进行解密。
- 使用混合后量子 TLS 运行基准测试。混合密钥交换会增加 TLS 握手中某些消息的大小和处理时间，但在大多数情况下，整体性能影响应该是难以察觉的。
- 请尝试从不同位置进行连接。根据您的请求所采用的网络路径，您可能会发现具有深度数据包检测 (DPI) 功能的传统中间主机、代理或防火墙会阻止该请求。这可能是由于在 TLS 握手ClientHello部分使用了新的密钥交换组，或者是较大的密钥交换消息所致。如果您在解决这些问题时遇到问题，请与您的安全团队或 IT 管理员合作更新相关配置并解除对新的 TLS 密钥交换组的封锁。

要了解有关后量子 TLS 的更多信息，请参阅 Amazon KMS

有关在中使用混合后量子 TLS 的更多信息 Amazon KMS，请参阅以下资源。

- 要了解后量子密码学 Amazon，包括博客文章和研究论文的链接，请参阅后量子密码学。

- 有关 s2n-tls 的信息，请参阅推出新的开源 TLS 实施 s2n-tls 和使用 s2n-tls。
- 有关 Amazon 通用运行时 HTTP 客户端的信息，请参阅Amazon SDK for Java 2.x 开发人员指南中的配置 Amazon 基于 CRT 的 HTTP 客户端。
- 有关美国国家标准和技术研究所 (NIST) 开展的后量子密码加密技术项目的信息，请参阅后量子密码加密技术。
- 有关 NIST 后量子密码标准化的信息，请参阅 后量子密码标准化。

Amazon KMS 通过 VPC 终端节点连接到

您可以 Amazon KMS 通过虚拟私有云 (VPC) 中的私有接口终端节点直接连接。当您使用接口 VPC 终端节点时，您 Amazon KMS 的 VPC 和之间的通信完全在 Amazon 网络内进行。

Amazon KMS 支持由提供支持的亚马逊虚拟私有云（亚马逊 VPC）终端节点Amazon PrivateLink。每个 VPC 终端节点都由一个或多个弹性网络接口 (ENIs) 表示，其私有 IP 地址位于您的 VPC 子网中。

接口 VPC 终端节点直接连接您的 VPC， Amazon KMS 无需互联网网关、NAT 设备、VPN Amazon Direct Connect 连接或连接。您的 VPC 中的实例不需要公有 IP 地址即可与之通信 Amazon KMS。

区域

Amazon KMS 全部 Amazon Web Services 区域 支持 VPC 终端节点和 VPC 终端节点策略。Amazon KMS

Amazon KMS VPC 终端节点的注意事项

在为设置接口 VPC 终端节点之前 Amazon KMS，请查看Amazon PrivateLink 指南中的接口终端节点属性和限制主题。

Amazon KMS 对 VPC 终端节点的支持包括以下内容。

- 您可以使用 VPC 端点从 VPC 调用所有 Amazon KMS API 操作。
- 您可以创建连接到 Amazon KMS 区域终端节点或 Amazon KMS FIPS 终端节点的接口 VPC 终端节点。
- 您可以使用 Amazon CloudTrail 日志来审核您通过 VPC 终端节点对 KMS 密钥的使用情况。有关更多信息，请参阅 记录使用 VPC 终端节点的 Amazon KMS 请求。

主题

- 为创建 VPC 终端节点 Amazon KMS
- 连接到 Amazon KMS VPC 终端节点

- 使用 VPC 终端节点控制对 Amazon KMS 资源的访问
- 记录使用 VPC 终端节点的 Amazon KMS 请求

为创建 VPC 终端节点 Amazon KMS

您可以使用亚马逊 VPC 控制台或亚马逊 VPC API 为 Amazon KMS 创建 VPC 终端节点。按照步骤使用以下值之一创建接口端点。

- 要为创建 VPC 终端节点 Amazon KMS，请使用以下服务名称：

  ```
  com.amazonaws.region.kms
  ```

  例如，在美国西部（俄勒冈）区域 (us-west-2)，服务名称为：

  ```
  com.amazonaws.us-west-2.kms
  ```

- 要创建连接到 Amazon KMS FIPS 端点的 VPC 端点，请使用以下服务名称：

  ```
  com.amazonaws.region.kms-fips
  ```

  例如，在美国西部（俄勒冈）区域 (us-west-2)，服务名称为：

  ```
  com.amazonaws.us-west-2.kms-fips
  ```

为了更轻松地使用 VPC 终端节点，您可以为 VPC 终端节点启用私有 DNS 名称。如果选择 Enable DNS Name（启用 DNS 名称）选项，标准 Amazon KMS DNS 主机名将解析为您的 VPC 端点。例如，`https://kms.us-west-2.amazonaws.com` 将解析为连接到服务名称 `com.amazonaws.us-west-2.kms` 的 VPC 端点。

此选项可让您更轻松地使用 VPC 终端节点。 Amazon SDKs 和默认 Amazon CLI 使用标准 Amazon KMS DNS 主机名，因此您无需在应用程序和命令中指定 VPC 终端节点 URL。

有关更多信息，请参阅 Amazon PrivateLink 指南中的通过接口端点访问服务。

连接到 Amazon KMS VPC 终端节点

您可以使用 Amazon SDK、或， Amazon KMS 通过 VPC 终端节点连接到 Amazon Tools for PowerShell。 Amazon CLI要指定 VPC 终端节点，请使用其 DNS 名称。

例如，此 list-keys 命令使用 `endpoint-url` 参数指定 VPC 终端节点。要使用类似命令，请将示例中的 VPC 终端节点 ID 替换为您账户中的 ID。

```
$ aws kms list-keys --endpoint-url https://vpce-1234abcdf5678c90a-09p7654s-us-east-1a.ec2.us-east-1.vpce.amazonaws.com
```

所需的权限

要使使用 VPC 终端节点的 Amazon KMS 请求成功，委托人需要来自两个来源的权限：

- 密钥策略、IAM policy 或者授权必须授予委托人对资源（KMS 密钥或别名）调用操作的权限。
- VPC 终端节点策略必须授予委托人使用终端节点发出请求的权限。

例如，密钥策略可能授予委托人对特定 KMS 密钥调用 Decrypt 的权限。但是，VPC 终端节点策略可能不允许该委托人通过使用终端节点对该 KMS 密钥调用 `Decrypt`。

或者，VPC 终端节点策略可能允许委托人使用终端节点调DisableKey用某些 KMS 密钥。但是，如果委托人没有来自密钥策略、IAM policy 或授权的权限，请求将失败。

您可以在创建终端节点时创建 VPC 终端节点策略，并且可以随时更改 VPC 终端节点策略。使用 VPC 管理控制台或CreateVpcEndpoint或ModifyVpcEndpoint操作。您也可以使用 Amazon CloudFormation 模板创建和更改 VPC 终端节点策略。有关使用 VPC 管理控制台的帮助，请参阅 Amazon PrivateLink 指南中的创建接口终端节点和修改接口终端节点。

私有主机名

如果在创建 VPC 终端节点时启用了私有主机名，则无需在 CLI 命令或应用程序配置中指定 VPC 终端节点 URL。标准 Amazon KMS DNS 主机名将解析为您的 VPC 端点。 Amazon CLI 和默认 SDKs 使用此主机名，因此您可以开始使用 VPC 终端节点连接到 Amazon KMS 区域终端节点，而无需更改脚本和应用程序中的任何内容。

要使用私有主机名，您的 VPC 的 `enableDnsHostnames` 和 `enableDnsSupport` 属性必须设置为 `true`。要设置这些属性，请使用ModifyVpcAttribute操作。有关详细信息，请参阅《Amazon VPC 用户指南》中的查看和更新 VPC 的 DNS 属性。

使用 VPC 终端节点控制对 Amazon KMS 资源的访问

当请求来自 VPC 或使用 VPC 终端节点时，您可以控制对 Amazon KMS 资源和操作的访问权限。为此，请在密钥策略或 IAM policy 中使用以下全局条件键之一。

- 使用 `aws:sourceVpce` 条件键基于 VPC 终端节点授予或限制访问。

- 使用 aws:sourceVpc 条件键基于托管私有终端节点的 VPC 授予或限制访问。

> **Note**
>
> 根据您的 VPC 终端节点创建密钥策略和 IAM policy 时要小心。如果政策声明要求请求来自特定 VPC 或 VPC 终端节点，则代表您使用 Amazon KMS 资源的集成 Amazon 服务发出的请求可能会失败。有关帮助信息，请参阅 在具有 Amazon KMS 权限的策略中使用 VPC 终端节点条件。
>
> 此外，当请求来自 Amazon VPC 终端节点时，aws:sourceIP 条件键也不起作用。要限制对 VPC 终端节点的请求，请使用 aws:sourceVpce 或 aws:sourceVpc 条件键。有关更多信息，请参阅《Amazon PrivateLink 指南》中的 VPC 终端节点和 VPC 终端节点服务的身份和访问管理。

您可以使用这些全局条件密钥来控制对 Amazon KMS keys （KMS 密钥）、别名的访问权限，以及对此类操作的访问权限 CreateKey，这些操作不依赖于任何特定资源。

例如，以下示例密钥策略允许用户仅在请求使用指定的 VPC 终端节点时，才使用 KMS 密钥执行某些加密操作。当用户向发出请求时 Amazon KMS，会将请求中的 VPC 终端节点 ID 与策略中的aws:sourceVpce条件键值进行比较。如果它们不匹配，则请求会被拒绝。

要使用类似的策略，请将占位符 Amazon Web Services 账户 ID 和 VPC 终端节点 IDs 替换为账户的有效值。

JSON

```
{
    "Id": "example-key-1",
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "EnableIAMpolicies",
            "Effect": "Allow",
            "Principal": {"AWS":["111122223333"]},
            "Action": ["kms:*"],
            "Resource": "*"
        },
        {
            "Sid": "Restrict usage to my VPC endpoint",
```

```
            "Effect": "Deny",
            "Principal": "*",
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*"
            ],
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                "aws:sourceVpce": "vpce-1234abcdf5678c90a"
                }
            }
        }

    ]
}
```

您还可以使用 aws:sourceVpc 条件键基于 VPC 终端节点所在的 VPC 限制对您的 KMS 密钥的访问。

以下示例密钥策略仅允许来自 vpc-12345678 的命令管理 KMS 密钥。另外，它只允许来自 vpc-2b2b2b2b 的命令使用 KMS 密钥执行加密操作。如果应用程序在一个 VPC 中运行，但您使用第二个隔离的 VPC 执行管理功能，则可以使用这样的策略。

要使用类似的策略，请将占位符 Amazon Web Services 账户 ID 和 VPC 终端节点 IDs 替换为账户的有效值。

JSON

```
{
    "Id": "example-key-2",
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAdministrativeActionsFromVPC",
            "Effect": "Allow",
            "Principal": {
                "AWS": "111122223333"
            },
```

```
            "Action": [
                "kms:Create*",
                "kms:Enable*",
                "kms:Put*",
                "kms:Update*",
                "kms:Revoke*",
                "kms:Disable*",
                "kms:Delete*",
                "kms:TagResource",
                "kms:UntagResource"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:sourceVpc": "vpc-12345678"
                }
            }
        },
        {
            "Sid": "AllowKeyUsageFromVPC2b2b2b2b",
            "Effect": "Allow",
            "Principal": {
                "AWS": "111122223333"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:GenerateDataKey*"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:sourceVpc": "vpc-2b2b2b2b"
                }
            }
        },
        {
            "Sid": "AllowReadActionsEverywhere",
            "Effect": "Allow",
            "Principal": {
                "AWS": "111122223333"
            },
            "Action": [
                "kms:Describe*",
```

```
                "kms:List*",
                "kms:Get*"
            ],
            "Resource": "*"
        }
    ]
}
```

记录使用 VPC 终端节点的 Amazon KMS 请求

Amazon CloudTrail 记录使用 VPC 终端节点的所有操作。当请求 Amazon KMS 使用 VPC 终端节点时，VPC 终端节点 ID 会出现在记录该请求的Amazon CloudTrail 日志条目中。您可以使用终端节点 ID 来审计 Amazon KMS VPC 终端节点的使用情况。

但是，您的 CloudTrail 日志不包括其他账户中的委托人请求的操作或其他账户中对 KMS 密钥和别名的 Amazon KMS 操作请求。此外，为了保护您的 VPC，被 VPC 端点策略拒绝但却以其他方式允许的请求不记录在 Amazon CloudTrail 中。

例如，此示例日志条目记录了使用 VPC 终端节点的 GenerateDataKey 请求。vpcEndpointId 字段出现在日志条目的末尾。

```
{
  "eventVersion":"1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "EX_PRINCIPAL_ID",
    "arn": "arn:aws:iam::111122223333:user/Alice",
    "accessKeyId": "EXAMPLE_KEY_ID",
    "accountId": "111122223333",
    "userName": "Alice"
  },
  "eventTime":"2018-01-16T05:46:57Z",
  "eventSource":"kms.amazonaws.com",
  "eventName":"GenerateDataKey",
  "awsRegion":"eu-west-1",
  "sourceIPAddress":"172.01.01.001",
  "userAgent":"aws-cli/1.14.23 Python/2.7.12 Linux/4.9.75-25.55.amzn1.x86_64
 botocore/1.8.27",
  "requestParameters":{
    "keyId":"1234abcd-12ab-34cd-56ef-1234567890ab",
    "numberOfBytes":128
```

```
  },
  "responseElements":null,
  "requestID":"a9fff0bf-fa80-11e7-a13c-afcabff2f04c",
  "eventID":"77274901-88bc-4e3f-9bb6-acf1c16f6a7c",
  "readOnly":true,
  "resources":[{
    "ARN":"arn:aws:kms:eu-
west-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "accountId":"111122223333",
    "type":"AWS::KMS::Key"
  }],
  "eventType":"AwsApiCall",
  "recipientAccountId":"111122223333",
  "vpcEndpointId": "vpce-1234abcdf5678c90a"
}
```

双堆栈端点支持

Amazon KMS 提供支持 IPv4 和 IPv6 客户端的双栈公共端点。双栈端点使客户端能够 Amazon KMS 使用 IPv4 或 IPv6地址与之通信。有关 Amazon KMS 终端节点的更多信息，请参阅终Amazon Key Management Service 端节点和配额。

的 Amazon KMS 双栈公共端点同时`https://kms.`*`your-region`*`.api.aws`支持 IPv4和 IPv6 客户端。 Amazon KMS 也可以使用您的虚拟私 IPv6 有云 (VPC) 进行私密访问 Amazon PrivateLink。 IPv4 有关为其创建私有接口 VPC 终端节点的更多信息 Amazon KMS，请参阅Amazon KMS 通过 VPC 终端节点连接到。

有关您的 IPv6 地址的更多信息 VPCs，请参阅《亚马逊虚拟私有云用户指南》中的 Amazon VPC 的工作原理。有关如何将您的 VPC 配置为双堆栈模式的更多信息，请参阅 Amazon Virtual Private Cloud 用户指南中的您的 VPCs 和子网的 IP 地址。

功能已过时不可用 IPv6

Amazon KMS 无法 IPv6 与 Amazon CloudHSM 密钥库或外部密钥库进行通信。此限制并不妨碍您拨 Amazon KMS APIs 打电话 IPv6。

Amazon KMS 概念

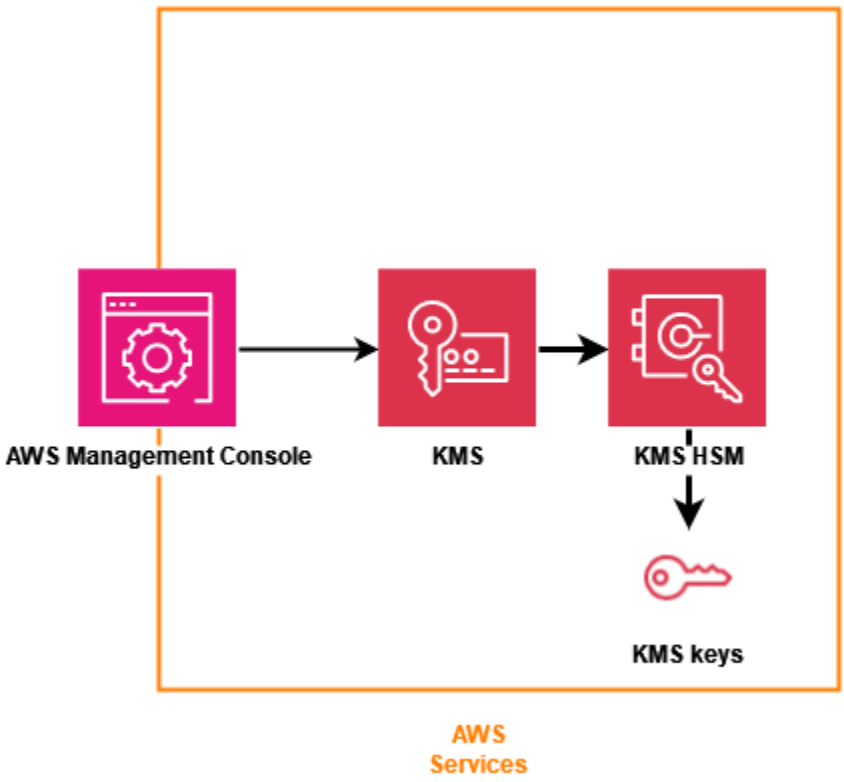
了解 Amazon Key Management Service (Amazon KMS) 中使用的基本术语和概念，以及它们如何协同工作以帮助保护您的数据。

简介 Amazon KMS

Amazon Key Management Service (Amazon KMS) 提供用于生成和管理加密密钥的 Web 界面，并充当保护数据的加密服务提供商。 Amazon KMS 提供与服务集成的传统密钥管理 Amazon 服务，通过集中管理和审计 Amazon，提供一致的客户密钥视图。

Amazon KMS 包括通过 Amazon Web Services 管理控制台、命令行界面和 RESTful API 操作的 Web 界面，用于请求由经 FIPS 140-3 验证的硬件安全模块组成的分布式队列的加密操作（ ）。HSMs Amazon KMS HSM 是一款多芯片独立硬件加密设备，旨在提供专用的加密功能，以满足安全性和可扩展性要求。 Amazon KMS您可以在作为 Amazon KMS keys管理的密钥下建立自己的基于 HSM 的加密层次结构。这些密钥仅在内存中可用， HSMs 并且仅在处理您的加密请求所需的必要时间内在内存中可用。您可以创建多个 KMS 密钥，每个密钥均由其密钥 ID 表示。只有在每个客户管理 Amazon 的 IAM 角色和账户下，才能创建、删除客户托管的 KMS 密钥，或者用于加密、解密、签名或验证数据。您可以通过创建附加到密钥的策略来定义对谁可以管理 KMS 密钥的 and/or 使用进行访问控制。此类策略允许您为每个 API 操作定义特定于应用程序的用途。

此外，大多数 Amazon 服务都支持使用 KMS 密钥对静态数据进行加密。此功能允许客户通过控制访问 KMS 密钥的方式和时间来控制 Amazon 服务访问加密数据的方式和时间。



Amazon KMS 是一项分层服务，由面向 Web Amazon KMS 的主机和一层组成。 HSMs这些分层主机的分组构成了 Amazon KMS 堆栈。对的所有请求都 Amazon KMS 必须通过传输层安全协议 (TLS) 发出，并在 Amazon KMS 主机上终止。 Amazon KMS 主机仅允许使用提供完美前向保密性的密码套件进行 TLS。 Amazon KMS 使用适用于所有其他 Amazon API 操作的相同凭证和策略机制 Amazon Identity and Access Management (IAM) 对您的请求进行身份验证和授权。

Amazon KMS 设计目标

Amazon KMS 旨在满足以下要求。

持久性

加密密钥的耐久性旨在与中 Amazon最高耐久性服务的耐久性相当。一个加密密钥可以加密长时间累积的大量数据。

值得信赖

密钥的使用受您定义和管理的访问控制策略的保护。没有导出明文 KMS 密钥的机制。加密密钥的机密性至关重要。多名具有特定角色访问权限的基于法定人数的访问控制的 Amazon 员工需要对这些权限执行管理操作。 HSMs

低延迟和高吞吐量

Amazon KMS 提供延迟和吞吐量级别的加密操作，适合中 Amazon其他服务使用。

独立区域

Amazon 为需要限制不同区域数据访问的客户提供独立区域。可以在 Amazon Web Services 区域内隔离密钥使用。

随机数的安全来源

由于强大的密码学依赖于真正不可预测的随机数生成，因此 Amazon KMS 提供了高质量且经过验证的随机数来源。

审核

Amazon KMS 在 Amazon CloudTrail 日志中记录加密密钥的使用和管理。您可以使用 Amazon CloudTrail 日志来检查您的加密密钥的使用情况，包括 Amazon 服务代表您使用密钥的情况。

为了实现这些目标，该 Amazon KMS 系统包括一组管理 “域” 的 Amazon KMS 运营商和服务主机运营商（统称为 “运营商”）。域是一组按区域定义的 Amazon KMS 服务器 HSMs、和运营商。每个 Amazon KMS 操作员都有一个硬件令牌，其中包含用于验证其操作的私钥和公钥对。它们 HSMs 还有一个额外的私钥和公钥对，用于建立保护 HSM 状态同步的加密密钥。

Amazon KMS keys

您为在自己的加密应用程序中使用而创建和管理的 KMS 密钥属于客户托管密钥类型。客户托管密钥也可以与使用 KMS 密钥对 Amazon 服务代表您存储的数据进行加密的服务结合使用。对于想要完全控制密钥生命周期和使用情况的客户，建议使用客户托管密钥。账户中拥有客户托管密钥将按月收费。此外，使用 and/or 管理密钥的请求会产生使用成本。有关更多详细信息，请参阅 Amazon Key Management Service 定价。

在某些情况下，客户可能想要一项 Amazon 服务来加密其数据，但他们不想承担管理密钥的开销，也不想为密钥付费。Amazon 托管式密钥是您账户中存在的 KMS 密钥，但只能在某些情况下使用。具体而言，它只能在您运营的 Amazon 服务环境中使用，并且只能由密钥存在的账户中的委托人使用。您无法对这些密钥的生命周期或权限进行任何管理。当您在 Amazon 服务中使用加密功能时，您可能会看到 Amazon 托管式密钥；它们使用的是 “aws<service code>” 形式的别名。例如，`aws/ebs` 密钥只能用于加密 EBS 卷，并且只能用于与密钥位于同一账户中的 IAM 主体使用的卷。想想 Amazon 托管式密钥 一个范围缩小到只有你账户中的用户才能使用你账户中的资源的。您不能 Amazon 托管式密钥与其他账户共享在和下加密的资源。虽然您的账户中可以免费存在，但分配给该密钥的 Amazon 服务会向您收取任何使用此类密钥的费用。 Amazon 托管式密钥

Amazon 托管式密钥 是一种传统密钥类型，自 2021 年起不再为新 Amazon 服务创建。取而代之的是，新（和旧）Amazon 服务默认使用所谓的加密客户数据。Amazon 拥有的密钥 Amazon 拥有的密钥 是 KMS 密钥，位于 Amazon 服务管理的账户中，因此服务运营商可以管理其生命周期和使用权限。通过使用 Amazon 拥有的密钥， Amazon 服务可以透明地加密您的数据，并允许轻松地跨账户或跨区域共享数据，而无需担心密钥权限。 Amazon 拥有的密钥 用于提供更轻松、自动化程度更高的数据保护 encryption-by-default的工作负载。由于这些密钥由所有和管理 Amazon，因此您无需为其存在或使用付费，也无法更改其策略，无法审计这些密钥上的活动，也无法删除它们。在重视控制时使用客户托管密钥，在最重视便利性时使用 Amazon 拥有的密钥 。

	客户托管密钥	Amazon 托管式密钥	Amazon 拥有的密钥
密钥策略	完全由客户控制	由服务控制；客户可以查看	独家控制，只能由加密 Amazon 数据的服务查看
日志记录	CloudTrail 客户跟踪或事件数据存储	CloudTrail 客户跟踪或事件数据存储	客户无法查看
生命周期管理	客户管理轮换、删除和区域位置	Amazon KMS 管理轮换（每年）、删除和区域位置	Amazon Web Services 服务 管理轮换、删除和区域位置
定价	针对密钥存在收取的月度费用（按小时按比例计收）。此外还需收取密钥使用费	不收取月度费用；但调用者需要支付使用这些密钥的 API 费用	不向客户收费

您创建的 KMS 密钥是客户托管式密钥。使用 KMS 密钥加密服务资源的 Amazon Web Services 服务 通常会为您创建密钥。在您的 Amazon 账户中 Amazon Web Services 服务 创建的 KMS 密钥是Amazon 托管式密钥。在服务账户中 Amazon Web Services 服务 创建的 KMS 密钥是Amazon 拥有的密钥。

KMS 密钥的类型	可以查看 KMS 密钥元数据	可以管理 KMS 密钥	仅用于我的 Amazon Web Services 账户	自动轮换	定价
客户托管的密钥	支持	是	是	可选。	月度费用（按小时计费） 每次使用费用
Amazon 托管式密钥	是	否	是	必需。每年（大约 365 天）。	没有月度费用 按次使用费（有些人会为您 Amazon Web Services 服务 支付此费用）
Amazon 拥有的密钥	否	否	否	Amazon Web Services 服务管理轮换策略	无费用

Amazon 与之集成的服务对 KMS 密钥的支持各不 Amazon KMS相同。默认情况下，某些 Amazon 服务会使用 Amazon 拥有的密钥 或来加密您的数据 Amazon 托管式密钥。某些 Amazon 服务支持客户管理的密钥。其他 Amazon 服务支持所有类型的 KMS 密钥 Amazon 拥有的密钥，使您可以轻松获取 Amazon 托管式密钥、查看或控制客户托管的密钥。有关 Amazon 服务提供的加密选项的详细信息，请参阅该服务的用户指南或开发人员指南中的静态加密主题。

客户托管密钥

您创建的 KMS 密钥是客户托管式密钥。客户托管密钥是您 Amazon Web Services 账户 自己创建、拥有和管理的 KMS 密钥。您可以完全控制这些 KMS 密钥，包括建立和维护其密钥策略、IAM policy 和授权、启用和禁用它们、轮换其加密材料、添加标签、创建别名（引用了 KMS 密钥）以及计划删除 KMS 密钥。

客户托管密钥显示在 Amazon KMS的 Amazon Web Services 管理控制台 的 Customer managed keys（客户托管密钥）页面上。要明确地标识客户托管密钥，请使用 DescribeKey 操作。对于客户托管密钥，`DescribeKey` 响应的 `KeyManager` 字段的值为 `CUSTOMER`。

您可以在加密操作中使用客户托管密钥并在 Amazon CloudTrail 日志中审核其使用情况。此外，许多与 Amazon KMS集成的Amazon 服务使您能够指定客户托管密钥以保护为您存储和管理的数据。

客户托管密钥会产生月费以及超过免费套餐使用量的费用。它们将计入您账户的 Amazon KMS 配额。有关详细信息，请参阅 Amazon Key Management Service 定价和限额。

Amazon 托管式密钥

Amazon 托管式密钥是您账户中的 KMS 密钥，由与集成的Amazon 服务代表您创建、管理和使用 Amazon KMS。

有些 Amazon 服务允许您选择一个 Amazon 托管式密钥 或一个客户托管的密钥来保护您在该服务中的资源。通常，除非要求您控制保护资源的加密密钥，否则 Amazon 托管式密钥 不妨选择。您不必创建或维护密钥或密钥策略，并且永远不会产生 Amazon 托管式密钥月度费用。

您有权 Amazon 托管式密钥在您的账户中查看、查看其密钥策略并审核其在 Amazon CloudTrail 日志中的使用情况。但是，您不能更改其任何属性 Amazon 托管式密钥、轮换它们、更改其密钥策略或安排将其删除。而且，您不能直接 Amazon 托管式密钥 在加密操作中使用；创建加密操作的服务会代表您使用它们。

组织中的资源控制策略不适用于 Amazon 托管式密钥。

Amazon 托管式密钥 显示在 fo Amazon 托管式密钥r Amazon Web Services 管理控制台 的页面上 Amazon KMS。您也可以 Amazon 托管式密钥 通过别名进行识别，其格式为 `aws/service-name`aws/redshift，例如。要明确识别 Amazon 托管式密钥，请使用DescribeKey操作。对于 Amazon 托管式密钥，`DescribeKey` 响应的 `KeyManager` 字段的值为 `Amazon`。

所有这些 Amazon 托管式密钥 都是每年自动轮换的。您不能更改此轮换计划。

> **Note**
>
> 2022年5月，将轮换时间表 Amazon 托管式密钥 从每三年（约1,095天） Amazon KMS 改为每年（约365天）。

不收取月费 Amazon 托管式密钥. 超出免费套餐的使用可能会收取费用，但有些 Amazon 服务会为您支付这些费用。有关详细信息，请参阅服务的用户指南或开发人员指南中的静态加密主题。有关详细信息，请参阅 Amazon Key Management Service 定价。

Amazon 托管式密钥 不要计入您账户中每个区域的 KMS 密钥数量的资源配额。但是，当代表您账户中的委托人使用这些 KMS 密钥时，它们将计入请求配额。有关更多信息，请参阅 限额。

Amazon 拥有的密钥

Amazon 拥有的密钥是 Amazon 服务拥有和管理的 KMS 密钥的集合，用于多个密钥 Amazon Web Services 账户。尽管 Amazon 拥有的密钥 不在您的账户中 Amazon Web Services 账户，但 Amazon 服务可以使用 Amazon 拥有的密钥 来保护您账户中的资源。

某些 Amazon 服务允许您选择 Amazon 拥有的密钥 或客户管理的密钥。通常，除非要求您审核或控制保护资源的加密密钥， Amazon 拥有的密钥 否则不妨选择。 Amazon 拥有的密钥 完全免费（没有月费或使用费），它们不计入您账户的Amazon KMS 配额，而且易于使用。您不必创建或维护该密钥或其密钥策略。

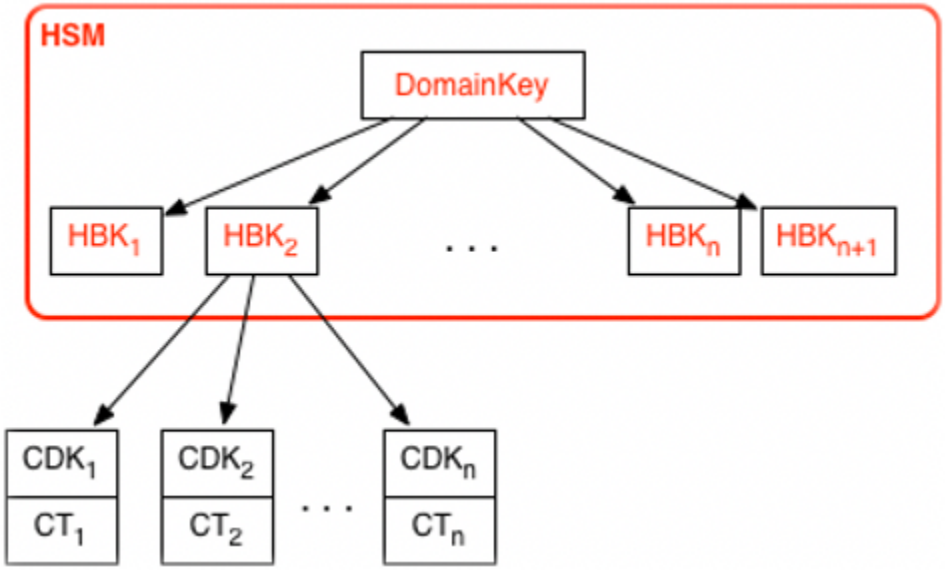
的轮换因服务 Amazon 拥有的密钥 而异。有关特定项轮换的信息 Amazon 拥有的密钥，请参阅该服务的用户指南或开发者指南中的静态加密主题。

Amazon KMS key 等级制度

您的密钥层次结构从顶级逻辑密钥开始，即 Amazon KMS key。KMS 密钥表示顶级密钥材料的容器，在 Amazon 服务命名空间中使用 Amazon Resource Name (ARN) 进行唯一定义。ARN 包含唯一生成的密钥标识符，即密钥 ID。KMS 密钥是根据用户通过 Amazon KMS发起的请求创建的。收到后，Amazon KMS 请求创建初始 HSM 支持密钥 (HBK)，将其放入 KMS 密钥容器中。HBK 在域中的 HSM 上生成，并且设计为永远不会以明文形式从 HSM 导出。相反地，HBK 在 HSM 管理的域密钥下以加密形式导出。 HBKs 这些导出的密钥令牌称为导出的密钥令牌 (EKTs)。

EKT 将导出到高持久性、低延迟的存储中。例如，假设您收到逻辑 KMS 密钥的 ARN。这表示您的密钥层次结构或加密上下文的顶部。您可以在自己的账户中创建多个 KMS 密钥，并像任何其他 Amazon 命名资源一样对 KMS 密钥设置策略。

在特定 KMS 密钥的层次结构中，可以将 HBK 视为 KMS 密钥的一个版本。当您想要轮换 KMS 密钥时 Amazon KMS，会创建一个新的 HBK，并将其与 KMS 密钥关联为 KMS 密钥的主动 HBK。较旧 HBKs 的数据会被保留，可用于解密和验证以前受保护的数据。但只有活动的加密密钥才能用于保护新信息。



您可以通过 Amazon KMS 请求使用您的 KMS 密钥直接保护信息，或者请求受您的 KMS 密钥保护的其他由 HSM 生成的密钥。这些密钥称为客户数据密钥，或 CDKs。 CDKs 可以加密后返回为密文 (CT)、纯文本或两者兼而有之。所有使用 KMS 密钥加密的对象（客户提供的数据或 HSM 生成的密钥）只能通过调用在 HSM 上进行解密。 Amazon KMS

返回的密文或解密后的有效载荷永远不会存储在其中。 Amazon KMS该信息通过与 Amazon KMS的 TLS 连接返回给您。这也适用于 Amazon 服务部门代表您拨打的电话。

密钥层次结构和特定密钥属性如下表中所示。

键	描述	生命周期
域密钥	仅在 HSM 内存中的 256 位 AES-GCM 密钥，用于包装 KMS 密钥（HSM 备用密钥）的版本。	每天轮换[1]
HSM 备用密钥	256 位对称密钥或者 RSA 或椭圆曲线私有密钥，用于保护客户数据和密钥，在域密钥下加密存储。。一个或多个 HSM 备用密钥组成 KMS 密钥（通过 keyId 表示）。	每年轮换[2]（可选配置）
派生加密密钥	仅在 HSM 内存中的 256 位 AES-GCM 密钥，用于加密客户数据和密钥。从每个加密的 HBK 派生。	每次加密时使用一次，并在解密时重新生成

键	描述	生命周期
客户数据密钥	以明文和密文形式从 HSM 导出的、用户定义的对称或非对称密钥。 在 HSM 备用密钥下加密，然后通过 TLS 通道返回给授权用户。	轮换和使用由应用程序控制

[1] Amazon KMS 可能会不时地将域密钥轮换放宽到最多每周一次，以处理域管理和配置任务。

[2] 由您代您 Amazon 托管式密钥 创建和管理 Amazon KMS 的默认值每年自动轮换。

密钥标识符 (KeyId)

密钥标识符用作 KMS 密钥的名称。它们可帮助您在控制台中识别 KMS 密钥。您可以使用它们来指示要在 Amazon KMS API 操作、密钥策略、IAM policy 和授权中使用的 KMS 密钥。密钥标识符值跟与 KMS 密钥关联的密钥材料完全无关。

Amazon KMS 定义了几个密钥标识符。创建 KMS 密钥时， Amazon KMS 会生成密钥 ARN 和密钥 ID，它们是 KMS 密钥的属性。创建别名时， Amazon KMS 会根据您定义的别名生成别名 ARN。您可以在和的 Amazon KMS API 中查看密钥 Amazon Web Services 管理控制台 和别名标识符。

在 Amazon KMS 控制台中，您可以按密钥 ARN、密钥 ID 或别名查看和筛选 KMS 密钥，并按密钥 ID 和别名进行排序。有关在控制台中查找密钥标识符的帮助，请参阅the section called “查找密钥 ID 和密钥 ARN”。

在 Amazon KMS API 中，用于标识 KMS 密钥的参数被命名`KeyId`或变体，例如`TargetKeyId`或`DestinationKeyId`。但是，这些参数的值并不仅限于 key IDs。一些参数可以使用任意有效的密钥标识符。有关每个参数的值的信息，请参阅《 Amazon Key Management Service API 参考》中的参数描述。

> **Note**
>
> 使用 Amazon KMS API 时，请注意您使用的密钥标识符。不同 APIs 需要不同的密钥标识符。通常，请在您的任务中使用最完整实用的密钥标识符。

Amazon KMS 支持以下密钥标识符。

密钥 ARN

密钥 ARN 是 KMS 密钥的 Amazon Resource Name (ARN)。它是 KMS 密钥唯一的完全限定标识符。密钥 ARN 包括 Amazon Web Services 账户、区域和密钥 ID。有关查找 KMS 密钥的密钥 ARN 的帮助，请参阅 the section called “查找密钥 ID 和密钥 ARN”。

密钥 ARN 的格式如下：

```
arn:<partition>:kms:<region>:<account-id>:key/<key-id>
```

以下是单区域 KMS 密钥的示例密钥 ARN。

```
arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

多区域密钥 ARNs 的密钥*key-id*元素以mrk-前缀开头。以下是多区域密钥的示例密钥 ARN。

```
arn:aws:kms:us-west-2:111122223333:key/mrk-1234abcd12ab34cd56ef1234567890ab
```

密钥 ID

密钥 ID 唯一地标识账户和区域中的 KMS 密钥。有关查找 KMS 密钥的密钥 ID 的帮助，请参阅 the section called “查找密钥 ID 和密钥 ARN”。

以下是单区域 KMS 密钥的示例密钥 ID。

```
1234abcd-12ab-34cd-56ef-1234567890ab
```

多区域密钥 IDs 的密钥以mrk-前缀开头。以下是多区域密钥的示例密钥 ID。

```
mrk-1234abcd12ab34cd56ef1234567890ab
```

别名 ARN

别名 ARN 是别名的亚马逊资源名称 (ARN)。 Amazon KMS 它是别名及所表示 KMS 密钥的唯一的完全限定标识符。别名 ARN 包括 Amazon Web Services 账户、地区和别名。

在任何给定时间，一个别名 ARN 标识一个特定的 KMS 密钥。但是，由于您可以更改与别名关联的 KMS 密钥，别名 ARN 在不同时间可以标识不同的 KMS 密钥。有关查找 KMS 密钥的别名 ARN 的帮助，请参阅 查找 KMS 密钥的别名和别名 ARN。

别名 ARN 的格式如下：

```
arn:<partition>:kms:<region>:<account-id>:alias/<alias-name>
```

以下是虚构的 ExampleAlias 的别名 ARN。

```
arn:aws:kms:us-west-2:111122223333:alias/ExampleAlias
```

别名

别名是最多 256 个字符的字符串。它唯一地标识某个账户和区域内关联的一个 KMS 密钥。在 Amazon KMS API 中，别名始终以开头alias/。有关查找 KMS 密钥的别名的帮助，请参阅 查找 KMS 密钥的别名和别名 ARN。

别名的格式如下：

```
alias/<alias-name>
```

例如：

```
alias/ExampleAlias
```

别名的 aws/ 前缀保留用于 Amazon 托管式密钥。您无法使用此前缀创建别名。例如，亚马逊简单存储服务 (Amazon S3) Simple Service 的别名如下所示。 Amazon 托管式密钥

```
alias/aws/s3
```

中的非对称密钥 Amazon KMS

非对称 KMS 密钥表示数学上相关的公有密钥和私有密钥对。公有密钥可以交给任何人，即使他们不可靠，但私有密钥必须保密。

在非对称 KMS 密钥中，私钥是在中创建的， Amazon KMS 并且永远不会处于 Amazon KMS 未加密状态。要使用私钥，您必须致电 Amazon KMS。您可以 Amazon KMS 通过调用 Amazon KMS API 操作来使用其中的公钥。或者，您可以下载公钥并在外部使用 Amazon KMS。

如果您的用例要求无法呼叫的用户在外部 Amazon 进行加密 Amazon KMS，那么非对称 KMS 密钥是一个不错的选择。但是，如果您要创建 KMS 密钥来加密您在 Amazon 服务中存储或管理的数据，请使

用对称加密 KMS 密钥。 Amazon 与之集成的服务仅 Amazon KMS使用对称加密 KMS 密钥来加密您的数据。这些服务不支持使用非对称 KMS 密钥进行加密。

使用对大于 4 KB 的邮件进行签名时 Amazon KMS，必须在签名 Amazon KMS 之前在外部对消息进行哈希处理。 Amazon KMS 提供了三个用于处理消息输入的`MessageType`选项：`RAW`用于纯文本消息（在其中 Amazon KMS 执行哈希）、`DIGEST`用于预哈希处理的消息（其中 Amazon KMS 跳过哈希步骤），以及`EXTERNAL_MU`专门用于输入为 64 字节代表 μ 值的 ML-DSA KMS 密钥规范。对于超过 4 KB 限制的大型消息，请在外部对消息进行哈希处理，并在调用 S Amazon KMSign and Verify 操作时使用 `MessageType:DIGEST`（或用`MessageType:EXTERNAL_MU`于 ML-DSA KMS 密钥）。Amazon KMS

Amazon KMS 支持多种类型的非对称 KMS 密钥。

RSA KMS 密钥

带有 RSA 密钥对的 KMS 密钥，用于加密和解密或签名和验证（但不能两者兼而有之）。 Amazon KMS 支持多种密钥长度，以满足不同的安全要求。

有关 Amazon KMS 支持 RSA KMS 密钥的加密和签名算法的技术细节，请参阅 RSA 密钥规范。

椭圆曲线（ECC）KMS 密钥

带有椭圆曲线密钥对的 KMS 密钥，用于签名和验证或派生共享密钥（但不能两者兼而有之）。Amazon KMS 支持几条常用的曲线。

有关 Amazon KMS 支持 ECC KMS 密钥的签名算法的技术细节，请参阅椭圆曲线密钥规范。

ML-DSA KMS 密钥

使用 ML-DSA 密钥对进行签名和验证的 KMS 密钥。ML-DSA 是由美国国家标准与技术研究院（NIST）开发的后量子密码术标准，旨在抵御量子计算带来的安全威胁。对于从 RSA 或椭圆曲线数字签名算法过渡到后量子安全密码术的组织，ML-DSA 是推荐使用的数字签名算法。

Amazon KMS 支持多种密钥长度，以满足不同的安全要求。有关 Amazon KMS 支持 ML-DSA KMS 密钥的签名算法的技术细节，请参阅 M L-DSA 密钥规范。

SM2 KMS 密钥（仅限中国区域）

带有密钥对的 KMS 密 SM2 钥，用于加密和解密、签名和验证或派生共享密钥（必须选择一种Key usage类型）。

有关 Amazon KMS 支持 SM2 KMS 密钥的加密和签名算法（仅限中国区域）的技术细节，SM2 请参阅密钥规范。

有关选择非对称密钥配置的帮助，请参阅 选择要创建的 KMS 密钥的类型。

区域

所有 Amazon Web Services 区域 支持的密钥都支持非对称 KMS 密钥和非对称数据密钥对。 Amazon KMS

了解更多

- 若要创建非对称 KMS 密钥，请参阅 创建非对称 KMS 密钥。
- 若要创建多区域非对称 KMS 密钥，请参阅 创建多区域主密钥。
- 要了解如何使用非对称 KMS 密钥签署消息和验证签名，请参阅 Amazon 安全博客中的采用 Amazon KMS的新的非对称密钥功能进行数字签名。
- 要了解删除非对称 KMS 密钥的特殊注意事项，请参阅Deleting asymmetric KMS keys。
- 要识别和查看非对称 KMS 密钥，请参阅识别非对称 KMS 密钥。

Amazon KMS 中的 HMAC 密钥

HMAC 散列消息认证码 KMS 密钥是用于生成和验证 Amazon KMS 内的 HMAC 的对称密钥。与每个 HMAC KMS 密钥关联的唯一密钥材料提供了 HMAC 算法所需的秘密密钥。您可以将 HMAC KMS 密钥与 `GenerateMac` 和 `VerifyMac` 操作结合使用以验证 Amazon KMS 中的数据的完整性和真实性。

HMAC 算法结合了加密哈希函数和共享密钥。他们获取消息和密钥，例如 HMAC KMS 密钥中的密钥材料，然后返回一个唯一的固定大小的代码或标签。即使是消息的一个字符发生了变化，或者密钥不完全相同，生成的标签也会完全不同。通过要求提供密钥，HMAC 还提供了真实性；如果没有密钥，就不可能生成相同的 HMAC 标签。HMAC 有时被称为对称签名，因为它们像数字签名一样工作，但使用单个密钥进行签名和验证。

Amazon KMS 使用的 HMAC KMS 密钥和 HMAC 算法符合 RFC 2104 中定义的行业标准。Amazon KMS GenerateMac 操作会生成标准的 HMAC 标签。这些密钥对在通过 FIPS 140-3 加密模块验证计划认证的 Amazon KMS 硬件安全模块中生成（中国（北京）和中国（宁夏）区域除外）并且绝不会使 Amazon KMS 处于未加密状态。要使用 HMAC KMS 密钥，您必须调用 Amazon KMS。

您可使用 HMAC KMS 密钥确定消息的真实性，例如 JSON Web 令牌（JWT）、令牌化的信用卡信息或提交的密码。它们也可以用作安全的密钥派生函数（KDF），尤其是在需要确定性密钥的应用程序中。

HMAC KMS 密钥比应用软件的 HMAC 具有优势，因为密钥材料完全在 Amazon KMS 内部生成和使用，受制于您对密钥设置的访问控制。

> **Tip**
>
> 最佳实践建议您限制包括 HMAC 在内的任何签名机制的有效时间。这阻止了行为者利用已签名信息反复地或是在消息被取代很久之后建立有效性的攻击。HMAC 标签不包含时间戳，但是您可以在令牌或消息中包含时间戳，以帮助您检测何时刷新 HMAC。

支持的加密操作

HMAC KMS 密钥只支持 `GenerateMac` 和 `VerifyMac` 加密操作。您不能使用 HMAC KMS 密钥加密数据或签名消息，也不能在 HMAC 操作中使用任何其他类型的 KMS 密钥。当您使用 `GenerateMac` 操作时，您可提供最多为 4096 个字节的消息、HMAC KMS 密钥以及与 HMAC 密钥规范兼容的 MAC 算法，`GenerateMac` 将计算 HMAC 标签。要验证 HMAC 标签，您必须提供 HMAC 标签以及相同的消息、HMAC KMS 密钥和 `GenerateMac` 用于计算原始 HMAC 标签的 MAC 算法。`VerifyMac` 操作计算 HMAC 标签并验证它与提供的 HMAC 标签是否相同。如果输入和计算的 HMAC 标签不相同，则验证失败。

HMAC KMS 密钥不支持自动密钥轮换，并且您无法在自定义密钥存储中创建 HMAC KMS 密钥。

如果您创建 KMS 密钥以加密 Amazon 服务中的数据，请使用对称加密密钥。您不能使用 HMAC KMS 密钥。

区域

Amazon KMS 支持的所有 Amazon Web Services 区域 均支持 HMAC KMS 密钥。

了解更多

- 要创建 HMAC KMS 密钥，请参阅 创建 HMAC KMS 密钥。
- 要创建多区域 HMAC KMS 密钥，请参阅 Amazon KMS 中的多区域密钥。
- 要检查 Amazon KMS 控制台为 HMAC KMS 密钥设置的默认密钥策略的差异，请参阅 the section called “允许密钥用户使用 KMS 密钥进行加密操作”。
- 要识别和查看 HMAC KMS 密钥，请参阅 识别 HMAC KMS 密钥。
- 要了解如何使用 HMAC 创建 JSON Web 令牌，请参阅 Amazon 安全博客中的如何在 Amazon KMS 内保护 HMAC。
- 收听播客：在官方 Amazon 播客上介绍 HMAC Amazon Key Management Service。

Amazon KMS 中的 ML-DSA 密钥

Amazon Key Management Service（Amazon KMS）支持用于后量子密码签名的模格数字签名算法（ML-DSA）。该实现遵循联邦信息处理标准（FIPS）204 标准，有助于抵御未来的量子计算威胁。Amazon KMS 在 FIPS 140-3 安全性 3 级认证硬件安全模块中创建和保护所有 ML-DSA 密钥和签名操作。为帮助恰当平衡安全性与性能需要，Amazon KMS 中的 ML-DSA 通过不同的密钥规范提供了三个不同的安全性等级，即 ML_DSA_44、ML_DSA_65 和 ML_DSA_87。

Amazon KMS 支持对使用 RAW 消息类型且大小不超过 4 KB 的消息进行非对称密钥签名。对于超过此大小的消息，必须按 NIST FIPS 204 第 6.2 节所定义，在外部计算 ML-DSA 签名中使用的 64 字节消息表示值 μ。在 Amazon KMS 签名操作中使用 EXTERNAL_MU 消息类型来指定这一预处理的 64 字节消息。使用同一消息和私有密钥时，由外部计算的 μ 所生成的签名与 RAW 生成的签名相同。请注意，此签名不同于 NIST FIPS 204 第 5.4 节中的“预哈希” ML-DSA 或 HashML-DSA。

有关使用 ML-DSA 和 EXTERNAL_MU 消息类型的更多信息，请参阅 ML-DSA 密钥规范。

有关使用 ML-DSA 和 EXTERNAL_MU 消息类型的示例，请参阅使用 ML-DSA 密钥对进行离线验证。

Amazon KMS 中的多区域密钥

Amazon KMS 支持多区域密钥，它们不同 Amazon Web Services 区域 中可以互换使用的 Amazon KMS keys，就好像您在多个区域中拥有相同的密钥一样。每组相关多区域密钥均具有相同的密钥材料和密钥 ID，因此您可以在一个 Amazon Web Services 区域 中将数据加密并将其解密到不同 Amazon Web Services 区域 中，而无需重新加密或跨区域调用 Amazon KMS。

与所有 KMS 密钥一样，多区域密钥永远不会使 Amazon KMS 处于未加密状态。您可以创建用于加密或签名的对称或非对称多区域密钥，创建用于生成和验证 HMAC 标签的 HMAC 多区域密钥，并创建带导入密钥材料的多区域密钥或 Amazon KMS 生成的密钥材料。您必须独立管理每个多区域密钥，包括创建别名和标签、设置其密钥策略和授权以及有选择性地启用和禁用它们。您可以在可使用单区域密钥进行的所有加密操作中使用多区域密钥。

多区域密钥是一个灵活而强大的解决方案，适用于许多常见的数据安全场景。

灾难恢复

> 在备份和恢复架构中，多区域密钥允许您在不中断的情况下处理加密数据，即使在 Amazon Web Services 区域 中断时亦可处理。备份区域中维护的数据可以在备份区域中解密，备份区域中新加密的数据可以在恢复该区域后在主区域中解密。

全球数据管理

在全球运营的企业需要将全球分布的数据一致地提供到各个 Amazon Web Services 区域。您可以在数据所在的所有区域中创建多区域密钥，然后将密钥用作单区域密钥，而不会出现跨区域调用的延迟或在每个区域中使用不同密钥重新加密数据的成本。

分布式签名应用程序

需要跨区域签名功能的应用程序可以使用多区域非对称签名密钥以在不同的 Amazon Web Services 区域 一致、反复地生成同样的数字签名。

如果将证书链与单个全局信任存储 [对于单个根证书颁发机构（CA）] 及根 CA 签名的区域中间 CA 结合使用，则不需要多区域密钥。但是，如果您的系统不支持中间 CA（如应用程序签名），则可以使用多区域密钥来实现区域证书的一致性。

跨多个区域的双活应用程序

某些工作负载和应用程序可以跨越双活架构中的多个区域。对于这些应用程序，多区域密钥可以通过提供相同的密钥材料对可能跨区域边界移动的数据进行并发加密和解密操作来降低复杂性。

您可以将多区域密钥与客户端加密库结合使用，例如 Amazon Encryption SDK、Amazon 数据库加密 SDK 和 Amazon S3 客户端加密。

与 Amazon KMS 集成以进行静态加密或数字签名的 Amazon 服务当前将多区域密钥视为单区域密钥。他们可能会重新包装或重新加密在区域之间移动的数据。例如，甚至是在复制受多区域密钥保护的对象时，Amazon S3 跨区域复制也会在目标区域的 KMS 密钥下解密和重新加密数据。

多区域键不是全局键。创建一个多区域主键，然后将其复制到您在 Amazon 分区中选择的区域。然后单独管理每个区域中的多区域键。Amazon 和 Amazon KMS 都不能代表您自动创建多区域密钥或将其复制到任何区域中。Amazon 服务在您的账户中为您创建的 KMS 密钥 Amazon 托管式密钥 始终都是单区域密钥。

对于中国区域，您可以使用多区域密钥功能在中国区域分区 (aws-cn) 内复制 KMS 密钥。例如，您可以将密钥从中国（北京）区域复制到中国（宁夏）区域，反之亦然。如果将密钥从一个中国区域复制到另一个中国区域，即表示您同意使用目标区域的 Amazon Key Management Service 并遵守目标区域的所有适用协议条款。不能将密钥从北京和宁夏区域复制到中国区域分区之外的 Amazon 区域。同样，也不能将密钥从中国区域分区之外的区域复制到北京和宁夏区域。

您不能将现有的单区域密钥转换为多区域密钥。此设计可确保使用现有单区域密钥保护的所有数据都保持相同的数据驻留和数据主权属性。

对于大多数数据安全需求，区域资源的区域隔离和容错能力使标准 Amazon KMS 单区域密钥称为最适合的解决方案。但是，当您需要跨多个区域加密或对客户端应用程序中的数据进行签名时，多区域密钥可能是解决方案。

区域

Amazon KMS 支持的所有 Amazon Web Services 区域均支持多区域密钥。

定价和配额

一组相关的多区域密钥中的每个密钥都被视为一个 KMS 密钥，用于定价和配额。Amazon KMS 配额是针对账户的每个区域单独计算的。每个区域中的多区域密钥的使用和管理将计入该区域的配额。

支持的 KMS 密钥类型

您可以创建以下类型的多区域 KMS 密钥：

- 对称加密 KMS 密钥
- 非对称 KMS 密钥
- HMAC KMS 密钥
- 具有导入密钥材料的 KMS 密钥

您不能在自定义密钥存储中创建多区域密钥。

了解更多

- 要了解如何控制对多区域 KMS 密钥的访问，请参阅 控制对多区域密钥的访问。
- 要创建任何类型的多区域主 KMS 密钥，请参阅 创建多区域主密钥。
- 要创建多区域副本 KMS 密钥，请参阅 创建多区域副本密钥。
- 要更新主区域，请参阅 更改一组多区域密钥中的主密钥。
- 要识别和查看多区域 KMS 密钥，请参阅 识别 HMAC KMS 密钥。
- 要了解删除多区域 KMS 密钥的特殊注意事项，请参阅 Deleting multi-Region keys。

术语和概念

以下术语和概念用于多区域密钥。

多区域密钥

多区域密钥是不同 Amazon Web Services 区域 中具有相同的密钥 ID 和密钥材料（以及其他共享属性）的一组 KMS 密钥之一。每个多区域密钥都是一个功能齐全的 KMS 密钥，可以完全独立于其相关的多区域密钥使用。由于所有的相关多区域密钥都具有相同的密钥 ID 和密钥材料，它们可互换操作，即任何 Amazon Web Services 区域 中的相关多区域密钥都可以解密由任何其他相关的多区域密钥加密的密文。

您可以在创建 KMS 密钥时设置其多区域属性。您不能更改现有密钥的多区域属性。您不能将单区域密钥转换为多区域密钥，或将多区域密钥转换为单区域密钥。要将现有工作负载移动到多区域方案中，您必须重新加密数据或使用新的多区域密钥创建新的签名。

多区域密钥可以是对称或非对称的，它可以使用 Amazon KMS 密钥材料或导入的密钥材料。您不能在自定义密钥存储中创建多区域密钥。

在一组相关的多区域密钥中，任何时候都只有一个主键。您可以在其他 Amazon Web Services 区域 中创建该主键的副本密钥。您还可以更新主区域，从而将主键更改为副本密钥，并将指定的副本密钥更改为主键。但是，您只能在每个 Amazon Web Services 区域 维持一个主键或副本密钥。所有区域都必须位于同一个 Amazon 分区。

您可以在相同或不同的 Amazon Web Services 区域 中拥有多组相关多区域密钥。尽管相关的多区域密钥是可互操作的，但不相关的多区域密钥不可互操作。

主键

多区域主键是一个 KMS 密钥，可以复制到同一个分区内的不同 Amazon Web Services 区域 中。每组多区域密钥只有一个主键。

主键与副本密钥的区别在以下几方面：

- 只有主键可以复制。
- 主键是其副本密钥的共享属性的来源，包括密钥材料和密钥 ID。
- 您只能在主键上启用和禁用自动密钥轮换。
- 您可以随时计划删除主键。但是，Amazon KMS 将不会删除主键，直到该主键的所有副本密钥都被删除。

不过，主密钥和副本密钥在任何加密属性中都没有区别。您可以互换使用主键及其副本密钥。

您不需要复制主键。您可以像使用任何 KMS 密钥一样使用它，并在有用时复制它。但是，由于多区域密钥与单区域密钥具有不同的安全属性，因此我们建议您仅在计划复制多区域密钥时才创建多区域密钥。

副本密钥

多区域副本密钥是一个 KMS 密钥，它具有与其主密钥和相关副本密钥相同的密钥 ID 和密钥材料，但位于不同的 Amazon Web Services 区域 中。

副本密钥是功能齐全的 KMS 密钥，具有其自己的密钥策略、授权、别名、标签和其他属性。它不是主键或任何其他密钥的副本或指针。即使某个副本密钥的主键及所有相关的副本密钥都被禁用，您也可以使用该副本密钥。您还可以将副本密钥转换为主键，将主键转换为副本密钥。副本密钥被创建后，仅依赖于其主键进行密钥轮换和更新主区域。

主密钥和副本密钥在任何加密属性中都没有区别。您可以互换使用主键及其副本密钥。通过主密钥或副本密钥加密的数据可以通过相同的密钥或任何相关的主密钥或副本密钥进行解密。

复制

您可以将多区域主键复制到同一个分区中的不同的 Amazon Web Services 区域。如果您这样做，Amazon KMS 会使用与主密钥相同的密钥 ID 和其他共享属性，在指定区域中创建一个多区域副本密钥。然后，它将密钥材料安全地跨区域边界传输，并将其与新的副本密钥相关联，一切都在 Amazon KMS 中进行。

共享属性

共享属性是与其副本密钥共享的多区域主键的属性。Amazon KMS 将创建具有与主键相同的共享属性值的副本密钥。然后，它会定期将主键的共享属性值与其副本密钥同步。您不能在副本密钥上设置这些属性。

以下是多区域密钥的共享属性。

- 密钥 ID — (密钥 ARN 的 `Region` 元素不同。)
- 密钥材料
- 密钥材料源
- 密钥规范和加密算法
- 密钥用法
- 自动密钥轮换 — 您只能在主键上启用和禁用自动密钥轮换。将使用共享密钥材料的所有版本创建新的副本密钥。有关更多信息，请参阅 Rotating multi-Region keys。

- 按需轮换 – 您只能在对主密钥执行按需轮换。将使用共享密钥材料的所有版本创建新的副本密钥。有关更多信息，请参阅 Rotating multi-Region keys。

您还可以将相关多区域密钥的主名称和副本名称视为共享属性。当您创建新的副本密钥或更新主键时，Amazon KMS 将更改同步到所有相关的多区域密钥。完成这些更改后，所有相关的多区域密钥都会准确列出其主键和副本密钥。

多区域密钥的所有其他属性都是独立属性，包括说明、密钥策略、授权、启用和禁用的密钥状态、别名和标签。您可以在所有相关的多区域密钥上为这些属性设置相同的值，但如果更改独立属性的值，Amazon KMS 不会同步它。

您可以跟踪多区域密钥的共享属性的同步情况。在您的 Amazon CloudTrail 日志中，查找 SynchronizeMultiRegionKey 事件。

多区域密钥的安全注意事项

仅在您需要 Amazon KMS 多区域密钥时使用它。多区域密钥为工作负载提供灵活且可扩展的解决方案，这些工作负载可在 Amazon Web Services 区域 之间移动加密数据或需要跨区域访问。如果您必须跨区域共享、移动或备份受保护的数据，或者需要为在不同区域运行的应用程序创建相同的数字签名，请考虑使用多区域密钥。

但是，创建多区域密钥的过程会跨 Amazon KMS 内的 Amazon Web Services 区域 边界移动您的密钥材料。由多区域密钥生成的密文可能会由多个地理位置的多个相关密钥进行解密。对于区域隔离的服务和资源也有很大的益处。每个 Amazon Web Services 区域 都是隔离的，独立于其他区域。区域提供容错能力、稳定性和弹性，还可以减少延迟。它们使您能够创建保持可用且不受其他区域中断影响的冗余资源。在 Amazon KMS 中，它们还确保每个密文只能通过一个密钥进行解密。

多区域密钥还会引发新的安全注意事项：

- 使用多区域密钥，控制访问和强制执行数据安全策略变得更加复杂。您需要确保在多个隔离区域的密钥上对策略进行一致的审计。您需要使用策略来强制实施边界，而不是依赖单独的密钥。

  例如，您需要对数据设置策略条件，以防止一个区域的薪酬团队能够读取另一个区域的工资单数据。此外，您还必须使用访问控制来防止一个区域中的多区域密钥保护一个租户的数据，而另一个区域中的相关多区域密钥保护另一个租户的数据的情况。
- 跨区域审计密钥也更为复杂。使用多区域密钥，您需要检查和协调多个区域的审计活动，以便全面了解受保护数据的关键活动。

- 遵守数据驻留要求可能会变得更加复杂。使用隔离的区域，您可以确保数据驻留和数据主权合规性。给定区域中的 KMS 密钥只能解密该区域中的敏感数据。在一个区域中加密的数据可以保持完全保护，并且在任何其他区域都无法访问。

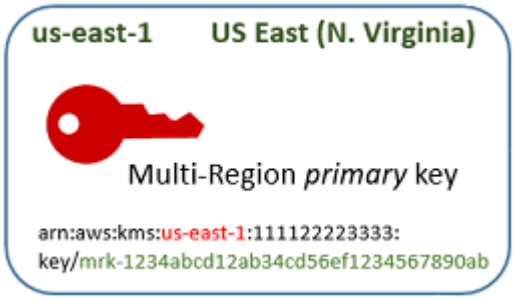
  要使用多区域密钥验证数据驻留和数据主权，您需要跨多个区域实施访问策略和编译 Amazon CloudTrail 事件。

为了使您能够更轻松地管理多区域密钥的访问控制，复制多区域密钥的权限 (kms:ReplicateKey) 独立于创建密钥的标准权限 (kms:CreateKey)。另外，Amazon KMS 支持多区域密钥的多个策略条件，包括 `kms:MultiRegion`，该条件允许或拒绝创建、使用或管理多区域密钥和 `kms:ReplicaRegion` 的权限，从而限制了多区域密钥可以复制到的区域。有关更多信息，请参阅 控制对多区域密钥的访问。

多区域密钥的工作原理

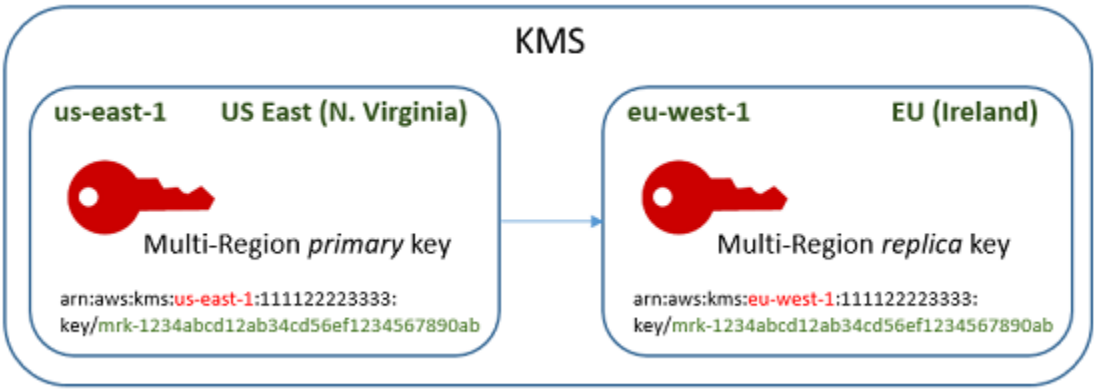
首先在 Amazon KMS 支持的 Amazon Web Services 区域 中创建一个对称或非对称多区域主键，如美国东部（弗吉尼亚州北部）区域。只有在创建密钥时，才能决定密钥是单区域还是多区域；以后不能更改此属性。与任何 KMS 密钥一样，您可以为多区域密钥设置密钥策略，并且可以创建授权，并添加别名和标签以进行分类和授权。（这些是独立属性，不与其他密钥共享或同步。）您可以在加密操作中使用多区域主键进行加密或签名。

您可以在 Amazon KMS 控制台中或使用 `MultiRegion` 参数被设置为 `true` 的 CreateKey API 创建多区域主键。请注意，多区域密钥具有一个以 `mrk-` 开头的独特密钥 ID。您可以使用 `mrk-` 前缀以编程方式识别 MRK。



如果您选择，您可以复制多区域主键到相同 Amazon分区中的一个或多个不同的 Amazon Web Services 区域 中，例如欧洲（爱尔兰）区域。当您这样做时，Amazon KMS 会使用与主键相同的密钥 ID 和其他共享属性在指定区域中创建一个副本密钥。然后，它将密钥材料安全地跨区域边界传输，并将其与目标区域中的新 KMS 密钥相关联，一切都在 Amazon KMS 中进行。结果会产生两个相关的多区域密钥（主键和副本密钥），它们可以互换使用。

您可以在 Amazon KMS 控制台中或使用 ReplicateKey API 创建多区域副本密钥。



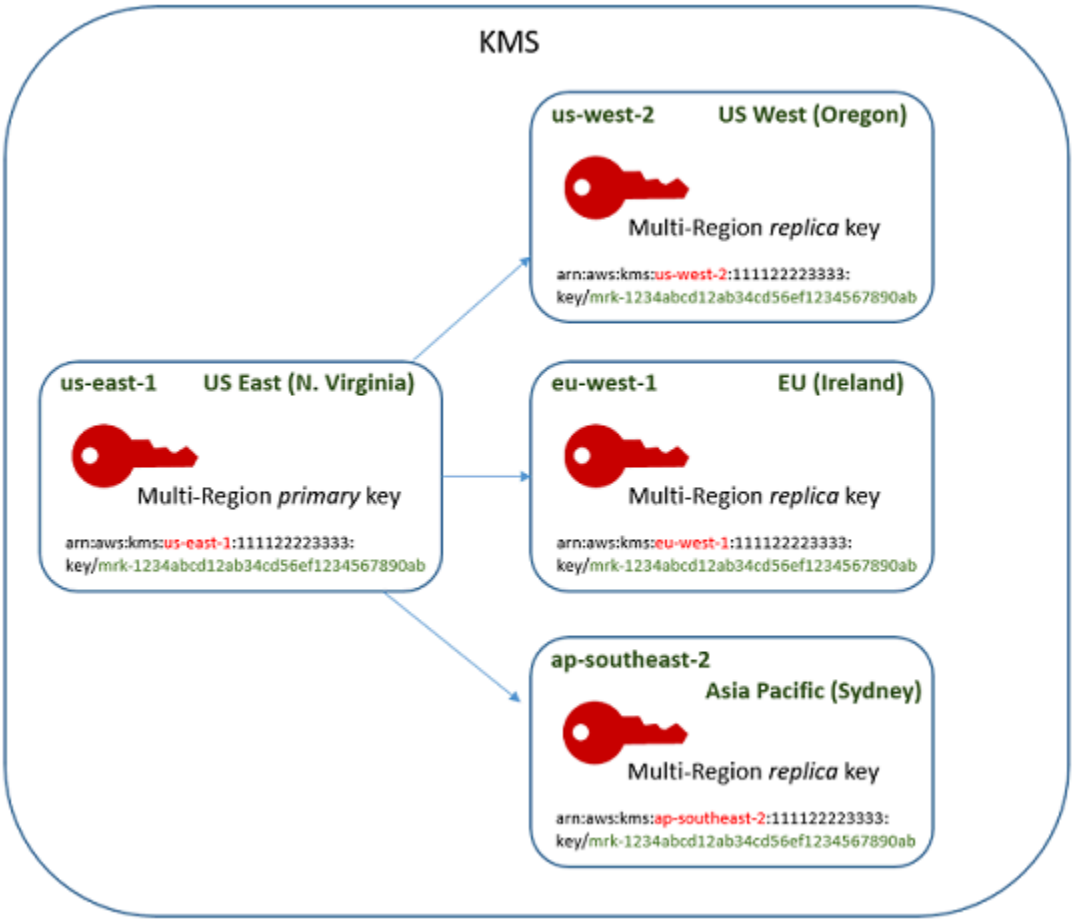
由此产生的多区域副本密钥是一个功能齐全的 KMS 密钥，具有与主键相同的共享属性。在所有其他方面，它是一个独立的 KMS 密钥，具有自己的说明、密钥策略、授权、别名和标签。启用或禁用多区域密钥对相关的多区域密钥没有影响。您可以在加密操作中独立使用主密钥和副本密钥，也可以协调它们的使用。例如，您可以使用美国东部（弗吉尼亚北部）区域的主键对数据进行加密，将数据移动到欧洲（爱尔兰）区域，然后使用副本密钥解密数据。

相关的多区域密钥具有相同的密钥 ID。它们的密钥 ARN（Amazon Resource Name）仅在 Region（区域）字段中有所不同。例如，多区域主键和副本密钥可能具有以下示例密钥 ARN。密钥 ID（密钥 ARN 中的最后一个元素）是相同的。两个密钥都具有多区域密钥的独特密钥 ID，该 ID 以 mrk- 开头。

```
Primary key: arn:aws:kms:us-
east-1:111122223333:key/mrk-1234abcd12ab34cd56ef12345678990ab
Replica key: arn:aws:kms:eu-
west-1:111122223333:key/mrk-1234abcd12ab34cd56ef12345678990ab
```

为了实现互操作性，需要具有相同的密钥 ID。加密时，Amazon KMS 将 KMS 密钥的密钥 ID 绑定到密文，因此只能使用该 KMS 密钥或具有相同密钥 ID 的 KMS 密钥来解密密文。此功能还使相关的多区域密钥易于识别，并且可以更轻松地互换使用它们。例如，在应用程序中使用它们时，您可以通过它们的共享密钥 ID 来引用相关的多区域密钥。然后，在必要时，指定区域或 ARN 以区分它们。

随着数据需求的变化，您可以将主键复制到相同分区中的其他 Amazon Web Services 区域，如美国西部（俄勒冈）和亚太地区（悉尼）区域。结果会产生四个具有相同的密材料和密钥 ID 的相关多区域密钥，如下图所示。您可以独立管理密钥。您可以独立使用它们，也可以以协调的方式使用它们。例如，您可以在亚太地区（悉尼）区域使用副本密钥对数据进行加密，将数据移动到美国西部（俄勒冈）区域，然后在美国西部（俄勒冈）区域使用副本密钥对其进行解密。



多区域密钥的其他注意事项包括以下内容。

同步共享属性 — 如果多区域密钥的共享属性发生更改，Amazon KMS 会自动同步从主键到其所有副本密钥的更改。您不能请求或强制执行共享属性的同步。Amazon KMS 会为您检测并同步所有更改。但是，您可以使用 CloudTrail 日志中的 SynchronizeMultiRegionKey 事件审计同步。

例如，如果在对称多区域主键上启用自动密钥轮替，Amazon KMS 会将该设置复制到其所有副本密钥中。轮换密钥材料时，轮换将在所有相关的多区域密钥之间同步，因此它们继续具有相同的当前密钥材料，并可访问所有旧版本的密钥材料。如果创建新的副本密钥，则该密钥具有与所有相关多区域密钥相同的当前密钥材料，并可访问密钥材料的所有以前版本。有关更多信息，请参阅 Rotating multi-Region keys。

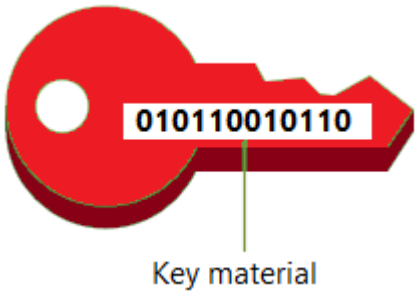
更改主键 — 每组多区域密钥必须只有一个主键。主键是唯一可以复制的密钥。它也是其副本密钥的共享属性的来源。但是，您可以将主键更改为副本密钥，并将其中一个副本密钥提升为主键。您可以执行此操作，以便从特定区域删除多区域主键，或将主键定位在离项目管理员更近的区域中。有关更多信息，请参阅 更改一组多区域密钥中的主密钥。

删除多区域密钥 — 与所有 KMS 密钥一样，您必须在 Amazon KMS 将其删除之前计划多区域密钥的删除。密钥处于待删除状态时，您无法在任何加密操作中使用它。然而，Amazon KMS 将不会删除多区域主键，直到该主键的所有副本密钥都被删除。有关详细信息，请参阅Deleting multi-Region keys。

导入密钥的 Amazon KMS 密钥材料

您可以使用您提供的密钥材料创建 Amazon KMS keys （ KMS 密钥 ）。

KMS 密钥是数据密钥的逻辑表示形式。KMS 密钥的元数据包括用于执行加密操作的密钥材料的 ID。在创建 KMS 密钥时，默认情况下， Amazon KMS 会生成该 KMS 密钥的密钥材料。但是，您可以创建不带密钥材料的 KMS 密钥，然后将自己的密钥材料导入该 KMS 密钥中，这个功能通常被称为“自带密钥 (BYOK)”。



> **Note**
>
> Amazon KMS 不支持解密通过外部对称加 Amazon KMS 密 KMS 密钥加密的任何密文 Amazon KMS，即使密文是使用导入的密钥材料在 KMS 密钥下加密的。 Amazon KMS 不会发布此任务所需的密文格式，并且格式可能会更改，恕不另行通知。

当您使用导入的密钥材料时，您仍需对密钥材料负责，同时 Amazon KMS 允许使用密钥材料的副本。出于以下一个或多个原因，您可以选择执行该操作：

- 证明使用符合您要求的熵源生成了密钥材料。
- 将您自己的基础设施中的密钥材料与 Amazon 服务一起使用，并用于管理 Amazon KMS 其中的密钥材料的生命周期 Amazon。
- 在中使用现有的、成熟的密钥 Amazon KMS，例如用于代码签名、PKI 证书签名和证书固定的应用程序的密钥
- 为中的密钥材料设置过期时间 Amazon 并手动将其删除，但也可以使其在将来再次可用。相比之下，计划密钥删除需要 7 到 30 天的等待期限，之后，您不能恢复已删除的 KMS 密钥。

- 拥有密钥材料的原始副本，并将其保存在外部，以便在密钥材料 Amazon 的整个生命周期中提高耐用性和灾难恢复。
- 对于非对称密钥和 HMAC 密钥，导入会创建兼容且可互操作的密钥，这些密钥可在内部和外部运行。 Amazon

支持的 KMS 密钥类型

Amazon KMS 支持以下类型的 KMS 密钥的导入密钥材料。您无法将密钥材料导入自定义密钥存储中的 KMS 密钥。

- 对称加密 KMS 密钥
- 非对称 KMS 密钥（ML-DSA 密钥除外）
- HMAC KMS 密钥
- 支持的所有类型的多区域密钥。

区域

所有支持的密钥材料均 Amazon Web Services 区域 支持导入的密钥材料。 Amazon KMS

在中国区域，对称加密 KMS 密钥的密钥材料要求与其他区域不同。有关更多信息，请参阅 步骤 3：加密密钥材料。

了解更多

- 要创建具有导入密钥材料的 KMS 密钥，请参阅删除具有导入密钥材料的 KMS 密钥。
- 要创建警报以在 KMS 密钥中导入的密钥材料即将到期时通知您，请参阅针对导入密钥材料的到期创建 CloudWatch 警报。
- 要将密钥材料重新导入 KMS 密钥中，请参阅重新导入密钥材料。
- 要将新密钥材料导入 KMS 密钥以进行按需轮换，请参阅导入新密钥材料和执行按需密钥轮换。
- 要识别和查看带导入的密钥材料的 KMS 密钥，请参阅识别带导入的密钥材料的 KMS 密钥。
- 如需了解删除带导入的密钥材料的 KMS 密钥的特殊注意事项，请参阅Deleting KMS keys with imported key material。

导入的密钥材料的特殊注意事项

在决定将密钥材料导入之前 Amazon KMS，您应该了解导入的密钥材料的以下特征。

您可以生成密钥材料

您负责使用符合您的安全要求的随机源来生成密钥材料。

您需要对可用性和持久性负责

Amazon KMS 旨在保持导入的密钥材料的高可用性。但是 Amazon KMS 不能将进口密钥材料的耐久性保持在与 Amazon KMS 生成的密钥材料相同的水平。有关更多信息，请参阅 保护导入的密钥材料。

您可以删除密钥材料

您可以从 KMS 密钥中删除导入的密钥材料，以便立即使 KMS 密钥不可用。此外，当您将密钥材料导入 KMS 密钥中时，您可以确定密钥是否到期，并设置其到期时间。到期时间到来时， Amazon KMS 删除密钥材料。如果没有密钥材料，则该 KMS 密钥无法用于任何加密操作。要还原密钥，必须将相同密钥材料重新导入到密钥中。

您无法更改非对称密钥和 HMAC 密钥的密钥材料

当您将密钥材料导入 KMS 密钥中时，该 KMS 密钥将与该密钥材料永久关联。您可以重新导入相同的密钥材料，但不能将不同的密钥材料导入该 KMS 密钥。而且，您不能为具有导入密钥材料的 KMS 密钥启用自动密钥轮换。但是，您可以手动轮换带有导入密钥材料的 KMS 密钥。

您可以按需轮换对称加密密钥

带有导入密钥材料的对称加密密钥支持按需轮换。您可以将多个密钥材料导入此类密钥，并使用按需轮换来更新当前密钥材料。当前密钥材料用于加密和解密，但其他（非当前）密钥材料只能用于解密。

您无法更改密钥材料源

专用于导入的密钥材料的 KMS 密钥拥有一个 EXTERNAL 的源值，该值无法更改。您不能将导入的密钥材料的 KMS 密钥转换为使用任何其他来源（包括）的密钥材料 Amazon KMS。同样，您不能将包含密钥材料的 KMS Amazon KMS 密钥转换为专为导入的密钥材料而设计的密钥。

您无法导出密钥材料

您无法导出您导入的任何密钥材料。 Amazon KMS 无法以任何形式将导入的密钥材料退还给您。您必须在外部保存导入的密钥材料的副本 Amazon，最好是在密钥管理器中，例如硬件安全模块 (HSM)，以便在删除密钥材料或密钥材料过期时可以重新导入。

您可创建具有导入密钥材料的多区域密钥

具有导入密钥材料的多区域具有包含导入密钥材料的 KMS 密钥的功能，并且可以在 Amazon Web Services 区域之间进行互操作。要创建具有导入密钥材料的多区域密钥，您必须将相同的密钥材料

导入 KMS 主密钥和每个副本密钥。有关导入多区域密钥的密钥材料的更多详细信息，请参阅Multi-Region keys。

非对称密钥和 HMAC 密钥具有可移植性和互操作性

您可以在外部使用非对称密钥材料和 HMAC 密钥材料，与具有相同导入密钥材料的 Amazon Amazon KMS 密钥进行互操作。

与 Amazon KMS 对称密文不同，对称密文与算法中使用的 KMS 密钥密不可分，它 Amazon KMS 使用标准 HMAC 和非对称格式进行加密、签名和 MAC 生成。因此，这些密钥是可移植的，并且支持传统的托管密钥方案。

当您的 KMS 密钥导入了密钥材料后，您可以使用外部导入的密钥材料 Amazon 来执行以下操作。

- HMAC 密钥 – 您可以使用导入的密钥材料验证由 HMAC KMS 密钥生成的 HMAC 标签。您还可以将 HMAC KMS 密钥与导入的密钥材料一起使用，以验证由外部的密钥材料生成的 HMAC 标签。 Amazon
- 非对称加密密钥 — 您可以使用外部的私有非对称加密密钥 Amazon 来解密由 KMS 密钥和相应的公钥加密的密文。您还可以使用非对称 KMS 密钥来解密在外部生成的非对称密文。 Amazon
- 非对称签名密钥 — 您可以使用带有导入密钥材料的非对称签名 KMS 密钥来验证由您的私有签名密钥在外部生成的数字签名。 Amazon您还可以在外部使用非对称公有签名密钥 Amazon 来验证非对称 KMS 密钥生成的签名。
- 非对称密钥协议密钥 – 您可以使用带导入的密钥材料的非对称密钥协议 KMS 密钥来与 Amazon 外的对等方派生共享密钥。

如果您在相同的 Amazon Web Services 区域中将相同的密钥材料导入不同的 KMS 密钥中，则这些密钥也是可以互操作的。要在不同版本中创建可互操作的 KMS 密钥 Amazon Web Services 区域，请使用导入的密钥材料创建多区域密钥。

RSA 私有密钥

- Amazon KMS 要求导入的 RSA 私钥具有符合 FIPS 186-5 第 A 节中描述的测试的主要因素。1.3。其他软件或设备可能会使用其他算法来验证 RSA 私有密钥的这些质因子。在极少数情况下，使用其他算法验证的密钥可能不会被 Amazon KMS接受。

对称加密密钥不可移植或互操作

生 Amazon KMS 成的对称密文不可移植或互操作。 Amazon KMS 不会发布可移植性所需的对称密文格式，并且格式可能会更改，恕不另行通知。

- Amazon KMS 即使您使用已导入的密钥材料，也无法解密您在外部加密的 Amazon对称密文。

- Amazon KMS 不支持解密外部的任何 Amazon KMS 对称密文，即使密文是在 KMS 密钥下使用导入的密钥材料加密的。 Amazon KMS
- 具有相同导入密钥材料的 KMS 密钥不可互操作。 Amazon KMS 生成每个 KMS 密钥特有的密文的对称密文。此加密文字格式保证只有加密数据的 KMS 密钥才能解密数据。

此外，您不能使用任何 Amazon 工具（例如Amazon Encryption SDK或 Amazon S3 客户端加密）来解密对称密文 Amazon KMS 。

因此，您不能使用带有导入密钥材料的密钥来支持密钥托管安排，在这种安排中，有权有条件访问密钥材料的授权第三方可以在外部解密某些密文。 Amazon KMS要支持密钥托管，请使用 Amazon Encryption SDK 来通过独立于 Amazon KMS的密钥加密您的消息。

保护导入的密钥材料

您导入的密钥材料在传输中和静态时都受到保护。在导入密钥材料之前，您需要使用在 FIPS 140-3 加密模块验证计划下验证的 Amazon KMS 硬件安全模块 () 中生成的 RSA 密钥对的公钥来加密（或 “包装”HSMs）密钥材料。您可以使用包装公有密钥直接加密密钥材料，也可以使用 AES 对称密钥加密密钥材料，然后使用 RSA 公有密钥加密 AES 对称密钥。

收到后，使用 HSM 中的相应私钥对 Amazon KMS 密钥材料进行解密，然后使用仅存在于 Amazon KMS HSM 易失性存储器中的 AES 对称密钥对其进行重新加密。您的密钥材料绝不会让 HSM 处于纯文本状态。它仅在使用时解密，并且仅在使用中解密。 Amazon KMS HSMs

您的 KMS 密钥与导入的密钥材料的使用完全取决于您在 KMS 密钥上设置的访问控制策略。此外，您还可以使用别名和标签来识别和控制对 KMS 密钥的访问。您可以使用 Amazon CloudTrail等服务启用和禁用密钥，以及查看和监控密钥。

但是，您将保留密钥材料的唯一故障保护副本。作为这种额外控制措施的回报，您应对进口密钥材料的耐用性和整体可用性负责。 Amazon KMS 旨在保持导入的密钥材料的高可用性。但是 Amazon KMS 不能将进口密钥材料的耐久性保持在与 Amazon KMS 生成的密钥材料相同的水平。

在以下情况下，这种持久性的差异是有意义的：

- 当您为导入的密钥材料设置过期时间时，将在密钥材料到期后将其 Amazon KMS 删除。 Amazon KMS 不会删除 KMS 密钥或其元数据。您可以创建一个 Amazon CloudWatch 警报，在导入的密钥材料即将到期时通知您。

  您无法删除为 KMS 密钥 Amazon KMS 生成的密钥材料，也不能将 Amazon KMS 密钥材料设置为过期。

- 手动删除导入的密钥材料时，Amazon KMS 会删除密钥材料，但不会删除 KMS 密钥或其元数据。相比之下，计划删除密钥需要等待 7 到 30 天，之后会 Amazon KMS 永久删除 KMS 密钥、其元数据和密钥材料。
- 万一发生某些影响整个地区的故障 Amazon KMS（例如完全断电），Amazon KMS 则无法自动恢复导入的密钥材料。但是，Amazon KMS 可以恢复 KMS 密钥及其元数据。

您必须在您控制的系统之外保留一份导入 Amazon 的密钥材料的副本。我们建议您将导入的密钥材料的可导出副本存储在密钥管理系统中，例如 HSM。根据最佳实践要求，应将 Amazon KMS 生成的对 KMS 密钥 ARN 和生成的密钥材料 ID 的引用与该密钥材料的可导出副本一同保存。如果您导入的密钥材料被删除或过期，则其关联的 KMS 密钥将无法使用，直到您重新导入相同的密钥材料。如果您导入的密钥材料永久丢失，则以 KMS 密钥加密的任何加密文字都将无法恢复。

> **Important**
>
> 对称加密密钥可以有多个与之关联的密钥材料。一旦您删除其中任何一个密钥材料或其中任何一个密钥材料过期（除非已删除或即将到期的密钥材料为或），则整个 KMS 密钥将无法使用。`PENDING_ROTATION` `PENDING_MULTI_REGION_IMPORT_AND_ROTATION`您必须重新导入与密钥关联的所有已过期或已删除密钥材料，然后该密钥才能用于密码操作。

CloudHSM 密钥存储中的 KMS 密钥

您可以在 Amazon CloudHSM 密钥存储中创建、查看、管理、使用和计划删除 Amazon KMS keys。您使用的过程与用于其他 KMS 密钥的过程非常相似。唯一的区别是您在创建 KMS 密钥时指定了 Amazon CloudHSM 密钥存储。然后，Amazon KMS 在与 Amazon CloudHSM 密钥存储关联的 Amazon CloudHSM 集群中为 KMS 密钥创建不可提取的密钥材料。在 Amazon CloudHSM 密钥存储中使用 KMS 密钥时，会在集群中的 HSM 中执行加密操作。

支持的功能

除了此部分中讨论的过程之外，您还可以使用 Amazon CloudHSM 密钥存储中的 KMS 密钥执行下列操作：

- 使用密钥策略、IAM policy 和授权来授予对 KMS 密钥的访问权限。
- 启用和禁用 KMS 密钥。
- 分配标签并创建别名，然后使用基于属性的访问权限控制（ABAC）授予对 KMS 密钥的访问权限。
- 使用 KMS 密钥执行以下加密操作：

- Encrypt
- Decrypt
- GenerateDataKey
- GenerateDataKeyWithoutPlaintext
- ReEncrypt

自定义密钥存储不支持生成非对称数据密钥对操作，如 GenerateDataKeyPair 和 GenerateDataKeyPairWithoutPlaintext。

- 将 KMS 密钥和与 Amazon KMS 集成的 Amazon 服务结合使用并支持客户托管密钥。
- 在 Amazon CloudTrail 日志和 Amazon CloudWatch 监控工具中跟踪 KMS 密钥的使用情况。

不支持的 特征

- Amazon CloudHSM 密钥存储仅支持对称加密 KMS 密钥。您无法在 Amazon CloudHSM 密钥存储中创建 HMAC KMS 密钥、非对称 KMS 密钥或非对称数据密钥对。
- 您无法向 Amazon CloudHSM 密钥存储中的 KMS 密钥导入密钥材料。Amazon KMS 为 Amazon CloudHSM 集群中的 KMS 密钥生成密钥材料。
- 您无法启用或禁用 Amazon CloudHSM 密钥存储中 KMS 密钥的密钥材料的自动轮换。

在 Amazon CloudHSM 密钥存储中使用 KMS 密钥

当您在请求中使用 KMS 密钥时，按 KMS 密钥的 ID 或别名对其进行标识；您无需指定 Amazon CloudHSM 密钥存储或 Amazon CloudHSM 集群。响应包含为任何对称加密 KMS 密钥返回的相同字段。

但是，在 Amazon CloudHSM 密钥存储中使用 KMS 密钥时，加密操作完全在与 Amazon CloudHSM 密钥存储关联的 Amazon CloudHSM 集群中执行。该操作使用集群中与您选择的 KMS 密钥关联的密钥材料。

要做到这一点，必须满足以下条件。

- KMS 密钥的密钥状态必须为 `Enabled`。要查找密钥状态，请使用 Amazon KMS 控制台中的状态字段或 DescribeKey 响应中的 `KeyState` 字段。
- Amazon CloudHSM 密钥存储必须连接到其 Amazon CloudHSM 集群。其在 Amazon KMS 控制台中的 Status（状态）或在 DescribeCustomKeyStores 响应中的 `ConnectionState` 必须为 `CONNECTED`。
- 与自定义密钥存储关联的 Amazon CloudHSM 集群必须包含至少一个活动 HSM。要查找集群中的活动 HSM 的数量，请使用 Amazon KMS 控制台、Amazon CloudHSM 控制台或 DescribeClusters 操作。

- Amazon CloudHSM 集群必须包含 KMS 密钥的密钥材料。如果已从集群中删除密钥材料，或者已从未包含密钥材料的备份中创建 HSM，则加密操作将失败。

如果未满足这些条件，则加密操作失败，并且 Amazon KMS 会返回 `KMSInvalidStateException` 异常。通常，您只需重新连接 Amazon CloudHSM 密钥存储。有关其他帮助，请参阅如何修复失败的 KMS 密钥。

在 Amazon CloudHSM 密钥存储中使用 KMS 密钥时，请注意每个 Amazon CloudHSM 密钥存储中的 KMS 密钥针对加密操作共享自定义密钥存储请求限额。如果您超过该配额，则 Amazon KMS 将返回 `ThrottlingException`。如果与 Amazon CloudHSM 密钥存储关联的 Amazon CloudHSM 集群正在处理大量命令（包括与 Amazon CloudHSM 密钥存储不相关的命令），则您可能以较低速率获得 `ThrottlingException`。如果您收到任何请求的 `ThrottlingException`，请降低您的请求速率并重试这些命令。有关自定义密钥存储请求限额的详细信息，请参阅 自定义密钥存储请求限额。

了解更多

- 要了解有关 Amazon CloudHSM 密钥存储的更多信息，请参阅 Amazon CloudHSM 密钥存储。
- 要在 Amazon CloudHSM 密钥存储中创建 KMS 密钥，请参阅 在密钥库中创建 KMS Amazon CloudHSM 密钥。
- 要识别和查看 Amazon CloudHSM 密钥存储中的 KMS 密钥，请参阅 识别密钥库中的 KMS Amazon CloudHSM 密钥。
- 要查找 Amazon CloudHSM 密钥存储中的 KMS 密钥和密钥材料，请参阅 在 Amazon CloudHSM 密钥存储中查找 KMS 密钥。
- 要了解删除 Amazon CloudHSM 密钥存储中的 KMS 密钥的特殊注意事项，请参阅从 Amazon CloudHSM 密钥存储中删除 KMS 密钥。

外部密钥存储中的 KMS 密钥

若要在外部密钥存储中创建、查看、管理、使用和计划删除 KMS 密钥，您使用的过程与用于其他 KMS 密钥的过程非常相似。但是，在外部密钥存储中创建 KMS 密钥时，您需要指定外部密钥存储和外部密钥。在外部密钥存储中使用 KMS 密钥时，外部密钥管理器将使用指定的外部密钥执行加密和解密操作。

Amazon KMS 无法在外部密钥管理器中创建、查看、更新或删除任何加密密钥。Amazon KMS 绝不会直接访问您的外部密钥管理器或任何外部密钥。所有加密操作请求均由您的外部密钥存储代理调解。若要在外部密钥存储中使用 KMS 密钥，必须先将托管 KMS 密钥的外部密钥存储连接到其外部密钥存储代理。

支持的特征

除了此部分中讨论的过程之外，您还可以使用外部密钥存储中的 KMS 密钥执行下列操作：

- 使用密钥策略、IAM policy 和授权以控制对 KMS 密钥的访问。
- 启用和禁用 KMS 密钥。这些操作不会影响外部密钥管理器中的外部密钥。
- 分配标签并创建别名，然后使用基于属性的访问权限控制（ABAC）授予对 KMS 密钥的访问权限。
- 使用 KMS 密钥执行以下加密操作：
  - Encrypt
  - Decrypt
  - GenerateDataKey
  - GenerateDataKeyWithoutPlaintext
  - ReEncrypt

  自定义密钥存储不支持生成非对称数据密钥对操作，如 GenerateDataKeyPair 和 GenerateDataKeyPairWithoutPlaintext。
- 将 KMS 密钥和与 Amazon KMS 集成的 Amazon Web Services 服务 结合使用并支持客户托管密钥。

不支持的 特征

- 外部密钥存储仅支持对称加密 KMS 密钥。您无法在外部密钥存储中创建 HMAC KMS 密钥或非对称 KMS 密钥。
- 外部密钥存储中的 KMS 密钥不支持 GenerateDataKeyPair 和 GenerateDataKeyPairWithoutPlaintext。
- 您不能使用 AWS::KMS::Key Amazon CloudFormation 模板创建外部密钥存储或在外部密钥存储中创建 KMS 密钥。
- 外部密钥存储不支持多区域密钥。
- 外部密钥存储不支持拥有导入密钥材料的 KMS 密钥。
- 外部密钥存储中的 KMS 密钥不支持自动密钥轮换。

在外部密钥存储中使用 KMS 密钥

在请求中使用 KMS 密钥时，请通过密钥 ID、密钥 ARN、别名或别名 ARN 识别 KMS 密钥。您无需指定外部密钥存储。响应包含为任何对称加密 KMS 密钥返回的相同字段。但是，在外部密钥存储中使用 KMS 密钥时，外部密钥管理器将使用与 KMS 密钥关联的外部密钥执行加密和解密操作。

为了确保外部密钥存储中经 KMS 密钥加密的加密文字至少与经标准 KMS 密钥加密的加密文字一样安全，Amazon KMS 将使用双重加密。首先在 Amazon KMS 中使用 Amazon KMS 密钥材料对数据进行加密。然后，外部密钥管理器使用 KMS 密钥的外部密钥对其进行加密。若要解密双重加密的加密文字，首先由外部密钥管理器使用 KMS 密钥的外部密钥解密加密文字。然后在 Amazon KMS 中使用 KMS 密钥的 Amazon KMS 密钥材料进行解密。

要做到这一点，必须满足以下条件。

- KMS 密钥的密钥状态必须为 `Enabled`。要查找密钥状态，请查看 Amazon KMS 控制台中客户托管自主管理型密钥的状态字段或 DescribeKey 响应中的 `KeyState` 字段。
- 托管 KMS 密钥的外部密钥存储必须连接到其外部密钥存储代理，也就是说，外部密钥存储的连接状态必须为 `CONNECTED`。

  您可以在 Amazon KMS 控制台中的 External key stores（外部密钥存储）页面上或在 DescribeCustomKeyStores 响应中查看连接状态。外部密钥存储的连接状态也显示在 Amazon KMS 控制台中 KMS 密钥的详细信息页面上。在详细信息页面上，选择 Cryptographic configuration（加密配置）选项卡，然后查看 Custom key store（自定义密钥存储）部分中的 Connection state（连接状态）字段。

  如果连接状态为 `DISCONNECTED`，则必须先将其连接。如果连接状态为 `FAILED`，则必须解决问题，断开外部密钥存储的连接，再将其连接。有关说明，请参阅连接和断开外部密钥存储。
- 外部密钥存储代理必须能够找到外部密钥。
- 必须启用外部密钥并且必须执行加密和解密操作。

  外部密钥的状态独立于 KMS 密钥，并且不受 KMS 密钥的密钥状态变化影响，包括启用和禁用 KMS 密钥。同样，禁用或删除外部密钥不会更改 KMS 密钥的密钥状态，但使用关联的 KMS 密钥执行的加密操作将失败。

如果未满足这些条件，则加密操作失败，并且 Amazon KMS 会返回 `KMSInvalidStateException` 异常。您可能需要重新连接外部密钥存储或使用外部密钥管理器工具来重新配置或修复外部密钥。有关其他帮助，请参阅the section called “排查外部密钥存储的问题”。

在外部密钥存储中使用 KMS 密钥时，请注意每个外部密钥存储中的 KMS 密钥针对加密操作共享自定义密钥存储请求限额。如果您超过该配额，则 Amazon KMS 将返回 `ThrottlingException`。有关自定义密钥存储请求限额的详细信息，请参阅 自定义密钥存储请求限额。

了解更多

- 要了解有关外部密钥存储的更多信息，请参阅 外部密钥存储。

- 要了解有关外部密钥存储中密钥材料的更多信息，请参阅 外部密钥。
- 要在外部密钥存储中创建 KMS 密钥，请参阅 在外部密钥存储中创建 KMS 密钥。
- 要识别和查看外部密钥存储中的 KMS 密钥，请参阅 识别外部密钥存储中的 KMS 密钥。
- 要了解删除外部密钥存储中的 KMS 密钥的特殊注意事项，请参阅从外部密钥存储中删除 KMS 密钥。

Amazon KMS 密码学要点

Amazon KMS 使用可配置的加密算法，因此系统可以从一种已批准的算法或模式快速迁移到另一种已批准的算法或模式。初始原定设置加密算法集是从联邦信息处理标准（经 FIPS 批准）算法中选择的，用于确保其安全属性和性能。

要详细了解首选和可接受的密码算法，请参阅支持的加密算法。

熵和随机数生成

Amazon KMS 密钥生成是在中执行的 Amazon KMS HSMs。 HSMs 实现使用 NIST SP800-90A Deterministic Random Bit Generator (DRBG) CTR_DRBG using AES-256 的混合随机数生成器。它采用 384 位熵的非确定性随机位生成器进行植入，并使用额外的熵进行更新，以便在每次调用加密材料时提供预测阻力。

对称密钥操作（仅加密）

其中使用的所有对称密钥加密命令都 HSMs 使用高级加密标准 (AES)，在 G alois 计数器模式 (GCM) 中使用 256 位密钥。解密的类似调用使用反函数。

AES-GCM 是一种经过身份验证的加密方案。除了对明文进行加密以生成密文外，它还计算密文上的身份验证标签和需要身份验证的任何其他数据（附加身份验证数据或 AAD）。身份验证标签有助于确保数据来自声称的来源，并且密文和 AAD 未被修改。

通常，我们的描述中会 Amazon 忽略包含 AAD，尤其是在提及数据密钥的加密时。在这些情况下，周围的文本暗示要加密的结构在要加密的明文和要保护的明文 AAD 之间进行分区。

Amazon KMS 提供了将密钥材料导入到 Amazon KMS key 而不是依赖 Amazon KMS 生成密钥材料的选项。可以使用 RSAES-OAEP 对导入的密钥材料进行加密，以便在传输到 HS M 的过程中保护密钥。 Amazon KMS RSA 密钥对是在上 Amazon KMS HSMs生成的。导入的密钥材料在 Amazon KMS HSM 上解密，然后在 AES-GCM 下重新加密，然后由服务存储。

非对称密钥操作（加密、数字签名和签名验证）

Amazon KMS 支持使用非对称密钥操作进行加密、数字签名和密钥协议操作。非对称密钥操作依赖于数学上相关的公有密钥和私有密钥对，可用于加密和解密、签名和签名验证或派生共享密钥。私钥永远不会处于 Amazon KMS 未加密状态。您可以 Amazon KMS 通过调用 Amazon KMS API 操作在内部使用公钥，也可以下载公钥并在外部使用 Amazon KMS。

Amazon KMS 支持以下非对称密码。

- RSA-OAEP（用于加密）与 RSA-PSS 和 RSA-PKCS-#1-v1_5（用于签名和验证）- 支持 RSA 密钥长度（以位为单位）：2048、3072 和 4096；从而满足不同的安全要求。
- 椭圆曲线（ECC）– 用于签名和验证或派生共享密钥，但不能两者同时使用。支持 ECC 曲线：NIST P256、P384、P521、SECP 256k1、Ed25519。
- ML-DSA：用于签名和验证。支持的 ML-DSA 密钥规范包括：ML_DSA_44、ML_DSA_65 和 ML_DSA_87。
- SM2 （仅限中国区域）— 用于加密和解密、签名和验证或获取共享密钥，但您必须选择一种密钥用法。支持 SM2 PKE 进行加密，支持 SM2 DSA 进行签名。

密钥派生函数

密钥派生函数用于从初始密钥或密钥中派生其他密钥。 Amazon KMS 使用密钥派生函数 (KDF) 来推导每个调用的密钥，以获取下每个加密的密钥。 Amazon KMS key所有 KDF 操作在计数器模式下使用 KDF，使用带有 [0] 的 HMAC SHA256 [FIPS197FIPS18]。256 位派生密钥与 AES-GCM 一起使用，用于加密或解密客户数据和密钥。

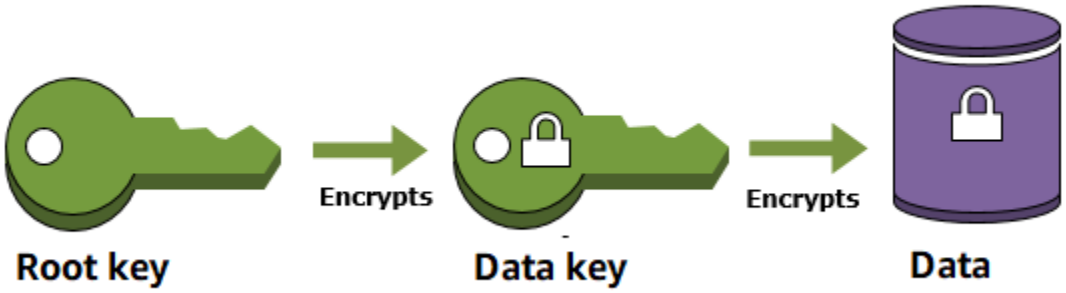
Amazon KMS 内部使用数字签名

数字签名还用于验证 Amazon KMS 实体之间的命令和通信。所有服务实体都有一个椭圆曲线数字签名算法 (ECDSA) 密钥对。它们执行 ECDSA，如 Use of Elliptic Curve Cryptography (ECC) Algorithms in Cryptographic Message Syntax (CMS)（在加密消息语法 [CMS] 中使用椭圆曲线加密 [ECC]）和 X9.62-2005：Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)（金融服务行业的公有密钥加密：椭圆曲线数字签名算法 [ECDSA]）中所定义。这些实体使用联邦信息处理标准出版物 FIPS PUB 180-4 中定义的安全哈希算法，即。SHA384这些密钥在曲线 secp384r1 (NIST-P384) 上生成。

信封加密

在您加密数据后，数据将受到保护，但您必须保护加密密钥。一种策略是对其进行加密。信封加密 是一种加密方法，它使用数据密钥对明文数据进行加密，然后使用其他密钥对数据密钥进行加密。

您甚至可以使用其他加密密钥对数据加密密钥进行加密，并且在另一个加密密钥下加密该加密密钥。但是，最后，一个密钥必须以明文形式保留，以便您可以解密密钥和数据。此顶级明文密钥加密密钥称为根密钥。



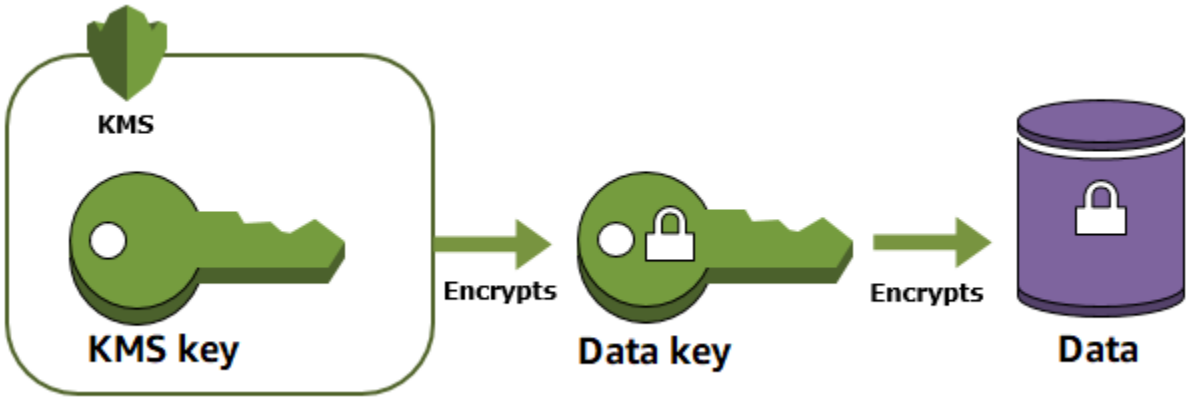
Amazon KMS 通过安全地存储和管理加密密钥来帮助您保护密钥。存储在 Amazon KMS名为 Amazon KMS keys Amazon KMS FIPS 140-3 安全等级 3 的硬件安全模块中的根密钥永远不要加密。要使用 KMS 密钥，必须致电 Amazon KMS。

许多加密系统中使用的基本结构是信封加密。信封加密使用两个或更多加密密钥来保护消息。通常，一个密钥派生自较长期间的静态密钥 k，另一个密钥是每条消息密钥 msgKey，该密钥生成以加密消息。信封通过加密以下消息形成：ciphertext = Encrypt(msgKey, message)。然后，消息密钥使用长期静态密钥进行加密：encKey = Encrypt(k, msgKey)。最后，这两个值 (encKey, ciphertext) 打包成一个结构，或信封加密的消息。

具有 k 访问权限的收件人可以打开信封加密的消息，方法是首先解密加密的密钥，然后解密消息。

Amazon KMS 提供了管理这些长期静态密钥和自动执行数据信封加密过程的能力。

除了 Amazon KMS 服务中提供的加密功能外，Encryption Amazon SDK 还提供客户端信封加密库。您可以使用这些库来保护您的数据和用于加密该数据的加密密钥。



信封加密可提供以下多种优势：

- 保护数据密钥

  加密数据密钥时，您无需担心存储加密数据密钥，因为数据密钥本身就受到加密的保护。您可以安全地将加密数据与加密数据密钥一起存储。

- 使用多个密钥加密相同的数据

  加密操作可能非常耗时，特别是要加密的数据是大型对象时。您可以只重新加密保护原始数据的数据密钥，而无需使用不同的密钥多次重新加密原始数据。

- 结合多种算法的优势

  通常，与公有密钥算法相比，对称密钥算法速度更快，生成的密文更小。但公有密钥算法可提供固有的角色分离和更轻松的密钥管理。信封加密让您可以每种策略的优势结合起来。

加密操作

在中 Amazon KMS，加密操作是使用 KMS 密钥保护数据的 API 操作。由于 KMS 密钥保留在其中 Amazon KMS，因此您必须调用 Amazon KMS 才能在加密操作中使用 KMS 密钥。

要使用 KMS 密钥执行加密操作，请使用 Amazon SDKs、 Amazon Command Line Interface (Amazon CLI) 或。 Amazon Tools for PowerShell无法在 Amazon KMS 控制台中执行加密操作。有关使用多种编程语言调用加密操作的示例，请参阅使用的代码示 Amazon KMS 例 Amazon SDKs。

下表列出了 Amazon KMS 加密操作。它还显示操作中使用的 KMS 密钥的密钥类型和密钥使用要求。

操作	密钥类型	密钥用法
Decrypt	对称或非对称	ENCRYPT_DECRYPT
DeriveSharedSecret	非对称	KEY_AGREEMENT
Encrypt	对称或非对称	ENCRYPT_DECRYPT
GenerateDataKey	对称	ENCRYPT_DECRYPT
GenerateDataKeyPair	对称 [1] 在自定义密钥存储中的 KMS 密钥中不受支持。	ENCRYPT_DECRYPT
GenerateDataKeyPairWithoutPlaintext	对称 [1] 在自定义密钥存储中的 KMS 密钥中不受支持。	ENCRYPT_DECRYPT
GenerateDataKeyWithoutPlaintext	对称	ENCRYPT_DECRYPT
GenerateMac	HMAC	GENERATE_VERIFY_MAC
GenerateRandom	不适用。此操作不使用 KMS 密钥。	不适用
ReEncrypt	对称或非对称	ENCRYPT_DECRYPT
Sign	非对称	SIGN_VERIFY
Verify	非对称	SIGN_VERIFY
VerifyMac	HMAC	GENERATE_VERIFY_MAC

[1] 生成受对称加密 KMS 密钥保护的非对称数据密钥对。

有关加密操作的权限的信息，请参阅 the section called “权限参考”。

为了让所有用户都能 Amazon KMS 快速响应且功能强大， Amazon KMS 请为每秒调用的加密操作数量设定配额。有关详细信息，请参阅the section called “加密操作的共享配额”。

支持的加密算法

加密算法

以下表格总结了 Amazon 部署到各种服务的密码算法、密码、模式和密钥大小，以用于保护您的数据。不应将其视为 Amazon 中所有可用加密选项的详尽列表。算法可分为两类：

- 首选算法，符合 Amazon 安全和性能标准。
- 可接受算法，可用来满足某些应用程序的兼容性需要，但不是首选。

非对称密码术

下表列出了支持的加密、密钥协议和数字签名用非对称算法。

类型	算法	状态
加密	RSA-OAEP（2048 或 3072 位模数）	可接受
加密	HPKE（P-256 或 P-384、HKDF 和 AES-GCM）	可接受
密钥协议	ML-KEM-768 或 ML-KEM-1024	首选（抗量子）
密钥协议	ECDH(E) 与 P-384 结合	可接受
密钥协议	ECDH(E) 与 P-256、P-521 或 X25519 结合	可接受
密钥协议	ECDH(E) 与 brainpool P256r1、brainpoolP384r1 或 brainpoolP512r1 结合	可接受

类型	算法	状态
Signatures	ML-DSA-65 或 ML-DSA-87	首选（抗量子）
Signatures	SLH-DSA	首选（抗量子软件/固件签名）
Signatures	ECDSA 与 P-384 结合	可接受
Signatures	ECDSA 与 P-256、P-521 或 Ed25519 结合	可接受
Signatures	RSA-2048 或 RSA-3072	可接受

非对称加密

下表列出了支持的加密、经身份验证加密和密钥封装用对称算法。

类型	算法	状态
经过身份验证的加密	AES-GCM-256	首选
经过身份验证的加密	AES-GCM-128	可接受
经过身份验证的加密	ChaCha20/Poly1305	可接受
加密模式	AES-XTS-256（用于块存储）	首选
加密模式	AES-CBC/CTR（未经身份验证模式）	可接受
密钥包装	AES-GCM-256	首选
密钥包装	AES-KW 或 AES-KWP 与 256 位密钥结合	可接受

加密函数

下表列出了支持的哈希化、密钥派生、消息身份验证和密码哈希处理用算法。

类型	算法	状态
哈希	SHA2-384	首选
哈希	SHA2-256	可接受
哈希	SHA3	可接受
密钥派生	HKDF_Expand 或 HKDF 与 SHA2-256 结合	首选
密钥派生	计数器模式 KDF 与 HMAC-SHA2-256 结合	可接受
消息身份验证码	HMAC-SHA2-384	首选
消息身份验证码	HMAC-SHA2-256	可接受
消息身份验证码	KMAC	可接受
密码哈希处理	scrypt 与 SHA384 结合	首选
密码哈希处理	PBKDF2	可接受

要详细了解 Amazon 中部署的密码算法，请参阅 Cryptography algorithms and Amazon Web Services 服务。

KMS 密钥访问权限和权限

要使用 Amazon KMS，您必须拥有 Amazon 可用于对请求进行身份验证的凭证。证书必须包括访问 Amazon 资源的权限 Amazon KMS keys 和别名。除非明确提供了 KMS 密钥的权限，否则任何 Amazon 委托人均无权访问 KMS 密钥，并且从不被拒绝。不存在使用或管理 KMS 密钥的隐式权限或自动权限。

要控制对 KMS 密钥的访问，您可以使用下列策略机制。

- 密钥策略：每个 KMS 密钥都有密钥策略。密钥策略也是控制访问 KMS 密钥的主要机制。您只能使用密钥策略来控制访问，这意味着对 KMS 密钥的所有访问均在单个文档（密钥策略）中进行定义。有关使用密钥策略的更多信息，请参阅密钥政策。
- IAM policy ：您可以将 IAM policy 与密钥策略和授权结合使用，以控制对 KMS 密钥的访问。通过用这种方式控制访问，您可以管理 IAM 中各 IAM 身份的所有权限。若要使用 IAM policy 允许访问 KMS 密钥，密钥策略必须明确允许此访问。有关使用 IAM; 策略的更多信息，请参阅 IAM 策略。
- 授权：您可以将密钥策略与 IAM policy 结合使用，以允许对 KMS 密钥的访问。通过用这种方式控制访问权限，您可以在密钥策略中允许访问 KMS 密钥，并允许有关身份将其访问权限委托给其他身份。有关使用授权的更多信息，请参阅Amazon KMS 中的授权。

KMS 密钥政策

管理 Amazon KMS 资源访问权限的主要方法是使用策略。策略是用于描述哪些委托人可以访问什么资源的文档。附加到 IAM 身份的策略称为基于身份的策略（或 IAM 策略），附加到其他类型资源的策略称为资源策略。 Amazon KMS KMS 密钥的资源策略称为密钥策略。

所有 KMS 密钥都具有密钥策略。如果您不提供一个，请为您 Amazon KMS 创建一个。 Amazon KMS 使用的默认密钥策略会有所不同，具体取决于您是在 Amazon KMS 控制台中创建密钥还是使用 Amazon KMS API。我们建议您编辑默认密钥政策，使其符合贵组织对最低权限的要求。

如果密钥和 IAM 委托人位于同一个 Amazon 账户中，则可以单独使用密钥策略来控制访问权限，这意味着对 KMS 密钥的全部访问范围是在单个文档（密钥策略）中定义的。但是，当一个账户中的调用者必须访问另一个账户中的密钥时，您不能仅使用密钥政策来授予访问权限。在跨账户场景中，必须将 IAM 策略附加到调用者的用户或角色上，明确允许调用者进行 API 调用。

您也可以将 IAM 策略与密钥政策和授权结合使用，以控制对 KMS 密钥的访问权限。要使用 IAM 策略控制对 KMS 密钥的访问权限，密钥政策必须授予账户使用 IAM 策略的权限。您可以指定启用 IAM 策略的密钥政策语句，也可以在密钥政策中明确指定允许的主体。

在制定策略时，请确保有严格的控制措施，限制可以执行以下操作的人员：

- 更新、创建和删除 IAM 和 KMS 密钥政策
- 对用户、角色和组附加和分离 IAM 策略
- 对您的 KMS 密钥附加和分离 KMS 密钥政策

KMS 密钥授权

除了 IAM 和密钥策略外，还 Amazon KMS 支持授权。授权提供了一种灵活而强大的权限委托方式。您可以使用授权向您的 Amazon 账户或其他 Amazon 账户中的 IAM 委托人发放有时限的 KMS 密钥访问权限。如果您在创建策略时不知道主体的名称，或者需要访问权限的主体经常发生变化，我们建议您授予有时限的访问权限。被授权者主体可以与 KMS 密钥位于同一账户中，也可以位于不同的账户中。如果主体和 KMS 密钥位于不同的账户中，则除了授予之外，您还必须指定 IAM 策略。授权需要额外的管理，因为您必须调用 API 来创建授权，并在不再需要时停用或撤销授权。

中的关键政策 Amazon KMS

密钥策略是的资源策略 Amazon KMS key。密钥政策是控制对 KMS 密钥访问的主要方法。每个 KMS 密钥必须只有一个密钥策略。密钥策略中的语句确定谁有权限使用 KMS 密钥以及如何使用 KMS 密钥。您还可使用 IAM policy 和授权来控制对 KMS 密钥的访问，但每个 KMS 密钥必须有一个密钥策略。

任何 Amazon 委托人（包括账户根用户或密钥创建者）都无权访问 KMS 密钥，除非在密钥策略、IAM 策略或授权中明确允许且从不被拒绝。

除非密钥策略明确允许，否则您不能使用 IAM policy 允许访问 KMS 密钥。未经密钥策略许可，允许权限的 IAM policy 无效。（您可以使用 IAM policy 来拒绝在未经密钥策略许可的情况下对 KMS 密钥的访问权限。）默认密钥策略启用 IAM policy。若要在密钥策略中启用 IAM policy，请添加 允许访问 Amazon Web Services 账户 并启用 IAM policy 中所述的策略语句。

与全局性的 IAM policy 不同，密钥策略是区域性策略。密钥策略仅控制对同一区域中 KMS 密钥的访问。该策略对其他地区的 KMS 密钥无效。

主题

- 创建密钥策略
- 默认密钥策略
- 查看密钥政策

- 更改密钥策略
- 关键策略中的 Amazon 服务权限

创建密钥策略

您可以在 Amazon KMS 控制台中创建和管理密钥策略，也可以使用 Amazon KMS API 操作（例如CreateKeyReplicateKey、和）PutKeyPolicy。

在 Amazon KMS 控制台中创建 KMS 密钥时，控制台将引导您完成基于控制台默认密钥策略创建密钥策略的步骤。使用CreateKey或时 ReplicateKey APIs，如果您未指定密钥策略，则这些策略将对以编程方式创建的密钥 APIs 应用默认密钥策略。使用 PutKeyPolicy API 时，需要指定密钥政策。

每个策略文档都可以有一个或多个策略语句。以下示例显示了具有一个策略语句的有效密钥策略文档。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "DescribePolicyStatement",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:user/Alice"
      },
      "Action": "kms:DescribeKey",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:KeySpec": "SYMMETRIC_DEFAULT"
        }
      }
    }
  ]
}
```

主题

- 密钥策略格式
- 密钥策略中的元素

- 示例 密钥策略

密钥策略格式

密钥策略文档必须符合以下规则：

- 最多 32KB（32768 字节）
- 密钥策略语句中的 Sid 元素可以包含空格。（IAM policy 文档的 Sid 元素中禁止使用空格。）

关键策略文档只能包含以下字符：

- 可打印的 ASCII 字符
- Basic Latin 和 Latin-1 Supplement 字符集中的可打印字符
- 制表符（\u0009）、换行符（\u000A）和回车（\u000D）特殊字符

密钥策略中的元素

密钥策略文档必须有以下元素：

版本

指定密钥策略文档版本。将版本设置为 2012-10-17（最新版本）。

语句

随附策略语句。密钥策略文档必须至少包含一个语句。

每个密钥策略语句最多包含六个元素。Effect、Principal、Action 和 Resource 是必需元素。

Sid

（可选）语句标识符（Sid），是可用于描述语句的任意字符串。密钥策略中的 Sid 可以包含空格。（您不能在 IAM policy Sid 元素中包含空格。）

效果

（必需）确定是允许还是拒绝该策略语句中的权限。有效值为 Allow 或 Deny。如果您没有显式允许对 KMS 密钥的访问，则隐式拒绝访问。您也可显式拒绝对 KMS 密钥的访问。您可以通过这种方式确保用户无法访问 CMK，即使其他策略允许访问也是如此。

主体

（必需）主题是获取策略语句中指定的权限的身份。您可以在密钥策略中将 IAM 用户、IAM 角色和某些 Amazon 服务指定 Amazon Web Services 账户为委托人。IAM 用户组 在任何策略类型中都不是有效主体。

星号值，例如，`"AWS": "*"` 表示所有账户中的所有 Amazon 身份。

> **Important**
>
> 除非您使用条件限制密钥政策，否则不要在允许权限的任何密钥政策语句将主体设置为星号（*）。星号赋予每个身份使用 KMS 密钥的 Amazon Web Services 账户 权限，除非其他策略声明明确拒绝。其他用户只要在自己的账户中拥有相应权限，就 Amazon Web Services 账户 可以使用您的 KMS 密钥。

> **Note**
>
> IAM 最佳实践不鼓励使用具有长期凭证的 IAM 用户。而应尽可能使用提供临时凭证的 IAM 角色。有关更多信息，请参阅《IAM 用户指南》中的 IAM 安全最佳实践。

当密钥策略语句中的主体是表示为 `arn:aws:iam::111122223333:root"` 的 Amazon Web Services 账户 主体时，该策略语句不授予任何 IAM 主体权限。相反，它 Amazon Web Services 账户 允许使用 IAM 策略委派密钥策略中指定的权限。[尽管在账户标识符中使用了“root”，但是格式为 `arn:aws:iam::111122223333:root"` 的主体并不代表 Amazon 账户根用户。但是，账户主体代表账户及其管理员（包括账户根用户）。]

当委托人是另一个 Amazon Web Services 账户 或其委托人时，只有在使用 KMS 密钥和密钥策略的区域中启用该账户时，权限才会生效。有关默认情况下未启用的区域（“选择加入区域”）的信息，请参阅《Amazon Web Services 一般参考》 中的 Managing Amazon Web Services 区域。

要允许其他用户 Amazon Web Services 账户 或其委托人使用 KMS 密钥，您必须在密钥策略和另一个账户的 IAM 策略中提供权限。有关更多信息，请参阅 允许其他账户中的用户使用 KMS 密钥。

操作

（必需）指定要允许或拒绝的 API 操作。例如，该kms:Encrypt操作对应于 “ Amazon KMS 加密” 操作。您可以在策略语句中列出多个 Action。有关更多信息，请参阅 权限参考。

> **Note**
>
> 如果密钥策略语句中缺少必要的 Action 元素，则策略语句将无效。没有 Action 元素的密钥策略语句不适用于任何 KMS 密钥。
> 当关键策略语句缺少其Action元素时， Amazon KMS 控制台会正确报告错误，但是即使策略语句无效，CreateKey和也会PutKeyPolicy APIs 成功。

资源

（必需）在密钥策略中，资源元素的值为 "*"，这意味着“本 KMS 密钥”。星号 ("*") 标识密钥策略附加到的 KMS 密钥。

> **Note**
>
> 如果密钥策略语句中缺少必要的 Resource 元素，则策略语句将无效。没有 Resource 元素的密钥策略语句不适用于任何 KMS 密钥。
> 当关键策略语句缺少其Resource元素时， Amazon KMS 控制台会正确报告错误，但是即使策略语句无效，CreateKey和也会PutKeyPolicy APIs 成功。

条件

（可选）条件指定要使密钥策略生效而必须满足的要求。使用条件， Amazon 可以评估 API 请求的上下文以确定政策声明是否适用。

要指定条件，您可以使用预定义的条件键。 Amazon KMS 支持Amazon 全局条件键和Amazon KMS 条件键。为了支持基于属性的访问控制 (ABAC)， Amazon KMS 提供了基于标签和别名控制对 KMS 密钥的访问的条件密钥。有关更多信息，请参阅 ABAC for Amazon KMS。

条件的格式为：

```
"Condition": {"condition operator": {"condition key": "condition value"}}
```

例如：

```
"Condition": {"StringEquals": {"kms:CallerAccount": "111122223333"}}
```

有关 Amazon 策略语法的更多信息，请参阅 Amazon IAM 用户指南中的 IAM 策略参考。

示例 密钥策略

以下示例显示了对称加密 KMS 密钥的完整密钥策略。在阅读本章中的密钥策略概念时，您可以将它作为参考。此密钥策略将之前默认密钥策略部分的示例策略语句合并为一个密钥策略，该密钥策略可完成以下操作：

- 允许示例 Amazon Web Services 账户 111122223333 对 KMS 密钥的完全访问权限。它其允许账户及其管理员（包括账户根用户在紧急情况下）在该账户中使用 IAM policy 来允许访问 KMS 密钥。
- 允许 `ExampleAdminRole` IAM 角色管理 KMS 密钥。
- 允许 `ExampleUserRole` IAM 角色使用 KMS 密钥。

JSON

```
{
    "Id": "key-consolepolicy",
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "EnableIAMUserPermissions",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "AllowKeyAdministratorsAccess",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/ExampleAdminRole"
            },
            "Action": [
                "kms:Create*",
                "kms:Describe*",
```

```
                "kms:Enable*",
                "kms:List*",
                "kms:Put*",
                "kms:Update*",
                "kms:Revoke*",
                "kms:Disable*",
                "kms:Get*",
                "kms:Delete*",
                "kms:TagResource",
                "kms:UntagResource",
                "kms:ScheduleKeyDeletion",
                "kms:CancelKeyDeletion",
                "kms:RotateKeyOnDemand"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowKeyUse",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/ExampleUserRole"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        },
        {
            "Sid": "AllowAttachmentPersistentResources",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:role/ExampleUserRole"
            },
            "Action": [
                "kms:CreateGrant",
                "kms:ListGrants",
                "kms:RevokeGrant"
            ],
            "Resource": "*",
            "Condition": {
```

```
                "Bool": {
                    "kms:GrantIsForAWSResource": "true"
                }
            }
        }
    ]
}
```

默认密钥策略

创建 KMS 密钥时，您可以为新的 KMS 密钥指定密钥策略。如果您不提供一个，请为您 Amazon KMS 创建一个。 Amazon KMS 使用的默认密钥策略会有所不同，具体取决于您是在 Amazon KMS 控制台中创建密钥还是使用 Amazon KMS API。

以编程方式创建 KMS 密钥时采用的默认密钥策略

当您使用 Amazon KMS API 以编程方式创建 KMS 密钥（包括使用Amazon SDKs、Amazon Command Line Interface或 Amazon Tools for PowerShell），并且未指定密钥策略时，会 Amazon KMS 应用非常简单的默认密钥策略。此默认密钥策略有一个策略声明，该声明向拥有 KMS 密钥的用户授予使用 IAM 策略的权限，以允许对 KMS 密钥进行所有 Amazon KMS 操作。 Amazon Web Services 账户 有关此策略语句的更多信息，请参阅允许访问 Amazon Web Services 账户 并启用 IAM policy。

使用创建 KMS 密钥时的默认密钥策略 Amazon Web Services 管理控制台

使用创建 KMS 密钥时 Amazon Web Services 管理控制台，密钥策略以允许访问 Amazon Web Services 账户 并启用 IAM 策略的策略声明开头。然后，控制台会添加密钥管理员声明、密钥用户声明以及（对于大多数密钥类型）允许委托人将 KMS 密钥用于其他 Amazon 服务的声明。您可以使用 Amazon KMS 控制台的功能来指定 IAM 用户、IAM 角色、 Amazon Web Services 账户 谁是密钥管理员以及谁是关键用户（或两者兼而有之）。

权限

- 允许访问 Amazon Web Services 账户 并启用 IAM policy
- 允许密钥管理员管理 KMS 密钥
- 允许密钥用户使用 KMS 密钥
  - 允许密钥用户使用 KMS 密钥进行加密操作
  - 允许密钥用户将 KMS 密钥与 Amazon 服务一起使用

允许访问 Amazon Web Services 账户 并启用 IAM policy

以下默认密钥策略语句至关重要。

- 它为拥有 KMS 密钥的人提供对 KMS 密钥的完全访问权限。 Amazon Web Services 账户

  与其他 Amazon 资源策略不同，Amazon KMS 密钥策略不会自动向账户或其任何身份授予权限。若要授予账户管理员权限，密钥策略必须包含提供此权限的显式语句，如下所示。
- 除密钥策略外，还允许账户使用 IAM policy 允许对 KMS 密钥进行访问。

  如果没有此权限，尽管拒绝访问密钥的 IAM policy 仍然有效，但是允许访问密钥的 IAM policy 将无效。
- 此权限通过向账户管理员（包括账户根用户）授予无法删除的访问控制权限，来降低密钥变得无法管理的风险。

以下密钥策略语句是以编程方式创建的 KMS 密钥的完整默认密钥策略。这是 Amazon KMS 控制台中创建的 KMS 密钥的默认密钥策略中的第一条策略声明。

```
{
  "Sid": "Enable IAM User Permissions",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:root"
   },
  "Action": "kms:*",
  "Resource": "*"
}
```

允许 IAM policy 允许对 KMS 密钥的访问。

上面显示的密钥策略声明向拥有密钥的用户授予使用 IAM 策略和密钥策略的权限，以允许对 KMS 密钥执行所有操作 (`kms:*`)。 Amazon Web Services 账户

本密钥策略声明中的主体是账户主体，其由此格式 (`arn:aws:iam::account-id:root`) 的 ARN 进行表示。账户委托人代表 Amazon 账户及其管理员。

当密钥策略语句中的主体是账户主体时，该策略语句不会向任何 IAM 主体授予使用 KMS 密钥的权限。相反，其授予账户使用 IAM policy 委托密钥语句中指定的权限。此默认密钥策略语句允许账户使用 IAM policy 委托 KMS 密钥上所有操作 (`kms:*`) 的权限。

可以降低 KMS 密钥变得不可管理的风险。

与其他 Amazon 资源策略不同， Amazon KMS 密钥策略不会自动向账户或其任何委托人授予权限。若要授予任何主体（包括账户主体）权限，您必须使用明确提供权限的密钥策略语句。您无需授予账户主体或任何主体访问 KMS 密钥的权限。但是，授予账户主体访问权限有助于防止无法管理密钥。

例如，假设您创建的密钥策略仅授予一个用户访问 KMS 密钥的权限。如果删除该用户，则密钥将变得无法管理，您必须联系 Amazon Support 才能重新获得 KMS 密钥的访问权限。

上面显示的密钥策略语句将向账户主体授予控制密钥的权限。账户委托人代表 Amazon Web Services 账户 及其管理员，包括账户 root 用户。除非您删除 Amazon Web Services 账户，否则账户根用户是唯一无法删除的主体。IAM 最佳实践不鼓励代表账户根用户采取操作，但紧急情况除外。但是，如果删除所有其他具有 KMS 密钥访问权限的用户和角色，则您可能需要充当账户根用户。

允许密钥管理员管理 KMS 密钥

控制台创建的默认密钥策略允许您选择账户中的 IAM 用户和角色，并使其成为密钥管理员。此语句称为密钥管理员语句。密钥管理员有权管理 KMS 密钥，但无权在加密操作中使用 KMS 密钥。在原定设置视图或策略视图中创建 KMS 密钥时，您可以将 IAM 用户和角色添加到密钥管理员列表。

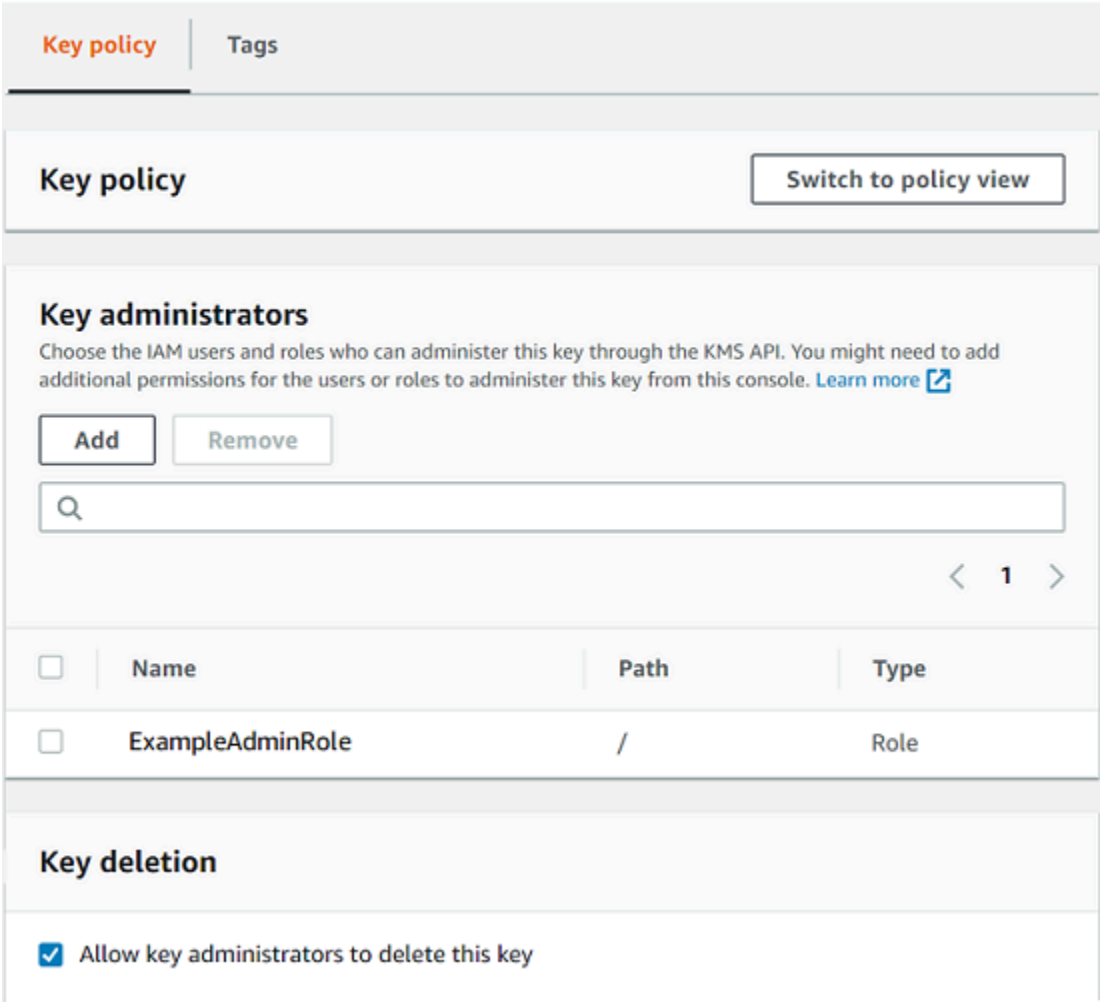
Warning

由于密钥管理员有权更改密钥策略和创建授权，因此他们可以向自己和其他人 Amazon KMS 授予此策略中未指定的权限。
有权管理标签和别名的委托人也可以控制对 KMS 密钥的访问。有关更多信息，请参阅 ABAC for Amazon KMS。

Note

IAM 最佳实践不鼓励使用具有长期凭证的 IAM 用户。而应尽可能使用提供临时凭证的 IAM 角色。有关更多信息，请参阅《IAM 用户指南》中的 IAM 安全最佳实践。

以下示例在 Amazon KMS 控制台的原定设置视图中显示了密钥管理员语句。



以下示例是 Amazon KMS 控制台的策略视图中的密钥管理员语句。此密钥管理员语句适用于单区域对称加密 KMS 密钥。

> **Note**
>
> Amazon KMS 控制台在语句标识符下将密钥管理员添加到密钥策略中"Allow access for Key Administrators"。如果修改此语句标识符，则可能会影响控制台显示您对该语句所做修改的方式。

```
{
  "Sid": "Allow access for Key Administrators",
  "Effect": "Allow",
  "Principal": {"AWS":"arn:aws:iam::111122223333:role/ExampleAdminRole"},
  "Action": [
    "kms:Create*",
```

```
    "kms:Describe*",
    "kms:Enable*",
    "kms:List*",
    "kms:Put*",
    "kms:Update*",
    "kms:Revoke*",
    "kms:Disable*",
    "kms:Get*",
    "kms:Delete*",
    "kms:TagResource",
    "kms:UntagResource",
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion",
    "kms:RotateKeyOnDemand"
  ],
  "Resource": "*"
}
```

最常见 KMS 密钥、单区域对称加密 KMS 密钥的原定设置密钥管理员语句允许以下权限。有关每个权限的详细信息，请参阅 Amazon KMS 权限。

当您使用 Amazon KMS 控制台创建 KMS 密钥时，控制台会将您指定的用户和角色添加到密钥管理员语句中的`Principal`元素中。

这些权限中有很多都包含通配符 (*)，使用它可以允许以指定动词开头的所有权限。因此，在 Amazon KMS 添加新的 API 操作时，自动允许密钥管理员使用它们。您不必更新密钥策略即可包含新操作。如果您希望将密钥管理员限制在一组固定的 API 操作中，则可以更改密钥策略。

`kms:Create*`

允许 `kms:CreateAlias` 和 `kms:CreateGrant`。（`kms:CreateKey` 权限仅在 IAM policy 中有效。）

`kms:Describe*`

允许 `kms:DescribeKey`。需要 `kms:DescribeKey` 权限才能查看 Amazon Web Services 管理控制台中 KMS 密钥的密钥详细信息页面。

`kms:Enable*`

允许 `kms:EnableKey`。对于对称加密 KMS 密钥，它还允许 `kms:EnableKeyRotation`。

kms:List*

允许 kms:ListGrants、kms:ListKeyPolicies 和 kms:ListResourceTags。（查看 Amazon Web Services 管理控制台中的 KMS 密钥所需的 kms:ListAliases 和 kms:ListKeys 权限仅在 IAM policy 中有效。）

kms:Put*

允许 kms:PutKeyPolicy。此权限允许密钥管理员更改此 KMS 密钥的密钥策略。

kms:Update*

允许 kms:UpdateAlias 和 kms:UpdateKeyDescription。对于多区域密钥，它允许此 KMS 密钥上的 kms:UpdatePrimaryRegion。

kms:Revoke*

允许 kms:RevokeGrant，即允许密钥管理员删除授权，即使管理员不是授权中的停用主体。

kms:Disable*

允许 kms:DisableKey。对于对称加密 KMS 密钥，它还允许 kms:DisableKeyRotation。

kms:Get*

允许 kms:GetKeyPolicy 和 kms:GetKeyRotationStatus。对于具有导入密钥材料的 KMS 密钥，它允许 kms:GetParametersForImport。对于非对称 KMS 密钥，它允许 kms:GetPublicKey。需要 kms:GetKeyPolicy 权限才能查看 Amazon Web Services 管理控制台中 KMS 密钥的密钥策略。

kms:Delete*

允许 kms:DeleteAlias。对于具有导入密钥材料的密钥，它允许 kms:DeleteImportedKeyMaterial。kms:Delete* 权限不允许密钥管理员删除 KMS 密钥 (ScheduleKeyDeletion)。

kms:TagResource

允许 kms:TagResource，以此允许密钥管理员向 KMS 密钥添加标签。由于标签也可用于控制对 KMS 密钥的访问，因此管理员通过此权限可允许或拒绝对 KMS 密钥的访问。有关更多信息，请参阅 ABAC for Amazon KMS。

kms:UntagResource

允许 kms:UntagResource，以此允许密钥管理员从 KMS 密钥删除标签。由于标签可用于控制对密钥的访问，因此管理员通过此权限可允许或拒绝对 KMS 密钥的访问。有关更多信息，请参阅 ABAC for Amazon KMS。

kms:ScheduleKeyDeletion

允许 kms:ScheduleKeyDeletion，以此允许密钥管理员删除此 KMS 密钥。要删除此权限，请清除 Allow key administrators to delete this key（允许密钥管理员删除此密钥）选项。

kms:CancelKeyDeletion

允许 kms:CancelKeyDeletion，以此允许密钥管理员取消此 KMS 密钥的删除。要删除此权限，请清除 Allow key administrators to delete this key（允许密钥管理员删除此密钥）选项。

kms:RotateKeyOnDemand

允许 kms:RotateKeyOnDemand，此操作将允许密钥管理员按需轮换此 KMS 密钥中的密钥材料。

Amazon KMS 在创建特殊用途密钥时，将以下权限添加到默认密钥管理员语句中。

kms:ImportKeyMaterial

kms:ImportKeyMaterial 权限允许密钥管理员将密钥材料导入 KMS 密钥。仅当创建不含密钥材料的 KMS 密钥时，此权限才包含在密钥策略中。

kms:ReplicateKey

该kms:ReplicateKey权限允许密钥管理员在不同 Amazon 区域创建多区域主密钥的副本。仅当您创建多区域主键或副本键时，此权限才会包含在密钥策略中。

kms:UpdatePrimaryRegion

kms:UpdatePrimaryRegion 权限允许密钥管理员将多区域副本密钥更改为多区域主键。仅当您创建多区域主键或副本键时，此权限才会包含在密钥策略中。

允许密钥用户使用 KMS 密钥

控制台为 KMS 密钥创建的默认密钥策略允许您在账户中选择 IAM 用户和 IAM 角色以及外部角色 Amazon Web Services 账户，并使其成为密钥用户。

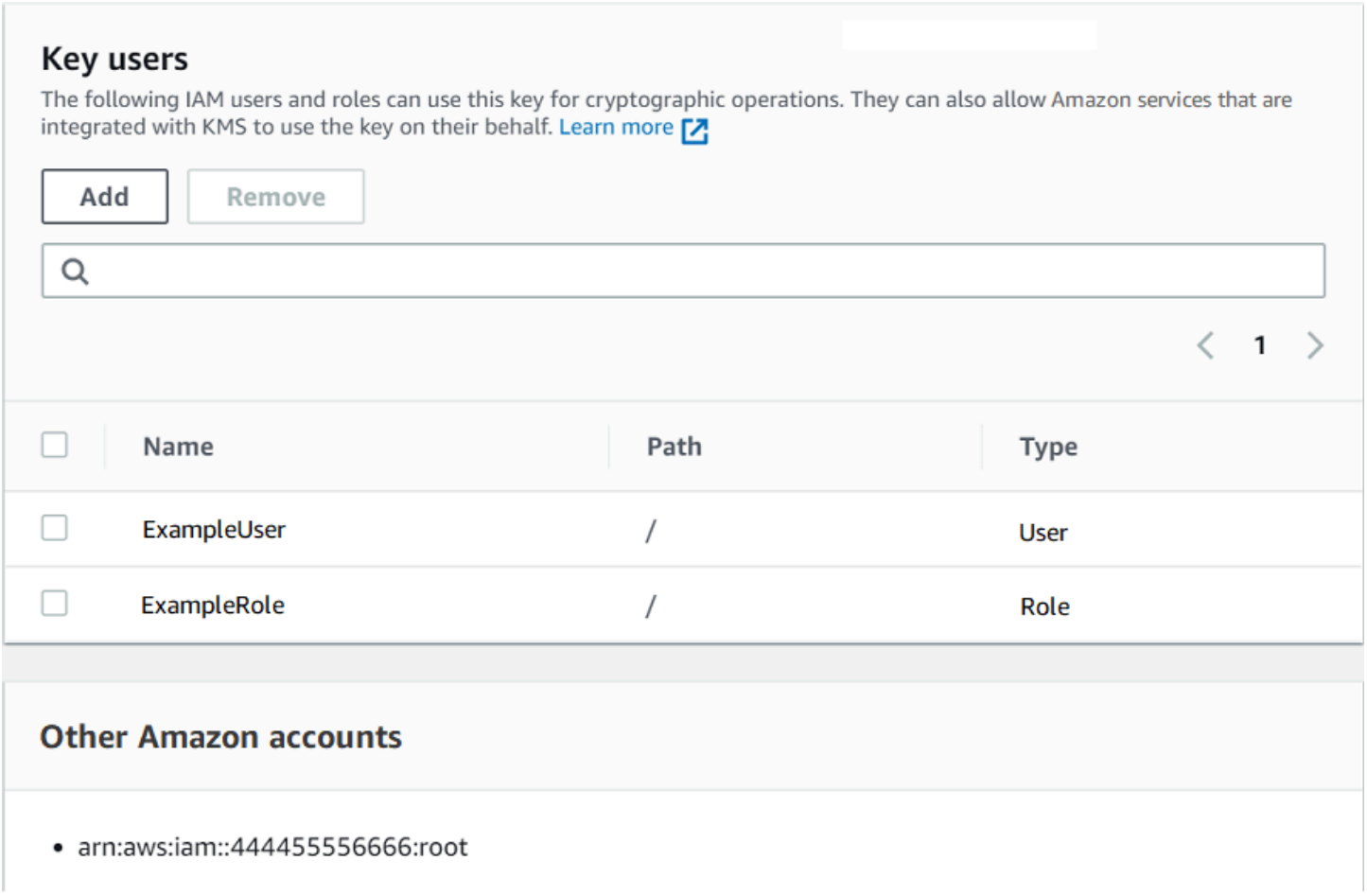
控制台将两个策略语句添加到密钥用户的密钥策略中。

- 直接使用 KMS 密钥 — 第一个密钥策略语句授予密钥用户将 KMS 密钥直接用于该类型 KMS 密钥支持的所有加密操作的权限。

- 将 KMS 密钥用于 Amazon 服务 — 第二项策略声明允许密钥用户允许与其集成的 Amazon 服务代表他们使用 KMS 密钥 Amazon KMS 来保护资源，例如 Amazon S3 存储桶和 Amazon DynamoDB 表。

创建 KMS 密钥时，您可以将 IAM 用户、IAM 角色和其他 Amazon Web Services 账户 角色添加到密钥用户列表中。您也可以使用控制台的默认密钥策略视图来编辑该列表，如下图所示。默认密钥策略视图可从密钥详细信息页面获取。有关允许其他 Amazon Web Services 账户 用户使用 KMS 密钥的更多信息，请参阅允许其他账户中的用户使用 KMS 密钥。

> **Note**
>
> IAM 最佳实践不鼓励使用具有长期凭证的 IAM 用户。而应尽可能使用提供临时凭证的 IAM 角色。有关更多信息，请参阅《IAM 用户指南》中的 IAM 安全最佳实践。



单区域对称的原定设置密钥用户语句允许以下权限。有关每个权限的详细信息，请参阅 Amazon KMS 权限。

当您使用 Amazon KMS 控制台创建 KMS 密钥时，控制台会将您指定的用户和角色添加到每个密钥用户语句中的`Principal`元素中。

> **Note**
>
> Amazon KMS 控制台在语句标识符`"Allow use of the key"`和下将密钥用户添加到密钥策略中`"Allow attachment of persistent resources"`。如果修改这些语句标识符，则可能会影响控制台显示您对该语句所做修改的方式。

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {"AWS": [
    "arn:aws:iam::111122223333:role/ExampleRole",
    "arn:aws:iam::444455556666:root"
  ]},
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*"
},
{
  "Sid": "Allow attachment of persistent resources",
  "Effect": "Allow",
  "Principal": {"AWS": [
    "arn:aws:iam::111122223333:role/ExampleRole",
    "arn:aws:iam::444455556666:root"
  ]},
  "Action": [
    "kms:CreateGrant",
    "kms:ListGrants",
    "kms:RevokeGrant"
  ],
  "Resource": "*",
```

```
  "Condition": {"Bool": {"kms:GrantIsForAWSResource": true}}
}
```

允许密钥用户使用 KMS 密钥进行加密操作

密钥用户有权在 KMS 密钥支持的所有加密操作中直接使用 KMS 密钥。他们还可以使用该DescribeKey操作在 Amazon KMS 控制台或使用 Amazon KMS API 操作来获取有关 KMS 密钥的详细信息。

默认情况下，Amazon KMS 控制台会将密钥用户语句添加到默认密钥策略中，如下例所示。由于支持的 API 操作不同，策略语句中针对对称加密 KMS 密钥、HMAC KMS 密钥、用于公有密钥加密的非对称 KMS 密钥以及用于签名和验证的非对称 KMS 密钥的操作略有不同。

对称加密 KMS 密钥

控制台将以下语句添加到对称加密 KMS 密钥的密钥策略中。

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"},
  "Action": [
    "kms:Decrypt",
    "kms:DescribeKey",
    "kms:Encrypt",
    "kms:GenerateDataKey*",
    "kms:ReEncrypt*"
  ],
  "Resource": "*"
}
```

HMAC KMS 密钥

控制台将以下语句添加到 HMAC KMS 密钥的密钥策略中。

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"},
  "Action": [
    "kms:DescribeKey",
```

```
    "kms:GenerateMac",
    "kms:VerifyMac"
  ],
  "Resource": "*"
}
```

用于公有密钥加密的非对称 KMS 密钥

对于密钥用法为 Encrypt and decrypt（加密和解密）的非对称 KMS 密钥，控制台将以下语句添加到其密钥策略中。

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:DescribeKey",
    "kms:GetPublicKey"
  ],
  "Resource": "*"
}
```

用于签名和验证的非对称 KMS 密钥

对于密钥用法为 Sign and verify（签名和验证）的非对称 KMS 密钥，控制台将以下语句添加到其密钥策略中。

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"},
  "Action": [
    "kms:DescribeKey",
    "kms:GetPublicKey",
    "kms:Sign",
    "kms:Verify"
  ],
  "Resource": "*"
```

```
}
```

用于派生共享密钥的非对称 KMS 密钥

对于密钥用法为密钥协议的非对称 KMS 密钥，控制台会将以下语句添加到其密钥政策中。

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"},
  "Action": [
    "kms:DescribeKey",
    "kms:GetPublicKey",
    "kms:DeriveSharedSecret"
  ],
  "Resource": "*"
}
```

这些语句中的操作赋予密钥用户以下权限。

kms:Encrypt

允许密钥用户使用此 KMS 密钥加密数据。

kms:Decrypt

允许密钥用户使用此 KMS 密钥解密数据。

kms:DeriveSharedSecret

允许密钥用户使用此 KMS 密钥派生共享密钥。

kms:DescribeKey

允许密钥用户获取有关此 KMS 密钥的详细信息，包括其标识符、创建日期和密钥状态。它还允许密钥用户在 Amazon KMS 控制台中显示有关 KMS 密钥的详细信息。

kms:GenerateDataKey*

允许密钥用户请求对称数据密钥或非对称数据密钥对，以执行客户端加密操作。控制台使用* 通配符表示对以下 API 操作的权限：GenerateDataKey、GenerateDataKeyWithoutPlaintextGenerateDataKeyPair、和GenerateDataKeyPairWithoutPlaintext。这些权限仅对加密数据密钥的对称 KMS 密钥有效。

kms: GenerateMac

允许密钥用户使用 HMAC KMS 密钥生成 HMAC 标签。

kms: GetPublicKey

允许密钥用户下载非对称 KMS 密钥的公有密钥。与您共享此公钥的各方可以对外部的数据进行加密 Amazon KMS。但是，这些密文只能通过调用 Amazon KMS中的 Decrypt 操作进行解密。

kms: ReEncrypt *

允许密钥用户重新加密最初使用此 KMS 密钥加密的数据，或使用此 KMS 密钥重新加密之前已加密的数据。该ReEncrypt操作需要同时访问源 KMS 密钥和目标 KMS 密钥。为此，可以允许对源 KMS 密钥具备 `kms:ReEncryptFrom` 权限，对目标 KMS 密钥具备 `kms:ReEncryptTo` 权限。但是，为简单起见，控制台允许对两个 KMS 密钥具备 `kms:ReEncrypt*` 权限（采用 * 通配符）。

kms:Sign

允许密钥用户使用此 KMS 密钥签署消息。

kms:Verify

允许密钥用户使用此 KMS 密钥验证签名。

kms: VerifyMac

允许密钥用户使用 HMAC KMS 密钥验证 HMAC 标签。

允许密钥用户将 KMS 密钥与 Amazon 服务一起使用

控制台中的默认密钥策略还为密钥用户提供了在使用授权的 Amazon 服务中保护其数据所需的授予权限。 Amazon 服务通常使用授权来获得使用 KMS 密钥的特定和有限权限。

此密钥策略声明允许密钥用户创建、查看和撤消对 KMS 密钥的授权，但前提是授权操作请求来自与 Amazon KMS集成的Amazon 服务。kms: GrantIsFor AWSResource 策略条件不允许用户直接调用这些授权操作。当密钥用户允许时， Amazon 服务可以代表用户创建授权，允许该服务使用 KMS 密钥来保护用户的数据。

密钥用户必须具备这些授权权限，才能一起使用 KMS 密钥和集成服务，但仅有这些权限还不够。密钥用户还必须具备使用集成服务的权限。有关向用户提供与集成的 Amazon 服务的访问权限的详细信息 Amazon KMS，请参阅集成服务的文档。

```
{
  "Sid": "Allow attachment of persistent resources",
```

```
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/ExampleKeyUserRole"},
  "Action": [
    "kms:CreateGrant",
    "kms:ListGrants",
    "kms:RevokeGrant"
  ],
  "Resource": "*",
  "Condition": {"Bool": {"kms:GrantIsForAWSResource": true}}
}
```

例如，密钥用户可以通过以下方式对 KMS 密钥使用这些权限。

- 将此 KMS 密钥与亚马逊弹性区块存储 (Amazon EBS) 和亚马逊弹性计算云 (EC2Amazon) 配合使用，将加密的 EBS 卷附加到实例。 EC2 密钥用户隐式授予 Amazon 使用 KMS 密钥将加密卷附加到实例的 EC2 权限。有关更多信息，请参阅 亚马逊 Elastic Block Store (Amazon EBS) 的使用方式 Amazon KMS。
- 将此 KMS 密钥用于 Amazon Redshift 以启动加密集群。密钥用户向 Amazon Redshift 隐式授予使用 KMS 密钥启动加密集群并创建加密快照的权限。有关更多信息，请参阅Amazon Redshift 如何使用 Amazon KMS。
- 将此 KMS 密钥与其他与 Amazon KMS集成的Amazon 服务一起使用，这些服务使用授权服务创建、管理或使用这些服务加密的资源。

默认密钥策略允许密钥用户向所有使用授权的集成服务委托授权权限。但是，您可以创建自定义密钥策略，将权限限制为指定 Amazon 服务。有关更多信息，请参阅 kms: ViaService 条件键。

查看密钥政策

您可以使用 Amazon KMS 控制台或 Amazon KMS API Amazon 托管式密钥中的GetKeyPolicy操作查看 Amazon KMS 客户托管密钥或账户中的密钥策略。但这些技术不能用于查看其他 Amazon Web Services 账户中的 KMS 密钥的密钥策略。

要了解有关 Amazon KMS 密钥策略的更多信息，请参阅中的关键政策 Amazon KMS。要了解如何确定哪些用户和角色有权访问 KMS 密钥，请参阅 the section called “确定访问权限”。

使用控制 Amazon KMS 台

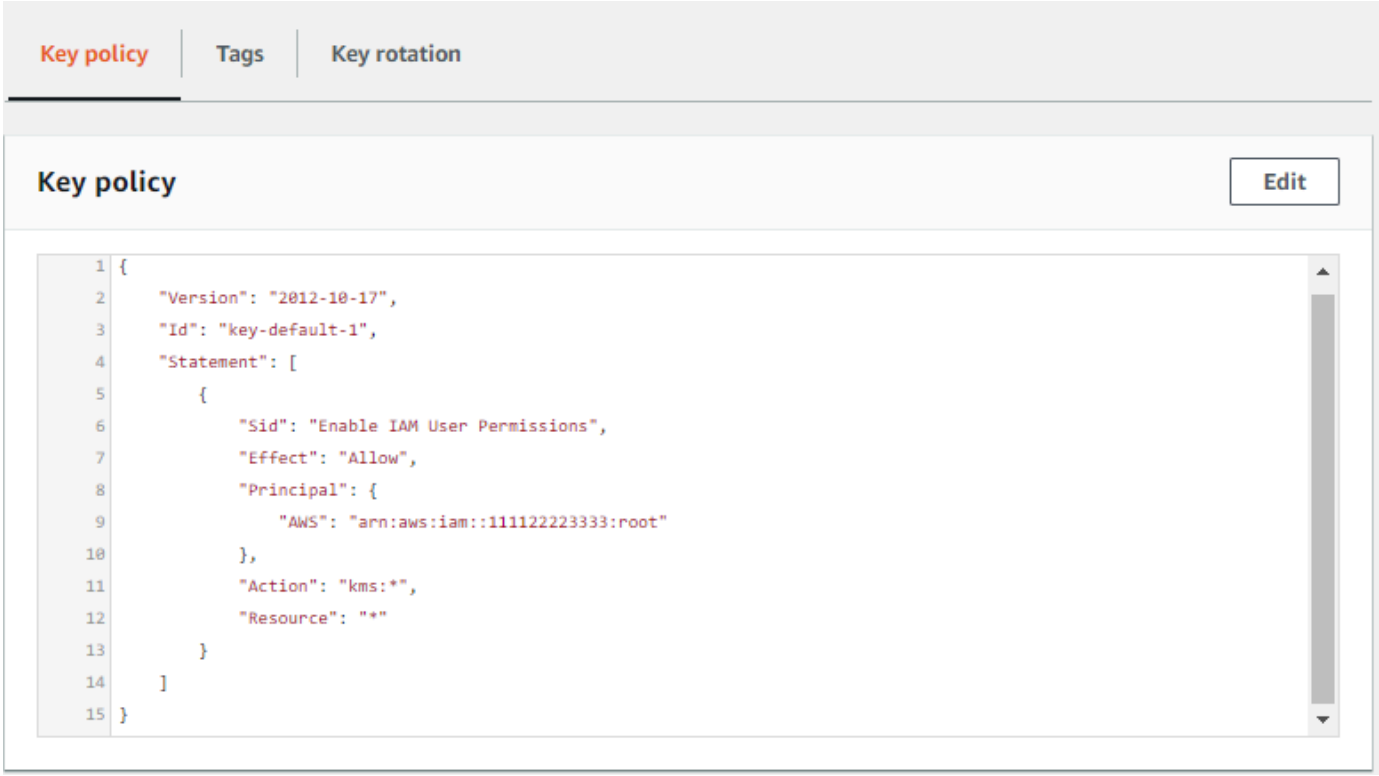
授权用户可以在 Amazon Web Services 管理控制台的 Key policy（密钥策略）选项卡上查看 Amazon 托管式密钥 或客户托管密钥的密钥策略。

要在中查看 KMS 密钥的密钥策略 Amazon Web Services 管理控制台，您必须拥有 k ms: ListAliases、kms: DescribeKey 和 kms: GetKeyPolicy 权限。

1. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/km s 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 要查看您的账户中为您 Amazon 创建和管理的密钥，请在导航窗格中选择Amazon 托管密钥。要查看您账户中自己所创建和管理的密钥，请在导航窗格中选择 Customer managed keys (客户托管密钥)。
4. 在 KMS 密钥列表中，选择要检查的 KMS 密钥的别名或密钥 ID。
5. 选择 Key policy (密钥策略) 选项卡。

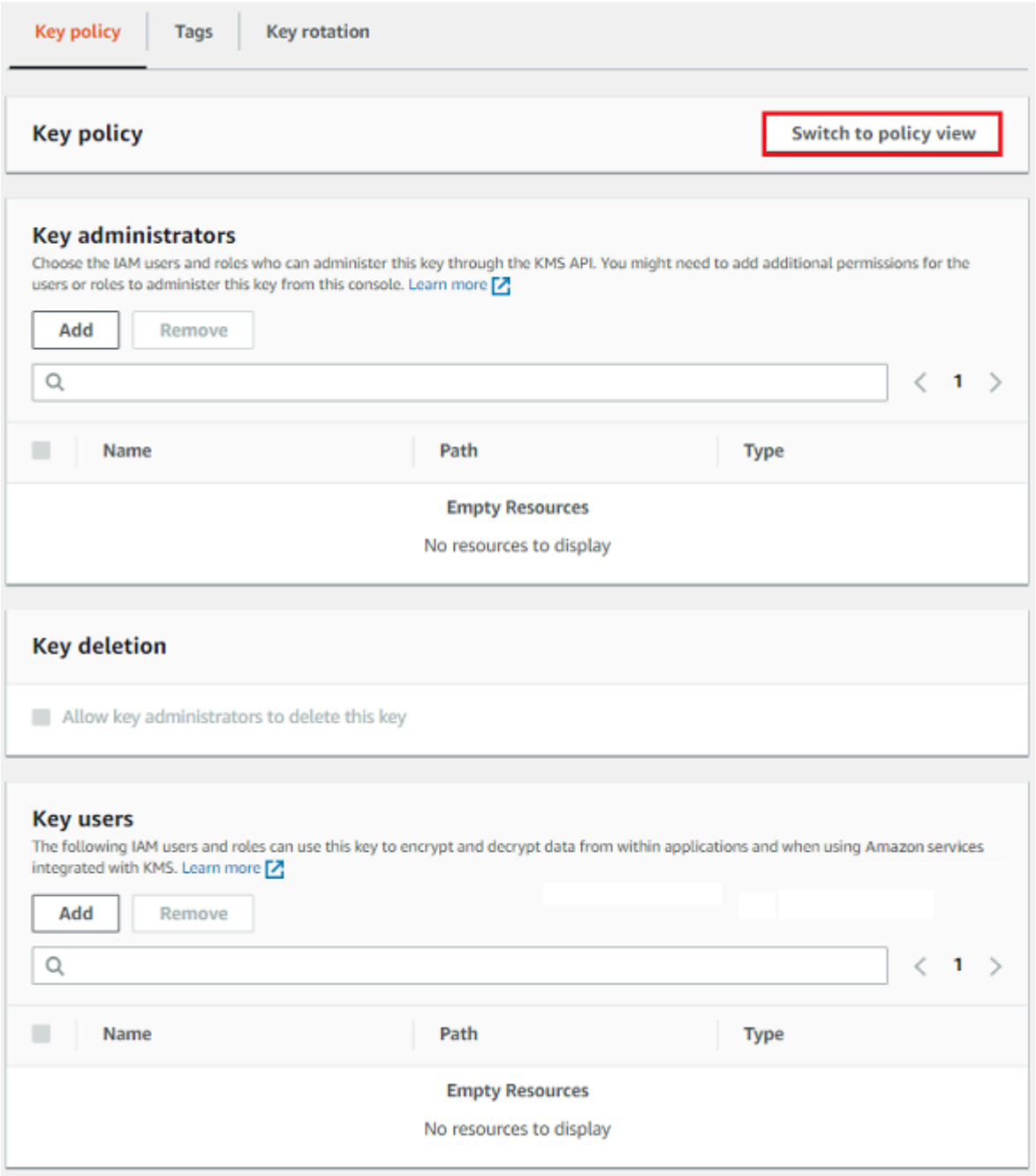
   在 Key policy（密钥策略）选项卡中，您可能会看到密钥策略文档。这是策略视图。在密钥策略语句中，可以看到由密钥策略授予 KMS 密钥访问权限的委托人，还可以看到他们能执行的操作。

   以下示例显示了默认密钥策略的策略视图。



   或者，如果您在中创建了 KMS 密钥 Amazon Web Services 管理控制台，则会看到默认视图，其中包含密钥管理员、密钥删除和密钥用户部分。要查看密钥策略文档，请选择 Switch to policy view (切换到策略视图)。

以下示例显示了默认密钥策略的默认视图。



使用 Amazon KMS API

要在中获取 KMS 密钥的密钥策略 Amazon Web Services 账户，请使用 Amazon KMS API 中的GetKeyPolicy操作。此操作不能用于查看其他帐户中的密钥策略。

以下示例使用 Amazon Command Line Interface (Amazon CLI) 中的get-key-policy命令，但您可以使用任何 Amazon SDK 来发出此请求。

请注意，PolicyName 参数是必需的，即使其唯一的有效值为 default。此外，此命令请求以文本而不是 JSON 形式输出，以便更易于查看。

在运行此命令之前，请将示例密钥 ID 替换为您账户中的有效密钥 ID。

```
$ aws kms get-key-policy --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --policy-name
 default --output text
```

响应应类似于下图，返回默认密钥策略。

JSON

```
{
  "Version":"2012-10-17",
  "Id" : "key-consolepolicy-3",
  "Statement" : [ {
  "Sid" : "EnableIAMUserPermissions",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "arn:aws:iam::111122223333:root"
    },
    "Action" : "kms:*",
    "Resource" : "*"
  } ]
}
```

更改密钥策略

您可以使用 Amazon Web Services 管理控制台 或PutKeyPolicy操作更改中某个 KMS 密钥的密钥策略。 Amazon Web Services 账户 但这些技术不能用于更改其他 Amazon Web Services 账户中的 KMS 密钥的密钥策略。

当更改密钥策略时，请注意以下规则：

- 您可以查看 Amazon 托管式密钥 或客户托管密钥的密钥策略，但只能更改客户托管密钥的密钥策略的策略由 Amazon 托管式密钥 在您的账户中创建 KMS 密钥的 Amazon 服务创建和管理。您无法查看或更改 Amazon 拥有的密钥 的密钥策略。

- 您可以在密钥策略 Amazon Web Services 账户 中添加或删除 IAM 用户、IAM 角色以及更改允许或拒绝这些委托人的操作。有关在密钥策略中指定委托人和权限的方法的更多信息，请参阅密钥政策。
- 您无法向密钥策略添加 IAM 组，但可以添加多个 IAM 用户和 IAM 角色。有关更多信息，请参阅 允许多个 IAM 主体访问 KMS 密钥。
- 如果您在密钥策略中 Amazon Web Services 账户 添加外部策略，则还必须在外部账户中使用 IAM 策略向这些账户中的 IAM 用户、群组或角色授予权限。有关更多信息，请参阅 允许其他账户中的用户使用 KMS 密钥。
- 所生成的密钥策略文档不能超过 32 KB（32,768 字节）。

如何更改密钥策略

您可以通过三种不同的方式更改密钥策略，如以下各部分所述。

主题

- 使用 Amazon Web Services 管理控制台 默认视图
- 使用 Amazon Web Services 管理控制台 策略视图
- 使用 Amazon KMS API

使用 Amazon Web Services 管理控制台 默认视图

您可以使用控制台中名为默认视图的图形界面来更改密钥策略。

如果以下步骤与您在此控制台中看到的内容不一致，可能意味着，此密钥策略不是由此控制台创建的。也可能意味着，修改此密钥策略的方式不受控制台的默认视图的支持。在这种情况下，请按照使用 Amazon Web Services 管理控制台 策略视图或使用 Amazon KMS API中的步骤操作。

1. 查看客户托管密钥的密钥策略，如 使用控制 Amazon KMS 台 中所述。（您无法更改的密钥策略 Amazon 托管式密钥。）
2. 确定要更改的内容。

   - 要添加或删除密钥管理员以及允许或阻止密钥管理员删除 KMS 密钥，请使用此页面的 Key administrators（密钥管理员）部分中的控件。密钥管理员管理 KMS 密钥，包括启用和禁用它、设置密钥策略以及启用密钥轮换。
   - 要添加或删除密钥用户，以及允许或禁止外部 Amazon Web Services 账户 用户使用 KMS 密钥，请使用页面密钥用户部分中的控件。密钥用户可以在加密操作（如加密、解密、重新加密和生成数据密钥）中使用 KMS 密钥。

使用 Amazon Web Services 管理控制台 策略视图

您可以使用控制台的策略视图更改密钥策略文档。

1. 查看客户托管密钥的密钥策略，如 使用控制 Amazon KMS 台 中所述。（您无法更改的密钥策略 Amazon 托管式密钥。）
2. 在密钥策略部分中，选择切换到策略视图。
3. 选择编辑。
4. 确定要更改的内容。
   - 要添加新语句，请选择添加新语句。然后，您可以从语句构建器面板列出的选项中选择新密钥策略语句的操作、主体和条件，也可手动输入策略语句元素。
   - 要移除密钥策略中的某个语句，请选择该语句，然后选择移除。检查所选的策略语句并确认您要将其移除。如果您决定不移除该语句，请选择取消。
   - 要编辑现有的密钥策略语句，请选择该语句。然后，您可以使用语句生成器面板来选择要修改的特定元素，也可手动编辑该语句。
5. 选择保存更改。

使用 Amazon KMS API

您可以使用该PutKeyPolicy操作来更改您的 KMS 密钥的密钥策略 Amazon Web Services 账户。但不能对其他 Amazon Web Services 账户中的 KMS 密钥使用此 API。

1. 使用GetKeyPolicy操作获取现有的密钥策略文档，然后将密钥策略文档保存到文件中。有关多种编程语言中的示例代码，请参阅 GetKeyPolicy与 Amazon SDK 或 CLI 配合使用。
2. 在您的首选文本编辑器中打开该密钥策略文档，编辑该密钥策略文档，然后保存文件。
3. 使用PutKeyPolicy操作将更新的密钥策略文档应用于 KMS 密钥。有关多种编程语言中的示例代码，请参阅 PutKeyPolicy与 Amazon SDK 或 CLI 配合使用。

有关将密钥策略从一个 KMS 密钥复制到另一个 KMS 密钥的GetKeyPolicy 示例，请参阅《Amazon CLI 命令参考》中的示例。

关键策略中的 Amazon 服务权限

许多 Amazon 服务 Amazon KMS keys 用来保护其管理的资源。在服务使用 Amazon 拥有的密钥 或 Amazon 托管式密钥 时，该服务会为这些 KMS 密钥建立和维护密钥策略。

但是，当您通过 Amazon 服务使用客户托管式密钥时，您可以设置并维护密钥策略。该密钥策略必须允许服务具有代表您保护资源所需的最低权限。建议您遵循最低权限原则：仅授予服务所需的权限。您可以通过了解服务需要哪些权限，并使用 Amazon 全局条件键和 Amazon KMS 条件键来优化权限，以有效做到这一点。

要查找服务在客户托管式密钥上需要的权限，请参阅该服务的加密文档。以下列表包含一些服务文档的链接：

- Amazon CloudTrail权限-为配置 Amazon KMS 密钥策略 CloudTrail
- 亚马逊 Elastic Block Stor e 权限-亚马逊 EC2 用户指南和亚马逊 EC2 用户指南
- Amazon Lambda 权限：Lambda 的静态数据加密
- Amazon Q 权限：Data encryption for Amazon Q
- Amazon Relational Database Service 权限：Amazon KMS 密钥管理
- Amazon Secrets Manager 权限：Authorizing use of the KMS key
- Amazon Simple Queue Service 权限：Amazon SQS Key management

将 IAM 策略与配合使用 Amazon KMS

您可以使用 IAM 策略以及密钥策略、授权和 VPC 终端节点策略来控制对您 Amazon KMS keys 的 in 的访问权限 Amazon KMS。

> **Note**
>
> 要使用 IAM policy 控制对 KMS 密钥的访问，KMS 密钥的密钥政策必须授予账户使用 IAM policy 的权限。具体而言，密钥策略必须包含启用 IAM policy 的策略语句。
> 本节介绍如何使用 IAM 策略来控制对 Amazon KMS 操作的访问权限。有关 IAM 的更多一般信息，请参阅 IAM 用户指南。

所有 KMS 密钥都必须具有密钥策略。IAM policy 是可选的。要使用 IAM policy 控制对 KMS 密钥的访问，KMS 密钥的密钥政策必须授予账户使用 IAM policy 的权限。具体而言，密钥策略必须包含启用 IAM policy 的策略语句。

IAM 策略可以控制对任何 Amazon KMS 操作的访问权限。与密钥策略不同，IAM 策略可以控制对多个 KMS 密钥的访问权限，并为多个相关 Amazon 服务的操作提供权限。但是，IAM 策略对于控制对操作的访问特别有用 CreateKey，例如无法由密钥策略控制的操作，因为它们不涉及任何特定的 KMS 密钥。

如果您 Amazon KMS 通过亚马逊虚拟私有云 (Amazon VPC) 终端节点进行访问，则还可以在使用终端节点时使用 VPC 终端节点策略来限制对 Amazon KMS 资源的访问。例如，在使用 VPC 终端节点时，您可能只允许您的委托人 Amazon Web Services 账户 访问您的客户托管密钥。有关详细信息，请参阅 VPC 端点策略。

有关编写和格式化 JSON 策略文档的帮助，请参阅 IAM 用户指南中的 IAM JSON 策略参考。

您可以通过以下方式使用 IAM policy：

- 将@@ 权限策略附加到角色以获得联合身份验证或跨账户权限 — 您可以将 IAM 策略附加到 IAM 角色以启用联合身份验证、允许跨账户权限或向在实例上 EC2 运行的应用程序授予权限。有关 IAM 角色各种使用场景的更多信息，请参阅 IAM 用户指南中的 IAM 角色。
- 将权限策略附加到用户或组 – 您可以附加允许某个用户或用户组调用 Amazon KMS 操作的策略。但是，IAM 最佳实践建议您尽可能使用具有临时凭证的身份，例如 IAM 角色。

以下示例显示了具有 Amazon KMS 权限的 IAM 策略。此策略允许附加到其上的 IAM 身份获取列出所有 KMS 密钥和别名。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:ListKeys",
      "kms:ListAliases"
    ],
    "Resource": "*"
  }
}
```

与所有 IAM policy 一样，此策略没有 `Principal` 元素。将 IAM policy 附加到 IAM 身份时，该身份将获取策略中指定的权限。

有关显示所有 Amazon KMS API 操作及其适用的资源的表格，请参阅权限参考。

允许多个 IAM 主体访问 KMS 密钥

IAM 组不是密钥策略中的有效委托人。要允许多个 IAM 用户和角色访问 KMS 密钥，请执行下列操作中的一种：

- 将 IAM 角色作为密钥策略中的主体。多个授权用户可以根据需要代入该角色。有关详细信息，请参阅《IAM 用户指南》中的 IAM 角色。

  虽然您可以在密钥策略中列出多个 IAM 用户，但不建议采用这种做法，因为这将要求您在每次授权用户列表发生变化时更新密钥策略。此外，IAM 最佳实践也不鼓励使用具有长期凭证的 IAM 用户。有关更多信息，请参阅《IAM 用户指南》中的 IAM 安全最佳实践。
- 使用 IAM policy 向 IAM 组授予权限。要执行此操作，请确保密钥策略包含一个 启用 IAM policy 以允许访问 KMS 密钥 的语句，创建一个 IAM policy 以允许访问该 KMS 密钥，然后 将该策略附加到 IAM 组（其中包含授权 IAM 用户）。使用此方式，您不需要在授权用户列表发生更改时更改任何策略。相反，您只需在相应的 IAM 组中添加或删除这些用户。有关详细信息，请参阅《IAM 用户指南》中的 IAM 用户组。

有关 Amazon KMS 密钥策略和 IAM 策略如何协同工作的更多信息，请参阅Amazon KMS 权限疑难解答。

IAM policy 的最佳实践

确保访问 Amazon KMS keys 权限对您的所有 Amazon 资源的安全至关重要。KMS 密钥用于保护您的 Amazon Web Services 账户中很多最敏感的资源。花点时间设计控制对 KMS 密钥的访问权限的密钥政策、IAM 策略、授权和 VPC 端点策略。

在控制对 KMS 密钥的访问的 IAM policy 语句中，使用最低权限原则。仅为 IAM 委托人授予他们对必须使用或管理的 KMS 密钥的所需权限。

以下最佳实践适用于控制 Amazon KMS 密钥和别名访问权限的 IAM 策略。有关一般性的 IAM policy 最佳实践，请参阅《IAM 用户指南》中的 IAM 安全最佳实践。

使用密钥策略

尽可能在影响一个 KMS 密钥的密钥策略中提供权限，而不是在可应用于许多 KMS 密钥的 IAM policy 中提供权限，包括其他 Amazon Web Services 账户中的权限。这对于诸如 kms: PutKeyPolicy 和 kms: 之类的敏感权限尤其重要，对于决定如何保护数据的加密操作 ScheduleKeyDeletion也是如此。

限制 CreateKey 权限

仅向需要密钥 (kms:CreateKey) 的委托人授予创建密钥 (kms:) 的权限。创建 KMS 密钥的委托人还会设置其密钥策略，以便他们可以授予自己和其他人使用和管理他们创建的 KMS 密钥的权限。允许此权限时，请考虑通过使用策略条件限制它。例如，您可以使用 kms: KeySpec 条件来限制对称加密 KMS 密钥的权限。

在 IAM policy 中指定 KMS 密钥

最佳实践是在策略语句的 Resource 元素中指定权限所应用到的每个 KMS 密钥的密钥 ARN。此实践限制委托人需要的 KMS 密钥的权限。例如，此 Resource 元素仅列出主体需要使用的 KMS 密钥。

```
"Resource": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321"
]
```

当指定 KMS 密钥不切实际时，请使用一个限制访问可信 Amazon Web Services 账户 和区域（例如）中的 KMS 密钥的Resourcearn:aws:kms:*region*:*account*:key/*值。或者限制对受信任方所有区域 (*) 的 KMS 密钥的访问权限 Amazon Web Services 账户，例如arn:aws:kms:*:*account*:key/*。

不能在 IAM 策略的 Resource 字段中使用密钥 ID、别名名称或者别名 ARN 来表示 KMS 密钥。如果您指定别名 ARN，则策略将应用于别名，而不是 KMS 密钥。有关别名的 IAM policy 的信息，请参阅。控制对别名的访问

在 IAM policy 中避免使用 "Resource": "*"

谨慎地使用通配符 (*)。在密钥策略中，Resource 元素中的通配符表示密钥策略附加到的 KMS 密钥。但是在 IAM 策略中，仅在Resource元素 ("Resource": "*") 中使用通配符即可将权限应用于委托人账户有权 Amazon Web Services 账户 使用的所有 KMS 密钥。这可能包括其他密钥中的 KMS 密钥 Amazon Web Services 账户，以及委托人账户中的 KMS 密钥。

例如，要在另一个账户中使用 KMS 密钥 Amazon Web Services 账户，委托人需要获得外部账户中 KMS 密钥的密钥策略以及自己账户中的 IAM 策略的许可。假设一个任意账户授予了您对其 KMS 密钥的 Amazon Web Services 账户 kms:Decrypt 权限。若如此，您账户中授予角色对所有 KMS 密钥 ("Resource": "*") 的 kms:Decrypt 权限的 IAM policy 将满足要求的 IAM 部分。因此，可以担任该角色的委托人现在可以使用不可信账户中的 KMS 密钥解密密文。他们的操作条目会出现在两个账户的 CloudTrail 日志中。

特别是，避免在允许以下 API 操作的策略语句中使用 "Resource": "*"。可以在其他的 KMS 密钥上调用这些操作 Amazon Web Services 账户。

- DescribeKey
- GetKeyRotationStatus
- 加密操作（加密、解密、、、、、GenerateDataKey、GenerateDataKeyPair、、GenerateDataKeyWithoutPlaintextGenerat名GetPublicKeyReEncrypt、验证）
- CreateGrant, ListGrants, ListRetirableGrants, RetireGrant, RevokeGrant

何时使用 "Resource": "*"

在 IAM policy 中，仅将 Resource 元素中的通配符用于需要它的权限。只有以下权限才需要 "Resource": "*" 元素。

- kms: CreateKey
- kms: GenerateRandom
- kms: ListAliases
- kms: ListKeys
- 自定义密钥存储库的权限，例如 k ms: CreateCustomKeyStore 和 kms: ConnectCustomKeyStore。

> **Note**
>
> 别名操作（kms:、kms: CreateAlias 、k ms: UpdateAlias、kms: DeleteAlias）的权限必须附加到别名和 KMS 密钥。您可以使用 IAM policy 中的 "Resource": "*" 来表示别名和 KMS 密钥，或者在 Resource 元素中指定别名和 KMS 密钥。有关示例，请参阅 控制对别名的访问。

本主题中的示例提供了有关设计 KMS 密钥的 IAM policy 的详细信息和指导。有关所有 Amazon 资源的 IAM 最佳实践，请参阅 IAM 用户指南中的 IAM 安全最佳实践。

在 IAM policy 语句中指定 KMS 密钥

您可以使用 IAM policy 来允许委托人使用或管理 KMS 密钥。KMS 密钥在策略语句的 Resource 元素中指定。

- 要在 IAM policy 语句中指定 KMS 密钥，必须使用其密钥 ARN。您不能使用密钥 ID、别名名称或别名 ARN 来标识 IAM policy 语句中的 KMS 密钥。

  例如：“Resource": "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab”

  要根据别名控制对 KMS 密钥的访问权限，请使用 kms: RequestAlias 或 kms: ResourceAliases 条件密钥。有关更多信息，请参阅 ABAC for Amazon KMS。

  仅在控制别名操作（例如、或）访问权限的策略声明中使用别名 ARN 作为CreateAlias资源。UpdateAliasDeleteAlias有关更多信息，请参阅 控制对别名的访问。
- 要在账户和区域中指定多个 KMS 密钥，请在密钥 ARN 的区域或资源 ID 位置中使用通配符 (*)。

  例如，要指定账户的美国西部（俄勒冈）区域中的所有 KMS 密钥，请使用“Resource": "arn:aws:kms:us-west-2:111122223333:key/*”。要指定账户的所有区域中的所有 KMS 密钥，请使用“Resource": "arn:aws:kms:*:111122223333:key/*”。
- 要表示所有 KMS 密钥，请单独使用通配符 ("*")。对于不使用任何特定 KMS 密钥（即、和）的操作 CreateKeyGenerateRandomListAliases，请使用此格式ListKeys。

在编写策略语句时，最佳实践是只指定委托人需要使用的 KMS 密钥，而不是授予他们对所有 KMS 密钥的访问权限。

例如，以下 IAM 策略声明仅允许委托人对策略声明Resource元素中列出的 KMS 密钥调用DescribeKeyGenerateDataKey、、Decrypt 操作。通过密钥 ARN 指定 KMS 密钥是一种最佳实践，可确保权限仅限于指定的 KMS 密钥。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:DescribeKey",
      "kms:GenerateDataKey",
      "kms:Decrypt"
    ],
    "Resource": [
```

```
      "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "arn:aws:kms:us-
west-2:111122223333:key/01234abcd-12ab-34cd-56ef-1234567890ab"
    ]
  }
}
```

要将权限应用于特定可信对象中的所有 KMS 密钥 Amazon Web Services 账户，可以在区域和密钥 ID 位置使用通配符 (*)。例如，以下策略语句允许委托人对两个可信示例账户的中的 KMS 密钥调用指定操作。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:DescribeKey",
      "kms:GenerateDataKey",
      "kms:GenerateDataKeyPair"
    ],
    "Resource": [
      "arn:aws:kms:*:111122223333:key/*",
      "arn:aws:kms:*:444455556666:key/*"
    ]
  }
}
```

您还可以单独在 Resource 元素中使用通配符 ("*")。由于它允许访问帐户有权使用的所有 KMS 密钥，因此建议主要用于没有特定 KMS 密钥的操作和 Deny 语句。您还可以在仅允许不太敏感的只读操作的策略语句中使用它。要确定某项 Amazon KMS 操作是否涉及特定的 KMS 密钥，请在中表的 “资源” 列中查找 KMS 密钥值the section called “权限参考”。

例如，以下策略语句使用 Deny 效果来禁止委托人对任何 KMS 密钥使用指定的操作。它在 Resource 元素中使用通配符来表示所有 KMS 密钥。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Deny",
    "Action": [
      "kms:CreateKey",
      "kms:PutKeyPolicy",
      "kms:CreateGrant",
      "kms:ScheduleKeyDeletion"
    ],
    "Resource": "*"
  }
}
```

以下策略语句单独使用通配符来表示所有 KMS 密钥。但它只允许不太敏感的只读操作和不适用于任何特定 KMS 密钥的操作。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:CreateKey",
      "kms:ListKeys",
      "kms:ListAliases",
      "kms:ListResourceTags"
    ],
    "Resource": "*"
  }
}
```

IAM 策略示例

在此部分中，可以找到允许执行各种 Amazon KMS 操作的权限的示例 IAM policy。

> **Important**
>
> 仅当 KMS 密钥的密钥策略也允许时，才允许提供以下策略中的某些权限。有关更多信息，请参阅 权限参考。

有关编写和格式化 JSON 策略文档的帮助，请参阅 IAM 用户指南中的 IAM JSON 策略参考。

示例

- 允许用户在 Amazon KMS 控制台中查看 KMS 密钥
- 允许用户创建 KMS 密钥
- 允许用户使用特定 KMS 密钥进行加密和解密 Amazon Web Services 账户
- 允许用户使用特定 Amazon Web Services 账户 和区域中的任何 KMS 密钥进行加密和解密
- 允许用户使用特定 KMS 密钥进行加密和解密
- 阻止用户禁用或删除任何 KMS 密钥

允许用户在 Amazon KMS 控制台中查看 KMS 密钥

以下 IAM 策略允许用户以只读方式访问 Amazon KMS 控制台。拥有这些权限的用户可以查看其中的所有 KMS 密钥 Amazon Web Services 账户，但他们无法创建或更改任何 KMS 密钥。

要在Amazon 托管式密钥和客户托管密钥页面上查看 K MS 密钥，即使密钥没有标签或别名 ListAliases，委托人也需要 k m s:、kms: 和 tag: GetResources 权限。ListKeys在 KMS 密钥详细信息页面上查看可选的 KMS 密钥表列和数据需要其余权限，尤其是 kms: DescribeKey。需要 ia m: ListUsers 和 ia ListRoles m: 权限才能在默认视图中毫无错误地显示密钥策略。要查看自定义密钥存储库页面上的数据以及自定义密钥存储库中 KMS 密钥的详细信息，委托人还需要 k ms: DescribeCustomKeyStores 权限。

如果您限制用户的控制台对特定 KMS 密钥的访问，控制台将显示不可见的每个 KMS 密钥的错误。

此策略包含两个策略语句。第一个策略语句中的 `Resource` 元素允许对示例 Amazon Web Services 账户的所有区域中的所有 KMS 密钥的指定权限。控制台查看器不需要额外的访问权限，因为 Amazon KMS 控制台仅显示委托人账户中的 KMS 密钥。即使他们有权在其他版本中查看 KMS 密钥，也是如此 Amazon Web Services 账户。其余 Amazon KMS 和 IAM 权限需要一个`"Resource": "*"`元素，因为它们不适用于任何特定的 KMS 密钥。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "ReadOnlyAccessForAllKMSKeysInAccount",
      "Effect": "Allow",
      "Action": [
        "kms:GetPublicKey",
        "kms:GetKeyRotationStatus",
        "kms:GetKeyPolicy",
        "kms:DescribeKey",
        "kms:ListKeyPolicies",
        "kms:ListResourceTags",
        "tag:GetResources"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*"
    },
    {
      "Sid": "ReadOnlyAccessForOperationsWithNoKMSKey",
      "Effect": "Allow",
      "Action": [
        "kms:ListKeys",
        "kms:ListAliases",
        "iam:ListRoles",
        "iam:ListUsers"
      ],
      "Resource": "*"
    }
  ]
}
```

允许用户创建 KMS 密钥

以下 IAM policy 允许用户创建所有类型的 KMS 密钥。`Resource`元素的值`*`是因为该`CreateKey`操作不使用任何特定的 Amazon KMS 资源（KMS 密钥或别名）。

要限制用户使用特定类型的 KMS 密钥，请使用 k ms: KeySpec、kms: KeyUsage 和 k ms: KeyOrigin 条件密钥。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": "kms:CreateKey",
    "Resource": "*"
  }
}
```

创建密钥的委托人可能需要一些相关权限。

- kms: PutKeyPolicy — 拥有kms:CreateKey权限的委托人可以为 KMS 密钥设置初始密钥策略。但是，CreateKey调用者必须拥有 k ms: PutKeyPolicy 权限，允许他们更改 KMS 密钥策略，或者必须指定BypassPolicyLockoutSafetyCheck参数CreateKey，但不建议这样做。CreateKey 调用方可以从 IAM policy 中获得对 KMS 密钥的 kms:PutKeyPolicy 权限，也可以将此权限包含在他们正在创建的 KMS 密钥的密钥策略中。
- kms: TagResource — 要在CreateKey操作期间向 KMS 密钥添加标签，CreateKey调用者必须在 IAM 策略中TagResource拥有 kms: 权限。将此权限包含在新 KMS 密钥的密钥策略中是不够的。但是，如果 CreateKey 调用方在初始密钥策略中包括 kms:TagResource，他们可以在创建 KMS 密钥后在单独调用中添加标签。
- kms: CreateAlias — 在 Amazon KMS 控制台中创建 KMS 密钥的委托人必须拥有 KMS: 密钥和别名的CreateAlias权限。（控制台进行两次调用；一次对 CreateKey，一次对 CreateAlias）。您必须在 IAM policy 中提供别名权限。您可以在密钥策略或 IAM policy 中提供 KMS 密钥权限。有关更多信息，请参阅 控制对别名的访问。

此外kms:CreateKey，以下 IAM 策略提供kms:TagResource对中所有 KMS 密钥的kms:CreateAlias权限 Amazon Web Services 账户 以及对账户所有别名的权限。它还包括一些只能在 IAM policy 中提供的有用的只读权限。

此 IAM policy 不包含 kms:PutKeyPolicy 权限或可以在密钥策略中设置的任何其他权限。它是在密钥策略中设置这些权限的最佳实践，在该密钥策略中，这些权限专门应用于一个 KMS 密钥。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPermissionsForParticularKMSKeys",
      "Effect": "Allow",
      "Action": "kms:TagResource",
      "Resource": "arn:aws:kms:*:111122223333:key/*"
    },
    {
      "Sid": "IAMPermissionsForParticularAliases",
      "Effect": "Allow",
      "Action": "kms:CreateAlias",
      "Resource": "arn:aws:kms:*:111122223333:alias/*"
    },
    {
      "Sid": "IAMPermissionsForAllKMSKeys",
      "Effect": "Allow",
      "Action": [
        "kms:CreateKey",
        "kms:ListKeys",
        "kms:ListAliases"
      ],
      "Resource": "*"
    }
  ]
}
```

允许用户使用特定 KMS 密钥进行加密和解密 Amazon Web Services 账户

以下 IAM 策略允许用户使用 111122223333 中的 Amazon Web Services 账户 任何 KMS 密钥加密和解密数据。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
```

```
    "Effect": "Allow",
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt"
    ],
    "Resource": "arn:aws:kms:*:111122223333:key/*"
  }
}
```

允许用户使用特定 Amazon Web Services 账户 和区域中的任何 KMS 密钥进行加密和解密

以下 IAM 策略允许用户使用美国西部（俄勒冈）地区的任何 KMS 密钥加密和解密数据。 Amazon Web Services 账户 111122223333

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt"
    ],
    "Resource": [
      "arn:aws:kms:us-west-2:111122223333:key/*"
    ]
  }
}
```

允许用户使用特定 KMS 密钥进行加密和解密

以下 IAM policy 允许用户使用 `Resource` 元素中指定的两个 KMS 密钥来加密和解密数据。在 IAM policy 语句中指定 KMS 密钥时，必须使用 KMS 密钥的密钥 ARN。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt"
    ],
    "Resource": [
      "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "arn:aws:kms:us-west-2:111122223333:key/01234abc-d12a-b34c-d56e-
f1234567890a'"
    ]
  }
}
```

阻止用户禁用或删除任何 KMS 密钥

以下 IAM policy 阻止用户禁用或删除任何 KMS 密钥，即使其他 IAM policy 或密钥策略允许这些权限时也是如此。以显式方式拒绝权限的策略将覆盖所有其他策略，甚至包括那些以显式方式允许相同权限的策略。有关更多信息，请参阅 Amazon KMS 权限疑难解答。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Deny",
    "Action": [
      "kms:DisableKey",
      "kms:ScheduleKeyDeletion"
    ],
    "Resource": "*"
  }
}
```

中的资源控制策略 Amazon KMS

资源控制策略 (RCPs) 是一种组织策略，可用于对组织中的 Amazon 资源实施预防性控制。 RCPs 帮助您大规模集中限制外部访问您的 Amazon 资源。 RCPs 补充服务控制策略 (SCPs)。虽然 SCPs 可用于集中设置组织中 IAM 角色和用户的最大权限，但 RCPs 可用于集中设置组织中 Amazon 资源的最大权限。

您可以使用 RCPs 管理组织中客户托管的 KMS 密钥的权限。 RCPs 仅凭向您的客户托管密钥授予权限是不够的。RCP 不会授予任何权限。RCP 针对各种身份可对受影响账户中资源执行的操作定义权限护栏或设置限制。管理员仍然必须将基于身份的策略附加到 IAM 用户或角色，或者附加密钥策略，才能实际授予权限。

> **Note**
>
> 组织中的资源控制策略不适用于 Amazon 托管式密钥。
> Amazon 托管式密钥 由 Amazon 服务代表您创建、管理和使用，您无法更改或管理其权限。

了解更多

- 有关的更多一般信息 RCPs，请参阅《Amazon Organizations 用户指南》中的资源控制策略。
- 有关如何定义的详细信息（包括示例）RCPs，请参阅《Amazon Organizations 用户指南》中的 RCP 语法。

以下示例展示了如何使用 RCP 来阻止外部主体访问您组织中的客户自主管理型密钥。此策略仅为示例，需要根据您自己独特的业务和安全需求对其进行定制。例如，您可能需要对该策略进行自定义，以允许业务合作伙伴访问。有关更多详细信息，请参阅 data perimeter policy examples repository。

> **Note**
>
> 即使 `Action` 元素将星号（`*`）指定为通配符，`kms:RetireGrant` 权限在 RCP 中也无效。
> 要详细了解如何确定 `kms:RetireGrant` 权限，请参阅停用和撤销授权。

JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "RCPEnforceIdentityPerimeter",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "kms:*",
            "Resource": "*",
            "Condition": {
                "StringNotEqualsIfExists": {
                    "aws:PrincipalOrgID": "my-org-id"
                },
                "Bool": {
                    "aws:PrincipalIsAWSService": "false"
                }
            }
        }
    ]
}
```

以下示例 RCP 要求 Amazon 服务委托人只有在请求来自您的组织时才能访问您的客户托管的 KMS 密钥。此策略仅对存在的 `aws:SourceAccount` 请求执行控制。这样可以确保不需要使用 `aws:SourceAccount` 的服务集成不会受到影响。如果请求上下文中存在 `aws:SourceAccount`，则 `Null` 条件将评估为 `true`，从而强制执行 `aws:SourceOrgID` 键。

有关混淆代理问题的更多信息，请参阅《IAM 用户指南》中的混淆代理问题。

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Sid": "RCPEnforceConfusedDeputyProtection",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "kms:*",
            "Resource": "*",
            "Condition": {
                "StringNotEqualsIfExists": {
```

```
                    "aws:SourceOrgID": "my-org-id"
                },
                "Bool": {
                    "aws:PrincipalIsAWSService": "true"
                },
                "Null": {
                    "aws:SourceAccount": "false"
                }
            }
        }
    ]
}
```

Amazon KMS 中的授权

授权是一种策略分析工具，允许 Amazon 主体将 KMS 密钥用于加密操作中。它还可以让他们查看 KMS 密钥 (DescribeKey) 以及创建和管理授权。在授权访问 KMS 密钥时，将考虑授权与密钥策略和 IAM policy。授权通常用于临时权限，因为您可以在不更改密钥策略或 IAM policy 的情况下创建授权、使用其权限并将其删除。

授权通常被与 Amazon KMS 集成的 Amazon 服务用来加密静态数据。该服务代表账户中的用户创建授权，使用其权限，并在其任务完成后立即停用授权。有关 Amazon 服务如何使用授权的详细信息，请参阅服务的《用户指南》或《开发人员指南》中的“静态加密”主题。

授权是一种非常灵活且有用的访问控制机制。当您为 KMS 密钥创建授权时，授权允许被授权主体对 KMS 密钥调用指定授权操作，前提是该授权中指定的所有条件都得到满足。

- 每个授权只允许访问一个 KMS 密钥。您可以在不同的 Amazon Web Services 账户 中为 KMS 密钥创建授权。
- 授权可以允许访问 KMS 密钥，但不能拒绝访问。
- 每个授权都有一名被授权主体。被授权主体可以在 KMS 密钥的同一个 Amazon Web Services 账户中或在不同的账户中代表一个或多个身份。
- 授权只能允许授权操作。授权操作必须由授权中的 KMS 密钥支持。如果您指定了不受支持的操作，CreateGrant 请求将失败，并显示 ValidationError 异常。
- 被授权主体可以使用授权给予他们的权限，而无需指定授权，就像权限来自密钥策略或 IAM policy 一样。但是，由于 Amazon KMS API 采用最终一致性模型，当您创建、停用或撤销授权时，可能会出现短暂的延迟，才能使更改在整个 Amazon KMS 中可用。要立即使用授权中的权限，请使用授权令牌。

- 授权委托人可以删除授权（停用或者撤销它）。删除授权会清除授权允许的所有权限。您不必确定要添加或删除哪些策略来撤销授权。
- Amazon KMS 限制每个 KMS 密钥上的授权数量。有关更多信息，请参阅 每个 KMS 密钥的授权数：50000。

在创建授权和给予其他人创建授权的权限时务必谨慎。创建授权的权限具有安全影响，就像允许 kms:PutKeyPolicy 权限设置策略一样。

- 有权为 KMS 密钥 (kms:CreateGrant) 创建授权的用户可以使用授权来允许用户和角色（包括 Amazon 服务）使用 KMS 密钥。委托人可以是您自己的 Amazon Web Services 账户 中的身份或其他账户或组织中的身份。
- 授权只能允许 Amazon KMS 运算符子集。您可以使用授权允许委托人查看 KMS 密钥，在加密操作中使用它，以及创建和停用授权。有关详细信息，请参阅授权操作。您还可以使用授权约束来限制对称加密密钥授权中的权限。
- 委托人可以获得从密钥策略或 IAM policy 创建授权的权限。通过策略获得的 kms:CreateGrant 权限的主体可以为基于 KMS 密钥的任何授权操作创建授权。这些主体无需拥有他们对密钥的授权权限。当您在策略中允许 kms:CreateGrant 权限时，您可以使用策略条件来限制此权限。
- 委托人还可以获得从授权创建授权的权限。这些主体只能委派他们被授予的权限，即使他们具有来自策略的其他权限也是如此。有关更多信息，请参阅 授予 CreateGrant 权限。

授权概念

为了有效地使用授权，您需要了解 Amazon KMS 使用的术语和概念。

授权约束

限制授权中的权限的条件。目前，Amazon KMS 基于请求中的加密上下文支持将授权约束用于加密操作。有关更多信息，请参阅 使用授权约束。

授权 ID

KMS 密钥的授权的唯一标识符。您可以使用授权 ID 和密钥标识符，以标识 RetireGrant 或 RevokeGrant 请求中的授权。

授权操作

您可以在授权中允许的 Amazon KMS 操作。如果您指定了其他操作，CreateGrant 请求将失败，并显示 ValidationError 异常。这些也是接受授权令牌的操作。有关这些权限的详细信息，请参阅 Amazon KMS 权限。

这些授权操作实际上代表使用操作的权限。因此，对于 ReEncrypt 操作，您可以指定 ReEncryptFrom、ReEncryptTo 或此两者 ReEncrypt*。

授权操作包括：

- 加密操作
  - Decrypt
  - DeriveSharedSecret
  - Encrypt
  - GenerateDataKey
  - GenerateDataKeyPair
  - GenerateDataKeyPairWithoutPlaintext
  - GenerateDataKeyWithoutPlaintext
  - GenerateMac
  - ReEncryptFrom
  - ReEncryptTo
  - Sign
  - Verify
  - VerifyMac
- 其他操作
  - CreateGrant
  - DescribeKey
  - GetPublicKey
  - RetireGrant

您允许的授权操作必须由授权中的 KMS 密钥支持。如果您指定了不受支持的操作，CreateGrant 请求将失败，并显示 ValidationError 异常。例如，对称加密 KMS 密钥的授权不能允许 Sign、Verify、GenerateMac 或 VerifyMac 操作。非对称 KMS 密钥的授权不能允许生成数据密钥或数据密钥对的操作。

授权令牌

Amazon KMS API 采用最终一致性模型。当您创建授权时，可能会出现短暂的延迟，才能使更改在整个 Amazon KMS 中可用。更改通常需要不到几秒钟的时间即可在整个系统中传播，但在某些情况下，可能需要几分钟。如果您尝试在系统中完全传播之前使用授权，您可能会收到访问被拒绝的错误。授权令牌允许您引用授权并立即使用授权权限。

授权令牌是代表授权的唯一、非秘密、长度可变的 base64 编码字符串。您可以使用授权令牌来标识任何授权操作中的授权。但是，由于令牌值是哈希摘要，它不会显示有关授权的任何详细信息。

授权令牌设计为仅在授权传播到整个 Amazon KMS 中时使用。之后，被授权者委托人可以在不提供授权令牌或授权的任何其他证据的情况下使用授权中的权限。您可以随时使用授权令牌，但是一旦授权达到最终一致性，Amazon KMS 就会使用授权来确定权限，而不是授权令牌。

例如，以下命令调用 GenerateDataKey 操作。它使用授权令牌来表示给予调用者（被授权者委托人）对指定的 KMS 密钥调用 `GenerateDataKey` 的权限的授权。

```
$ aws kms generate-data-key \
        --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
        --key-spec AES_256 \
        --grant-token $token
```

您还可以使用授权令牌来标识管理授权的操作中的授权。例如，停用委托人可以在调用 RetireGrant 操作时使用授权令牌。

```
$ aws kms retire-grant \
        --grant-token $token
```

`CreateGrant` 是返回授权令牌的唯一操作。您无法从任何其他 Amazon KMS 操作或从 CreateGrant 操作的 CloudTrail 日志事件中获取授权令牌。ListGrants 和 ListRetirableGrants 操作将返回授权 ID，而不是授权令牌。

有关更多信息，请参阅 使用授权令牌。

被授权者委托人

获取授权中指定的权限的身份。每个授权都有一个被授权主体，但被授权主体可以代表多个身份。

获授权主体可以是任何 Amazon 主体，包括 Amazon Web Services 账户（根）、IAM 用户、IAM 角色、联合角色或用户或代入的角色用户。被授权者委托人可以与 KMS 密钥位于同一账户中，也可以位于不同的账户中。但是，被授权者委托人不能是服务委托人、IAM 组，或 Amazon组织。

> **Note**
>
> IAM 最佳实践不鼓励使用具有长期凭证的 IAM 用户。而应尽可能使用提供临时凭证的 IAM 角色。有关更多信息，请参阅《IAM 用户指南》中的 IAM 安全最佳实践。

停用（授权）

终止授权。当您使用完权限时，将停用授权。

撤销和停用授权都会删除授权。但是，停用由授权中指定的委托人完成。撤消通常由密钥管理员执行。有关更多信息，请参阅 停用和撤销授权。

停用委托人

可以停用授权的委托人。您可以在授权中指定停用委托人，但这不是必需的。停用委托人可以是任何 Amazon 委托人，包括 Amazon Web Services 账户、IAM 用户、IAM 角色、联合身份用户以及代入的角色用户。停用委托人可以与 KMS 密钥位于同一账户中，也可以位于不同的账户中。

> **Note**
>
> IAM 最佳实践不鼓励使用具有长期凭证的 IAM 用户。而应尽可能使用提供临时凭证的 IAM 角色。有关更多信息，请参阅《IAM 用户指南》中的 IAM 安全最佳实践。

除了授权中指定的停用委托人外，还可通过在其中创建授权的 Amazon Web Services 账户 停用授权。如果授权允许 `RetireGrant` 操作，被授权者委托人可以停用授权。另外，Amazon Web Services 账户 或 停用委托人的 Amazon Web Services 账户 可以委派权限以停用相同 Amazon Web Services 账户 中的 IAM 委托人的授权。有关更多信息，请参阅 停用和撤销授权。

撤销（授予）

终止授权。您将撤销积极拒绝授权允许的权限的授权。

撤销和停用授权都会删除授权。但是，停用由授权中指定的委托人完成。撤消通常由密钥管理员执行。有关更多信息，请参阅 停用和撤销授权。

最终一致性（用于授权）

Amazon KMS API 采用最终一致性模型。当您创建、停用或撤销授权时，可能会出现短暂的延迟，才能使更改在整个 Amazon KMS 中可用。更改通常需要不到几秒钟的时间即可在整个系统中传播，但在某些情况下，可能需要几分钟。

如果您遇到意外错误，您可能会注意到这个短暂的延迟。例如，如果您尝试在整个 Amazon KMS 知晓授权前管理一个新的授权或使用新授权中的权限，您可能会收到访问被拒绝错误。如果您停用或撤销授权，则被授权者委托人可能仍然能够在短时间内使用其权限，直到完全删除该授权为止。典型的策略是重试请求，且一些 Amazon 开发工具包包括自动退避和重试逻辑。

Amazon KMS 具有缓解这一短暂延迟的功能。

- 要立即使用新授权中的权限，请使用授权令牌。您可以使用授权令牌来引用任何授权操作中的授权。有关说明，请参阅使用授权令牌。
- CreateGrant 操作具有 Name 参数，以防止重试操作创建重复的授权。

> **Note**
>
> 授权令牌将取代授权的有效性，直到服务中的所有终端节点都使用新的授权状态更新为止。在大多数情况下，最终一致性将在五分钟内实现。

有关更多信息，请参阅Amazon KMS 最终一致性。

Amazon KMS 授权的最佳实践

Amazon KMS 建议在创建、使用和管理授权时使用以下最佳实践。

- 将授权中的权限限制为被授权者委托人所需的权限。使用最小特权访问权限的原则。
- 使用特定的被授权者委托人（如 IAM 角色），并授予被授权委托人仅使用他们所需的 API 操作的权限。
- 使用加密上下文授权约束以确保调用方正在将 KMS 密钥用于预期目的。有关如何在请求中使用加密上下文以保护数据的详细信息，请参阅 Amazon 安全博客中的如何使用 Amazon Key Management Service 和 EncryptionContext 保护您的加密数据完整性。

> **Tip**
>
> 尽可能使用 EncryptionContextEqual 授权约束。EncryptionContextSubset 授权约束更难以正确使用。如果您需要使用它，请仔细阅读文档并测试授权约束以确保它按预期工作。

- 删除重复的授权。重复授权具有相同的密钥 ARN、API 操作、被授权者委托人、加密上下文和名称。如果您停用或撤销原始授予，但保留重复项授权，则剩余的重复授权将构成意外的权限提升。为了在重试 `CreateGrant` 请求时避免重复授权，请使用 Name 参数。要检测重复的授权，请使用 ListGrants 操作。如果您意外创建了重复授权，请尽快停用或撤销该授权。

> **Note**
>
> Amazon 托管密钥的授权可能看起来像重复授权，但具有不同的被授权者委托人。

`GranteePrincipal` 响应中的 `ListGrants` 字段通常包含授权的被授权者委托人。但是，当授权中的被授权者委托人是 Amazon 服务时，`GranteePrincipal` 字段包含服务委托人，该委托人可能表示多个不同的被授权者委托人。

- 请记住，授权不会自动过期。当权限不再需要时，立即停用或撤销授权。未删除的授权可能会对加密资源造成安全风险。

控制对授权的访问

您可以控制对在密钥策略、IAM policy 和授权中创建和管理授权的操作的访问权限。从授权中获得 `CreateGrant` 权限的委托人具有更有限的授权权限。

API 操作	密钥策略或 IAM policy	授权
CreateGrant	✓	✓
ListGrants	✓	-
ListRetirableGrants	✓	-
停用授权	（有限。请参阅 停用和撤销授权）	✓
RevokeGrant	✓	-

您在使用密钥策略或 IAM policy 以控制对创建和管理授权的操作的访问时，可以使用一个或多个以下策略条件来限制权限。Amazon KMS 支持以下所有与授权相关的条件键。有关详细信息和示例，请参阅 Amazon KMS 条件键。

kms:GrantConstraintType

允许委托人仅在授权包含指定的授权约束时创建授权。

kms:GrantIsForAWSResource

仅在仅在与 Amazon KMS 集成的 Amazon 服务代表主体发送请求时允许主体调用 `CreateGrant`、`ListGrants` 或 `RevokeGrant`。

kms:GrantOperations

允许委托人创建授权，但将授权限制为指定操作。

kms:GranteePrincipal

允许委托人仅为指定的被授权者委托人创建授权。

kms:RetiringPrincipal

允许委托人仅在授权指定特定的停用委托人时创建授权。

创建授权

在创建授权之前，请了解用于自定义授权的选项。您可以使用授权约束来限制授权中的权限。此外，了解授予 CreateGrant 权限的相关信息。获得从授权创建授权的权限的委托人在其可以创建的授权中受到限制。

主题

- 创建授予
- 授予 CreateGrant 权限

创建授予

要创建授权，请调用 CreateGrant 操作。指定 KMS 密钥，被授权者委托人，以及允许的授权操作的列表。您还可以指定一个可选的停用委托人。要自定义授权，请使用可选 Constraints 参数来定义授予约束。

当您创建、停用或撤销授权时，可能会出现短暂的延迟（通常不到五分钟），才能使更改在整个 Amazon KMS 中可用。有关更多信息，请参阅最终一致性（用于授权）。

例如，以下 CreateGrant 命令会创建一个授权，以允许被授权代入 keyUserRole 角色的用户对指定的 对称 KMS 密钥 调用 解密 操作。授权使用 RetiringPrincipal 参数，指定可以停用授权的委托人。其中还包含一个授权约束，仅当请求中的加密上下文包含 "Department": "IT" 时才允许该权限。

```
$  aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/keyUserRole \
    --operations Decrypt \
```

```
    --retiring-principal arn:aws:iam::111122223333:role/adminRole \
    --constraints EncryptionContextSubset={Department=IT}
```

如果您的代码重试 CreateGrant 操作，或者使用自动重试请求的 Amazon SDK，请使用可选的 Name 参数来防止创建重复授权。如果 Amazon KMS 针对与现有授权具有相同属性（包括名称）的授权获取一个 CreateGrant 请求，它会将该请求识别为重试，并且不创建新授权。您无法使用 Name 值来标识任何 Amazon KMS 操作中的授权。

> **Important**
>
> 不要在授权名称中包含机密或敏感信息。该名称可能以纯文本形式出现在 CloudTrail 日志和其他输出中。

```
$ aws kms create-grant \
    --name IT-1234abcd-keyUserRole-decrypt \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/keyUserRole \
    --operations Decrypt \
    --retiring-principal arn:aws:iam::111122223333:role/adminRole \
    --constraints EncryptionContextSubset={Department=IT}
```

有关演示如何通过多种编程语言创建授权的代码示例，请参阅 CreateGrant与 Amazon SDK 或 CLI 配合使用。

使用授权约束

授权约束设置授权给予被授权者委托人的权限的条件。授权约束取代密钥策略或 IAM policy 中的条件键。每个授权约束值最多可以包含 8 个加密上下文对。每个授权约束中的加密上下文值不能超过 384 个字符。

> **Important**
>
> 不要在此字段中包含机密或敏感信息。此字段可能会以纯文本形式显示在 CloudTrail 日志和其他输出中。

Amazon KMS 支持两个授权约束 EncryptionContextEquals 和 EncryptionContextSubset，两者都在加密操作的请求中建立了对加密上下文的要求。

加密上下文授权约束旨在与具有加密上下文参数的授权操作结合使用。

- 加密上下文约束仅在对称加密 KMS 密钥的授权中有效。使用其他 KMS 密钥的加密操作不支持加密上下文。
- 对于 DescribeKey 和 RetireGrant 操作，加密上下文约束将被忽略。DescribeKey 和 RetireGrant 不包括加密上下文参数，但您可以将这些操作包含在具有加密上下文约束的授权中。
- 您可以将授权中的加密上下文约束用于 CreateGrant 操作。加密上下文约束要求使用 CreateGrant 权限创建的任何授权具有同样严格或更严格的加密上下文约束。

Amazon KMS 支持以下加密上下文授权约束。

EncryptionContextEquals

使用 EncryptionContextEquals 为允许的请求指定精确的加密上下文。

EncryptionContextEquals 要求请求中的加密上下文对与授权约束中加密上下文对区分大小写的完全匹配。上下文对可以按任意顺序显示，不过每一对中的键和值不能有改变。

例如，如果 EncryptionContextEquals 授权约束需要 "Department": "IT" 加密上下文对，则授权仅在请求中的加密上下文完全是 "Department": "IT" 时，允许指定类型的请求。

EncryptionContextSubset

使用 EncryptionContextSubset 来要求请求包含特定的加密上下文对。

EncryptionContextSubset 要求请求中包含授权约束中的所有加密上下文对（区分大小写的完全匹配），但请求也可以包含其他的加密上下文对。上下文对可以按任意顺序显示，不过每一对中的键和值不能有改变。

例如，如果 EncryptionContextSubset 授权约束需要 Department=IT 加密上下文对，则授权在请求中的加密上下文为 "Department": "IT" 或者包含 "Department": "IT" 以及其他加密上下文对（例如 "Department": "IT","Purpose": "Test"）时，允许指定类型的请求。

要在对称加密 KMS 密钥的授权中指定加密上下文约束，请使用 CreateGrant 操作中的 Constraints 参数。此命令创建的授权将向被授权代入 keyUserRole 角色的用户调用 解密 操作的权限。不过，该权限仅在 Decrypt 请求中的加密上下文是 "Department": "IT" 加密上下文对时有效。

```
$ aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
```

```
    --grantee-principal arn:aws:iam::111122223333:role/keyUserRole \
    --operations Decrypt \
    --retiring-principal arn:aws:iam::111122223333:role/adminRole \
    --constraints EncryptionContextEquals={Department=IT}
```

生成的授权与以下项目类似。请注意，向 keyUserRole 角色授予的权限仅在 Decrypt 请求使用授权约束中指定的相同加密上下文对时有效。要查找对 KMS 密钥的授权，请使用 ListGrants 操作。

```
$ aws kms list-grants --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "Grants": [
        {
            "Name": "",
            "IssuingAccount": "arn:aws:iam::111122223333:root",
            "GrantId":
 "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a",
            "Operations": [
                "Decrypt"
            ],
            "GranteePrincipal": "arn:aws:iam::111122223333:role/keyUserRole",
            "Constraints": {
                "EncryptionContextEquals": {
                    "Department": "IT"
                }
            },
            "CreationDate": 1568565290.0,
            "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "RetiringPrincipal": "arn:aws:iam::111122223333:role/adminRole"
        }
    ]
}
```

为了满足 EncryptionContextEquals 授权约束，Decrypt 操作的请求中的加密上下文必须是 "Department": "IT" 对。来自被授予者委托人的以下请求将满足 EncryptionContextEquals 授权约束。

```
$ aws kms decrypt \
    --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab\
    --ciphertext-blob fileb://encrypted_msg \
    --encryption-context Department=IT
```

当授权约束为 EncryptionContextSubset 时，请求中的加密上下文对必须在授权约束中包含加密上下文对，不过请求也可以包括其他加密上下文对。以下授权约束要求请求中的一个加密上下文对是 "Deparment": "IT"。

```
"Constraints": {
   "EncryptionContextSubset": {
       "Department": "IT"
   }
}
```

来自被授权者委托人的以下请求将满足本示例中 EncryptionContextEqual 和 EncryptionContextSubset 授权约束的要求。

```
$ aws kms decrypt \
    --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --ciphertext-blob fileb://encrypted_msg \
    --encryption-context Department=IT
```

但是，来自被授权者委托人的以下请求将满足 EncryptionContextSubset 授权约束，但不满足 EncryptionContextEquals 授权约束。

```
$ aws kms decrypt \
    --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --ciphertext-blob fileb://encrypted_msg \
    --encryption-context Department=IT,Purpose=Test
```

Amazon 服务通常会在向其授予使用 中的 KMS 密钥的权限的授权中使用加密上下文约束。Amazon Web Services 账户例如，Amazon DynamoDB 使用类似下面的授权来获得为账户中的 DynamoDB 使用 Amazon 托管式密钥 的权限。此授权中的 EncryptionContextSubset 授权约束使授权中的权限仅在请求中的加密上下文包含 "subscriberID": "111122223333" 和 "tableName": "Services" 对时有效。此授权约束意味着，授权仅允许 DynamoDB 将指定的 KMS 密钥用于 Amazon Web Services 账户 中的特定表。

要获取此输出，请对您账户中的 DynamoDB 的 Amazon 托管式密钥 运行 ListGrants 操作。

```
$ aws kms list-grants --key-id 0987dcba-09fe-87dc-65ba-ab0987654321
```

```
{
    "Grants": [
        {
            "Operations": [
                "Decrypt",
                "Encrypt",
                "GenerateDataKey",
                "ReEncryptFrom",
                "ReEncryptTo",
                "RetireGrant",
                "DescribeKey"
            ],
            "IssuingAccount": "arn:aws:iam::111122223333:root",
            "Constraints": {
                "EncryptionContextSubset": {
                    "aws:dynamodb:tableName": "Services",
                    "aws:dynamodb:subscriberId": "111122223333"
                }
            },
            "CreationDate": 1518567315.0,
            "KeyId": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
            "GranteePrincipal": "dynamodb.us-west-2.amazonaws.com",
            "RetiringPrincipal": "dynamodb.us-west-2.amazonaws.com",
            "Name": "8276b9a6-6cf0-46f1-b2f0-7993a7f8c89a",
            "GrantId":
 "1667b97d27cf748cf05b487217dd4179526c949d14fb3903858e25193253fe59"
        }
    ]
}
```

授予 CreateGrant 权限

授权可以包括调用 CreateGrant 操作的权限。但是，当被授权者委托人获取从授权中，而不是从策略中调用 CreateGrant 的权限时，该权限将收到限制。

- 被授权者委托人只能创建允许父授权中部分或全部操作的授权。
- 他们创建的授权中的授权约束必须至少与父授权中的约束一样严格。

这些限制不适用于从策略中获取 CreateGrant 权限的委托人，尽管它们的权限可以由策略条件限制。

例如，考虑一个授权，该授权允许被授权委托人调用 GenerateDataKey、Decrypt 和 CreateGrant 操作。我们将允许 CreateGrant 权限的授权称为父授权。

```
# The original grant in a ListGrants response.
{
    "Grants": [
        {
            "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1572216195.0,
            "GrantId":
 "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a",
            "Operations": [
                "GenerateDataKey",
                "Decrypt",
                "CreateGrant
            ]
            "RetiringPrincipal": "arn:aws:iam::111122223333:role/adminRole",
            "Name": "",
            "IssuingAccount": "arn:aws:iam::111122223333:root",
            "GranteePrincipal": "arn:aws:iam::111122223333:role/keyUserRole",
            "Constraints": {
                "EncryptionContextSubset": {
                    "Department": "IT"
                }
            },
        }
    ]
}
```

被授权者委托人 exampleUser 可以使用此权限创建授权，其中包括在原始授权中指定的操作的任意子集，例如 CreateGrant 和 Decrypt。子级授权不能包含其他操作，如 ScheduleKeyDeletion 或者 ReEncrypt。

此外，子授权中的授权约束必须与父授权具有相同的限制或更严格的限制。例如，子授权可以添加对到父授权中的 EncryptionContextSubset 约束，但不能从中删除对。子授权可以将 EncryptionContextSubset 约束更改为 EncryptionContextEquals 约束，但不能反之。

IAM 最佳实践不鼓励使用具有长期凭证的 IAM 用户。而应尽可能使用提供临时凭证的 IAM 角色。有关更多信息，请参阅《IAM 用户指南》中的 IAM 安全最佳实践。

例如，被授权者委托人可以使用从父级授权那里获得的 CreateGrant 权限以创建以下子级授权。子级授权中的操作是父级授权中操作的子集，并且授权约束更严格。

```
# The child grant in a ListGrants response.
{
    "Grants": [
        {
            "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1572249600.0,
            "GrantId":
 "fedcba9999c1e2e9876abcde6e9d6c9b6a1987650000abcee009abcdef40183f",
            "Operations": [
                "CreateGrant"
                "Decrypt"
            ]
            "RetiringPrincipal": "arn:aws:iam::111122223333:user/exampleUser",
            "Name": "",
            "IssuingAccount": "arn:aws:iam::111122223333:root",
            "GranteePrincipal": "arn:aws:iam::111122223333:user/anotherUser",
            "Constraints": {
                "EncryptionContextEquals": {
                    "Department": "IT"
                }
            },
        }
    ]
}
```

子级授权中的被授权者委托人 anotherUser 可以使用它们的 CreateGrant 权限来创建授权。然而，anotherUser 创建的授权必须将操作包含在其父授权或子集中，并且授权约束必须相同或更严格。

查看授权

要查看授权，请使用 ListGrants 操作。您必须指定授权所适用的 KMS 密钥。您还可以按授权 ID 或被授权者委托人筛选授权列表。有关更多示例，请参阅ListGrants与 Amazon SDK 或 CLI 配合使用。

若要查看 Amazon Web Services 账户 和区域中具有特定停用委托人的所有授权，请使用 ListRetirableGrants。响应包括每项授权的详细信息。

> **Note**
>
> `GranteePrincipal` 响应中的 `ListGrants` 字段通常包含授权的被授权者委托人。但是，当授权中的被授权者委托人是 Amazon 服务时，`GranteePrincipal` 字段包含服务委托人，该委托人可能表示多个不同的被授权者委托人。

例如，以下命令列出 KMS 密钥的所有授权。

```
$  aws kms list-grants --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "Grants": [
        {
            "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1572216195.0,
            "GrantId":
 "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a",
            "Constraints": {
                "EncryptionContextSubset": {
                    "Department": "IT"
                }
            },
            "RetiringPrincipal": "arn:aws:iam::111122223333:role/adminRole",
            "Name": "",
            "IssuingAccount": "arn:aws:iam::111122223333:root",
            "GranteePrincipal": "arn:aws:iam::111122223333:user/exampleUser",
            "Operations": [
                "Decrypt"
            ]
        }
    ]
}
```

使用授权令牌

Amazon KMS API 采用最终一致性模型。创建授权时，授权可能不会立即生效。可能会出现短暂的延迟，才能使更改在整个 Amazon KMS 中可用。更改通常需要不到几秒钟的时间即可在整个系统中传播，但在某些情况下，可能需要几分钟。更改完全传播到整个系统后，被授权者主体可以使用授权中的权限，而无需指定授权令牌或授权的任何证据。然而，如果授权太新，尚未被所有 Amazon KMS 知晓，则请求可能会失败，并显示 `AccessDeniedException` 错误。

要立即使用新授权中的权限，请使用该授权的授权令牌。保存 CreateGrant 操作返回的授权令牌。然后为 Amazon KMS 操作提交请求中的授权令牌。您可以将授权令牌提交给任何 Amazon KMS 授权操作，且您可以在同一请求中提交多个授权令牌。

以下示例使用 CreateGrant 操作来创建允许 GenerateDataKey 和 Decrypt 操作的授权。它将保存 CreateGrant 在 token 变量中返回的授权令牌。然后，在调用 GenerateDataKey 操作时，它使用 token 变量中的授权令牌。

```
# Create a grant; save the grant token
$ token=$(aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:user/appUser \
    --retiring-principal arn:aws:iam::111122223333:user/acctAdmin \
    --operations GenerateDataKey Decrypt \
    --query GrantToken \
    --output text)

# Use the grant token in a request
$ aws kms generate-data-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --key-spec AES_256 \
    --grant-tokens $token
```

具有权限的主体也可以使用授权令牌来停用新授权，即使授权在整个 Amazon KMS 中可用之前。（RevokeGrant 操作不接受授权令牌。）有关更多信息，请参阅 停用和撤销授权。

```
# Retire the grant
$ aws kms retire-grant --grant-token $token
```

停用和撤销授权

要删除某个授权，请停用或撤销该授权。

RetireGrant 和 RevokeGrant 操作彼此非常相似。这两个操作都会删除授权，从而消除授权允许的权限。这些操作之间的主要区别在于它们是如何获得授权的。

RevokeGrant

与大多数 Amazon KMS 操作一样，对 RevokeGrant 操作的访问通过密钥策略和 IAM policy 控制。RevokeGrant API 可以由具有 kms:RevokeGrant 权限的任何委托人调用。该权限包含在授予给密钥管理员的标准权限中。通常，管理员会撤销授权以拒绝授权允许的权限。

RetireGrant

授权决定谁可以停用它。此设计使您能够控制授权的生命周期，而无需更改密钥策略或 IAM policy。通常，当您使用授权的权限时，将停用授权。

授权可以通过授权中指定的可选停用委托人停用。被授权者委托人还可以停用授权，但只有当他们也是停用委托人或者授权包括 `RetireGrant` 操作时。作为备份，在其中创建授权的 Amazon Web Services 账户 可以停用授权。

有一个可用于 IAM policy 的 `kms:RetireGrant` 权限，但它具有有限的实用工具。授权中指定的委托人无需 `kms:RetireGrant` 权限即可停用授权。单独的 `kms:RetireGrant` 权限不允许委托人停用授权。`kms:RetireGrant` 权限在密钥策略或资源控制策略中无效。

- 要拒绝停用授权的权限，您可以使用 IAM 策略中的 `kms:RetireGrant` 权限来执行 Deny 操作。
- 拥有 KMS 密钥的 Amazon Web Services 账户可以将 `kms:RetireGrant` 权限委托给该账户中的 IAM 主体。
- 如果停用主体是其他 Amazon Web Services 账户，则该其他账户中的管理员可以使用 `kms:RetireGrant` 将停用授权的权限委托给该账户中的 IAM 主体。

Amazon KMS API 采用最终一致性模型。当您创建、停用或撤销授权时，可能会出现短暂的延迟，才能使更改在整个 Amazon KMS 中可用。更改通常需要不到几秒钟的时间即可在整个系统中传播，但在某些情况下，可能需要几分钟。如果您需要在某个新授权可在整个 Amazon KMS 中使用前立即删除它，请使用授予令牌停用该授权。您不能使用授权令牌撤销授权。

的条件密钥 Amazon KMS

您可以在密钥策略和 IAM 策略中指定控制 Amazon KMS 资源访问的条件。仅当条件为 True 时，策略语句才有效。例如，您可能希望策略语句仅在特定日期后生效。或者，您可能希望策略语句根据 API 请求中是否存在特定值来控制访问。

要指定条件，请结合使用策略语句的 `Condition` 元素中的条件键与 IAM 条件运算符。有些条件键通常适用于 Amazon；另一些则特定于 Amazon KMS。

条件密钥值必须遵守密 Amazon KMS 钥策略和 IAM 策略的字符和编码规则。有关密钥策略文档规则的详细信息，请参阅 密钥策略格式。有关 IAM policy 文档规则的详细信息，请参阅《IAM 用户指南》中的 IAM 名称要求。

主题

- Amazon 全局条件键
- Amazon KMS 条件键
- Amazon KMS 经过验证的平台的条件密钥

Amazon 全局条件键

Amazon 定义全局条件密钥，这是一组策略条件密钥，适用于使用 IAM 进行访问控制的所有 Amazon 服务。 Amazon KMS 支持所有全局条件键。您可以在 Amazon KMS 密钥策略和 IAM 策略中使用它们。

例如，只有当请求中的委托人由条件键值中的 Amazon 资源名称 Amazon KMS key (ARN) 表示时，您才可以使用 aws: PrincipalArn 全局条件密钥允许访问 (KMS 密钥)。要在中支持基于属性的访问控制 (AB AC) Amazon KMS，您可以在 IAM 策略中使用 aws:ResourceTag/tag-key 全局条件密钥来允许访问带有特定标签的 KMS 密钥。

在委托人为 Amazon 服务主体的策略中，为了防止Amazon 服务被用作混淆的副手，您可以使用aws:SourceArn或aws:SourceAccount全局条件键。有关更多信息，请参阅 使用 aws:SourceArn 或 aws:SourceAccount 条件键。

有关 Amazon 全局条件密钥的信息，包括可用的请求类型，请参阅 IAM 用户指南中的Amazon 全局条件上下文密钥。有关在 IAM policy 中使用全局条件键的示例，请参阅 IAM 用户指南中的控制对请求的访问和控制标签密钥。

以下主题提供有关使用基于 IP 地址和 VPC 终端节点的条件键的特殊指导。

主题

- 在具有 Amazon KMS 权限的策略中使用 IP 地址条件
- 在具有 Amazon KMS 权限的策略中使用 VPC 终端节点条件
- 在 IAM 和 Amazon KMS 密钥策略中使用 IPv6 地址

在具有 Amazon KMS 权限的策略中使用 IP 地址条件

您可以使用 Amazon KMS 在集成 Amazon 服务中保护您的数据。但是，在允许或拒绝访问的同一策略声明中指定 IP 地址aws:SourceIp条件运算符或条件密钥时，请谨慎行事 Amazon KMS。例如，“Amazon 基于源 IP 拒绝访问” 中的Amazon策略将 Amazon 操作限制为来自指定 IP 范围的请求。

请考虑以下情况：

1. 您向 IAM 身份附加了如下所示的策略Amazon： Amazon 基于源 IP 拒绝访问。您将aws:SourceIp 条件键的值设置为该用户公司的 IP 地址范围。此 IAM 身份还附有其他政策，允许其使用 Amazon EBS EC2、Amazon 和 Amazon KMS。
2. 该身份尝试将加密的 EBS 卷附加到 EC2 实例。即使用户有权使用所有相关服务，此操作也会失败并显示授权错误。

第 2 步失败，因为解密卷加密数据密钥的请求来自与 Ama EC2 zon 基础设施关联的 IP 地址。Amazon KMS 要想成功，请求必须来自始发用户的 IP 地址。由于步骤 1 中的策略明确拒绝来自指定 IP 地址以外的 IP 地址的所有请求，因此 Amazon EC2 无法解密 EBS 卷的加密数据密钥。

此外，当请求来自 Amazon VPC 终端节点时，aws:SourceIP 条件键也不起作用。要限制对 VPC 终端节点（包括 Amazon KMS VPC 终端节点）的请求，请使用 aws:SourceVpce 或 aws:SourceVpc 条件键。有关更多信息，请参阅 Amazon VPC 用户指南 中的 VPC 终端节点 - 控制终端节点的使用。

在具有 Amazon KMS 权限的策略中使用 VPC 终端节点条件

Amazon KMS 支持由提供支持的亚马逊虚拟私有云（亚马逊 VPC）终端节点Amazon PrivateLink。当请求来自 VPC 或使用 VPC 终端节点时，您可以在密钥策略和 IAM 策略中使用以下全局条件密钥来控制对 Amazon KMS 资源的访问。有关更多信息，请参阅 使用 VPC 终端节点控制对 Amazon KMS 资源的访问。

- aws:SourceVpc 将访问限制为来自指定 VPC 的请求。
- aws:SourceVpce 将访问限制为来自指定 VPC 端点的请求。

如果您使用这些条件密钥来控制对 KMS 密钥的访问，则可能会无意中拒绝访问代表您使用的 Amazon Amazon KMS 服务。

请注意避免出现类似 IP 地址条件键示例的情况。如果您将对 KMS 密钥的请求限制在 VPC 或 VPC 终端节点上，则 Amazon KMS 从 Amazon S3 或 Amazon EBS 等集成服务对的调用可能会失败。即使源请求最终来源于 VPC 或 VPC 终端节点，也会发生这种情况。

在 IAM 和 Amazon KMS 密钥策略中使用 IPv6 地址

在尝试访问 Amazon KMS 之前 IPv6，请确保所有包含 IP 地址限制的密钥和 IAM 策略都已更新为包含 IPv6 地址范围。未更新以处理 IPv6 地址的基于 IP 的策略可能会导致客户端在开始使用时错误地丢失或获得访问权限 IPv6。有关 KMS 访问控制的一般指南，请参阅 KMS 密钥访问权限和权限。要了解 KMS 和双栈支持，请参阅双堆栈端点支持。

> **Important**
>
> 这些语句并未允许任何操作。请将这些语句与允许特定操作的其他语句结合使用。

以下语句明确拒绝来自 IPv4 地址192.0.2.*范围的请求访问所有 KMS 权限。此范围之外的所有 IP 地址均未被显式拒绝 KMS 权限。由于所有 IPv6 地址都在被拒绝的范围之外，因此此语句并未明确拒绝任何 IPv6 地址的 KMS 权限。

```
{
     "Sid": "DenyKMSPermissions",
     "Effect": "Deny",
    "Action": [
        "kms:*"
    ],
    "Resource": "*",
    "Condition": {
        "NotIpAddress": {
            "aws:SourceIp": [
                "192.0.2.0/24"
            ]
        }
    }
}
```

您可以修改Condition元素以拒绝 IPv4 (192.0.2.0/24) 和 IPv6 (2001:db8:1234::/32) 地址范围，如以下示例所示。

```
{
    "Sid": "DenyKMSPermissions",
    "Effect": "Deny",
    "Action": [
        "kms:*"
    ],
    "Resource": "*",
    "Condition": {
        "NotIpAddress": {
            "aws:SourceIp": [
                "192.0.2.0/24",
                "2001:db8:1234::/32"
            ]
        }
```

```
    }
}
```

Amazon KMS 条件键

Amazon KMS 提供了一组可在密钥策略和 IAM 策略中使用的条件密钥。这些条件键特定于 Amazon KMS。例如，在控制对对称加密 KMS 密钥的访问时，您可以使用 `kms:EncryptionContext:context-key` 条件键以请求特定的加密上下文。

API 操作请求的条件

许多 Amazon KMS 条件密钥根据 Amazon KMS 操作请求中参数的值来控制对 KMS 密钥的访问。例如，您可以在 IAM 策略中使用 k ms: KeySpec 条件密钥，仅当CreateKey请求中的KeySpec参数值为时才允许使用该CreateKey操作RSA_4096。

即使该参数未出现在请求中（例如当使用参数的默认值时），此类条件也会起作用。例如，您可以使用 k ms: KeySpec 条件密钥允许用户仅在KeySpec参数值为（默认值）时才使用该CreateKey操作。SYMMETRIC_DEFAULT此条件允许 KeySpec 参数值为 SYMMETRIC_DEFAULT 的请求，以及无 KeySpec 参数的请求。

API 操作中使用的 KMS 密钥的条件

某些 Amazon KMS 条件密钥可以根据操作中使用的 KMS 密钥的属性来控制对操作的访问。例如，您可以使用 kms: KeyOrigin 条件允许委托人仅GenerateDataKey在 KMS 密钥的密钥为时调用 KMS 密钥。Origin AWS_KMS要了解某个条件键能否以这种方式使用，请参阅条件键的说明。

该操作必须是 KMS 密钥资源操作，即为特定 KMS 密钥授权的操作。若要标识 KMS 密钥资源操作，请在操作和资源表中，在操作的 Resources 列中查找的 KMS key 的值。如果您将这种类型的条件密钥用于未经特定 KMS 密钥资源授权的操作（例如）ListKeys，则该权限无效，因为条件永远无法满足。授权 ListKeys 操作时未涉及 KMS 密钥资源，也无 KeySpec 属性。

以下主题描述了每个 Amazon KMS 条件键，并包括演示策略语法的示例策略语句。

使用带有条件键的集合运算符

当策略条件比较两组值（例如请求中的标签集和策略中的标签集）时，您需要告诉 Amazon 如何比较这两组值。为此，IAM 定义了两个集合运算符 ForAnyValue 和 ForAllValues。仅将集合运算符用于需要它们的多值条件键。不要将集合运算符用于单值条件键。像往常一样，在生产环境中使用策略语句之前，始终全面测试这些策略语句。

条件键是单值或多值。要确定 Amazon KMS 条件键是单值还是多值，请参阅条件键描述中的值类型列。

- 单值条件键在授权上下文（请求或资源）中最多具有一个值。例如，由于每个 API 调用只能来自一个 Amazon Web Services 账户，因此 k m CallerAccount s: 是单值条件密钥。不要将集合运算符用于单值条件键。
- 多值条件键在授权上下文（请求或资源）中具有多个值。例如，由于每个 KMS 密钥可以有多个别名，因此 k ms: ResourceAliases 可以有多个值。多值条件键需要一个集合运算符。

请注意，单值条件键和多值条件键之间的差异取决于授权上下文中的值数量；而不是策略条件中的值数量。

> **Warning**
>
> 将集合运算符用于单值条件键可能会创建过于宽容（或过于限制）的策略语句。仅将集合运算符用于多值条件键。
> 如果您创建或更新包含带有 kms:EncryptionContext:上下文密钥或`aws:RequestTag/`*`tag-key`*条件键的`ForAllValues`集合运算符的策略，Amazon KMS 则会返回以下错误消息：`OverlyPermissiveCondition: Using the ForAllValues set operator with a single-valued condition key matches requests without the specified [encryption context or tag] or with an unspecified [encryption context or tag]. To fix, remove ForAllValues.`

有关 `ForAnyValue` 和 `ForAllValues` 集合运算符的详细信息，请参阅 IAM 用户指南中的使用多个键和值。有关在单值条件下使用`ForAllValues`集合运算符的风险的信息，请参阅 IAM 用户指南中的安全警告- ForAllValues 使用单值密钥。

主题

- kms: BypassPolicyLockoutSafetyCheck
- kms: CallerAccount
- kms:CustomerMasterKeySpec（已弃用）
- kms:CustomerMasterKeyUsage（已弃用）
- kms: DataKeyPairSpec
- kms: EncryptionAlgorithm

- kms:EncryptionContext: 上下文密钥
- kms: EncryptionContextKeys
- kms: ExpirationModel
- kms: GrantConstraintType
- kms: GrantIsFor AWSResource
- kms: GrantOperations
- kms: GranteePrincipal
- kms: KeyAgreementAlgorithm
- kms: KeyOrigin
- kms: KeySpec
- kms: KeyUsage
- kms: MacAlgorithm
- kms: MessageType
- kms: MultiRegion
- kms: MultiRegionKeyType
- kms: PrimaryRegion
- kms: ReEncryptOnSameKey
- kms: RequestAlias
- kms: ResourceAliases
- kms: ReplicaRegion
- kms: RetiringPrincipal
- kms: RotationPeriodInDays
- kms: ScheduleKeyDeletionPendingWindowInDays
- kms: SigningAlgorithm
- kms: ValidTo
- kms: ViaService
- kms: WrappingAlgorithm
- kms: WrappingKeySpec

kms: BypassPolicyLockoutSafetyCheck

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:BypassPolicyLockoutSafetyCheck	布尔值	单值	CreateKey PutKeyPolicy	仅限 IAM policy 密钥策略和 IAM policy

kms:BypassPolicyLockoutSafetyCheck条件键根据请求中BypassPolicyLockoutSafetyCheck参数的值控制对CreateKey和PutKeyPolicy操作的访问权限。

以下示例 IAM policy 语句阻止用户绕过策略锁定安全检查，方式为当 CreateKey 请求中 BypassPolicyLockoutSafetyCheck 参数的值为 true. 时，拒绝用户创建 KMS 密钥的权限

```
{
  "Effect": "Deny",
  "Action": [
    "kms:CreateKey",
    "kms:PutKeyPolicy"
  ],
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:BypassPolicyLockoutSafetyCheck": true
    }
  }
}
```

您还可以在 IAM policy 或密钥策略中使用 kms:BypassPolicyLockoutSafetyCheck 条件键控制对 PutKeyPolicy 操作的访问权限。密钥策略中的以下示例策略语句阻止用户在更改 KMS 密钥的策略时绕过策略锁定安全检查。

此策略语句不是使用显式 Deny，而是结合使用 Allow 和 Null 条件运算符，以仅当请求不含 BypassPolicyLockoutSafetyCheck 参数时，允许访问。如果未使用此参数，则默认值为 false。此较弱的策略语句在少数有必要绕过的情况下可覆盖。

```
{
  "Effect": "Allow",
  "Action": "kms:PutKeyPolicy",
  "Resource": "*",
  "Condition": {
    "Null": {
      "kms:BypassPolicyLockoutSafetyCheck": true
    }
  }
}
```

另请参阅

- kms: KeySpec
- kms: KeyOrigin
- kms: KeyUsage

kms: CallerAccount

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:CallerAccount	字符串	单值	KMS 密钥资源操作 自定义密钥存储操作	密钥策略和 IAM 策略

您可以使用此条件键允许或拒绝对 Amazon Web Services 账户中所有身份（用户和角色）的访问。在密钥策略中，您可以使用 Principal 元素来指定策略语句所适用的身份。Principal 元素的语法未提供指定 Amazon Web Services 账户账户中的所有身份的方式。但是你可以通过将此条件键与指定所有 Amazon 身份的Principal元素结合起来来实现这种效果。

您可以使用它来控制对任何 KMS 密钥资源操作（即使用特定 KMS 密钥的任何 Amazon KMS 操作）的访问权限。若要标识 KMS 密钥资源操作，请在操作和资源表中，在操作的 Resources 列中查找的 KMS key 的值。它也适用于管理自定义密钥存储的操作。

例如，以下密钥策略语句演示了如何使用 kms:CallerAccount 条件键。本政策声明包含在 Amazon EBS Amazon 托管式密钥 的关键政策中。它将指定所有 Amazon 身份的Principal元素与kms:CallerAccount条件键相结合，以有效地允许访问 Amazon Web Services 账户 111122223333 中的所有身份。它包含一个额外的 Amazon KMS 条件密钥 (kms:ViaService)，通过仅允许通过 Amazon EBS 发出的请求来进一步限制权限。有关更多信息，请参阅 kms: ViaService。

```
{
  "Sid": "Allow access through EBS for all principals in the account that are
 authorized to use EBS",
  "Effect": "Allow",
  "Principal": {"AWS": "*"},
  "Condition": {
    "StringEquals": {
      "kms:CallerAccount": "111122223333",
      "kms:ViaService": "ec2.us-west-2.amazonaws.com"
    }
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:CreateGrant",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

kms:CustomerMasterKeySpec（已弃用）

kms:CustomerMasterKeySpec 条件键已弃用。请改用 kms: KeySpec 条件密钥。

kms:CustomerMasterKeySpec 和 kms:KeySpec 条件键的运行方式相同。只有名称不同。建议使用 kms:KeySpec。但是，为了避免破坏性更改， Amazon KMS 支持这两个条件键。

kms:CustomerMasterKeyUsage（已弃用）

kms:CustomerMasterKeyUsage 条件键已弃用。请改用 kms: KeyUsage 条件密钥。

kms:CustomerMasterKeyUsage 和 kms:KeyUsage 条件键的运行方式相同。只有名称不同。建议使用 kms:KeyUsage。但是，为了避免破坏性更改， Amazon KMS 支持这两个条件键。

kms: DataKeyPairSpec

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:DataKeyPairSpec	字符串	单值	GenerateDataKeyPair GenerateDataKeyPairWithoutPlaintext	密钥策略和 IAM policy

您可以使用此条件键根据请求中KeyPairSpec参数的值来控制对GenerateDataKeyPair和GenerateDataKeyPairWithoutPlaintext操作的访问权限。例如，您可以仅允许用户生成特定类型的数据密钥对。

以下示例密钥策略语句使用 kms:DataKeyPairSpec 条件键，以仅允许用户使用 KMS 密钥生成 RSA 数据密钥对。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": [
    "kms:GenerateDataKeyPair",
    "kms:GenerateDataKeyPairWithoutPlaintext"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:DataKeyPairSpec": "RSA*"
    }
  }
}
```

另请参阅

- kms: KeySpec

- the section called “kms: EncryptionAlgorithm”
- the section called “kms:EncryptionContext: 上下文密钥”
- the section called “kms: EncryptionContextKeys”

kms: EncryptionAlgorithm

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:EncryptionAlgorithm	字符串	单值	Decrypt Encrypt GenerateDataKey GenerateDataKeyPair GenerateDataKeyPairWithoutPlaintext GenerateDataKeyWithoutPlaintext ReEncrypt	密钥策略和 IAM 策略

您可以使用 kms:EncryptionAlgorithm 条件键，根据操作中使用的加密算法，控制对加密操作的访问。对于 “加密”、“解密” 和 “ReEncrypt操作”，它根据请求中EncryptionAlgorithm参数的值来控制访问权限。对于生成数据密钥和数据密钥对的操作，则根据用于加密数据密钥的加密算法控制访问。

此条件密钥对以外执行的操作没有影响 Amazon KMS，例如使用外部非对称 KMS 密钥对中的公钥进行加密。 Amazon KMS

EncryptionAlgorithm 请求中的参数

要允许用户仅将特定加密算法用于 KMS 密钥，请使用包含 Deny 效果和 StringNotEquals 条件运算符的策略语句。例如，以下示例密钥策略语句禁止可担任 ExampleRole 角色的主体，在指定的加密操作中使用此对称 KMS 密钥，除非请求中的加密算法为 RSAES_OAEP_SHA_256，与 RSA KMS 密钥结合使用的非对称加密算法。

与允许用户使用特定加密算法的策略语句不同，具有双重否定的策略语句（如上例），会阻止此 KMS 密钥的其他策略和授权允许此角色使用其他加密算法。此密钥策略语句中的 Deny 优先于任何包含 Allow 效果的密钥策略或 IAM policy，并且优先于此 KMS 密钥及其委托人的所有授权。

```
{
  "Sid": "Allow only one encryption algorithm with this asymmetric KMS key",
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*"
  ],
  "Resource": "*",
  "Condition": {
    "StringNotEquals": {
      "kms:EncryptionAlgorithm": "RSAES_OAEP_SHA_256"
    }
  }
}
```

用于操作的加密算法

您也可以使用 kms:EncryptionAlgorithm 条件键基于操作中使用的加密算法控制对操作的访问权限，即使在请求中未指定算法时也是如此。这使您能够要求或禁止 SYMMETRIC_DEFAULT 算法，它可能不会在请求中指定，因为它是默认值。

通过此功能，您还可以使用 kms:EncryptionAlgorithm 条件键，控制对生成数据密钥和数据密钥对的操作的访问。这些操作仅使用对称加密 KMS 密钥和 SYMMETRIC_DEFAULT 算法。

例如，此 IAM policy 限制其委托人只能使用对称加密。除非请求中指定的或操作中使用的加密算法为 SYMMETRIC_DEFAULT，否则策略将拒绝对示例帐户

中的任何 KMS 密钥进行加密操作的访问。包括GenerateDataKey*向权限添加GenerateDataKeyGenerateDataKeyWithoutPlaintextGenerateDataKeyPair、、和GenerateDataKeyPairWithoutPlaintext。条件对这些操作没有影响，因为它们始终使用对称加密算法。

```
{
  "Sid": "AllowOnlySymmetricAlgorithm",
  "Effect": "Deny",
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
  "Condition": {
    "StringNotEquals": {
      "kms:EncryptionAlgorithm": "SYMMETRIC_DEFAULT"
    }
  }
}
```

另请参阅

- the section called “kms: MacAlgorithm”
- kms: SigningAlgorithm

kms:EncryptionContext: 上下文密钥

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:EncryptionContext: *context-key*	字符串	单值	CreateGrant Encrypt Decrypt GenerateDataKey	密钥策略和 IAM policy

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
			GenerateDataKeyPair GenerateDataKeyPairWithoutPlaintext GenerateDataKeyWithoutPlaintext ReEncrypt RetireGrant	

您可以使用 kms:EncryptionContext:*context-key* 条件键，根据加密操作请求中的加密上下文，控制对对称加密 KMS 密钥的访问。使用此条件键可评估加密上下文对中的键和值。要仅评估加密上下文密钥或无论密钥或值如何，都需要加密上下文，请使用 kms: EncryptionContextKeys 条件密钥。

> **Note**
>
> 条件键值必须遵守密钥策略和 IAM policy 的字符规则。在加密上下文中有效的某些字符在策略中无效。您可能无法使用此条件键来表示全部有效的加密上下文值。有关密钥策略文档规则的详细信息，请参阅 密钥策略格式。有关 IAM policy 文档规则的详细信息，请参阅《IAM 用户指南》中的 IAM 名称要求。

您不能在使用非对称 KMS 密钥或 HMAC KMS 密钥的加密操作中指定加密上下文。非对称算法和 MAC 算法不支持加密上下文。

要使用 kms:EncryptionContext:上下文密钥条件密钥，请将*context-key*占位符替换为加密上下文密钥。使用加密上下文值替换 *context-value* 占位符。

```
"kms:EncryptionContext:context-key": "context-value"
```

例如，下面的条件键指定一个加密上下文，其键为 AppName，值为 ExampleApp (AppName = ExampleApp)。

```
"kms:EncryptionContext:AppName": "ExampleApp"
```

这是一个单值条件键。条件键中的密钥指定特定的加密上下文键 (context-key)。尽管您可以在每个 API 请求中包含多个加密上下文对，但具有指定 context-key 的加密上下文对只能有一个值。例如，kms:EncryptionContext:Department 条件键仅适用于具有 Department 密钥的加密上下文对，任何具有 Department 密钥的给定加密上下文对都只能有一个值。

不要将集合运算符用于 kms:EncryptionContext:*context-key* 条件键。如果您创建一个具有 Allow 操作，kms:EncryptionContext:*context-key* 条件键和 ForAllValues 集合运算符的策略语句，则条件允许没有加密上下文的请求以及带有未在策略条件中指定的加密上下文对的请求。

> **Warning**
>
> 请勿将 ForAnyValue 或 ForAllValues 集合运算符用于此单值条件键。这些集合运算符可以创建一个策略条件，该条件不需要您计划要求的值，并允许您计划禁止的值。
> 如果您创建或更新包含带有:conte xt-key 的ForAllValues集合运算符的策略， Amazon KMS 则会返回以下错误消息 kms:EncryptionContext：
> `OverlyPermissiveCondition:EncryptionContext: Using the ForAllValues set operator with a single-valued condition key matches requests without the specified encryption context or with an unspecified encryption context. To fix, remove ForAllValues.`

若要要求特定的加密上下文对，请结合使用 kms:EncryptionContext:*context-key* 条件键与 StringEquals 运算符。

以下示例密钥策略语句允许可以担任角色的委托人仅在请求中的加密上下文包括 AppName:ExampleApp 对时使用 GenerateDataKey 请求中的 KMS 密钥。允许使用其他加密上下文对。

密钥名称区分大小写。值是否区分大小写由条件运算符确定，例如 StringEquals。有关更多信息，请参阅 加密上下文条件区分大小写。

```
{
```

```
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:AppName": "ExampleApp"
    }
  }
}
```

要要求加密上下文对并禁止所有其他加密上下文对，请在策略声明kms:EncryptionContextKeys中同时使用 kms:EncryptionContext：上下文密钥和上下文密钥。以下密钥策略语句使用 kms:EncryptionContext:AppName 条件要求请求中的 AppName=ExampleApp 加密上下文对。它还结合使用 kms:EncryptionContextKeys 条件键与 ForAllValues 集合运算符以仅允许 AppName 加密上下文键。

ForAllValues 集合运算符将请求中的加密上下文键限制为 AppName。如果 kms:EncryptionContextKeys 条件与 ForAllValues 集合运算符在策略语句中单独使用，则此集合运算符将允许没有加密上下文的请求。但是，如果请求没有加密上下文，则 kms:EncryptionContext:AppName 条件将失败。有关 ForAllValues 集合运算符的详细信息，请参阅 IAM 用户指南中的使用多个键和值。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/KeyUsers"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:AppName": "ExampleApp"
    },
    "ForAllValues:StringEquals": {
      "kms:EncryptionContextKeys": [
        "AppName"
      ]
    }
  }
```

```
}
```

您还可以使用此条件键拒绝对特定操作的 KMS 密钥的访问。如果请求中的加密上下文包含 `Stage=Restricted` 加密上下文对，以下示例密钥策略语句使用 `Deny` 效果，以禁止委托人使用 KMS 密钥。此条件允许使用其他加密上下文对进行请求，包括具有 `Stage` 键和其他值的加密上下文对，例如 `Stage=Test`。

```
{
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Stage": "Restricted"
    }
  }
}
```

使用多个加密上下文对

您可以要求或禁止多个加密上下文对。您还可以要求几个加密上下文对之一。有关用于解释这些条件的逻辑的详细信息，请参阅 IAM 用户指南中的创建具有多个键或值的条件。

> **Note**
>
> 本主题的早期版本显示了使用`ForAnyValue`和`ForAllValues`集合运算符和 kms:EncryptionContext:conte xt-key 条件键的策略声明。将集合运算符与单值条件键结合使用可能会导致允许没有加密上下文和未指定加密上下文对的请求的策略。
> 例如，具有 `Allow` 效果、`ForAllValues` 集合运算符和 `"kms:EncryptionContext:Department": "IT"` 条件键的策略条件不会将加密上下文限制为“Department=IT”对。它允许没有加密上下文的请求和具有未指定加密上下文对的请求，例如 `Stage=Restricted`。
> 请查看您的政策，并使用上下文密钥将集合 kms:EncryptionContext运算符从任何条件中删除。尝试使用此格式创建或更新策略失败，并显示 `OverlyPermissiveCondition` 异常。要纠正此错误，请删除集合运算符。

若需要多个加密上下文对，请列出相同条件下的对。以下示例密钥策略语句需要两个加密上下文对 Department=IT 和 Project=Alpha。由于条件具有不同的键（kms:EncryptionContext:Department 和 kms:EncryptionContext:Project）它们由 AND 运算符隐式连接。允许其他加密上下文对，但它们不是必需的。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:Decrypt",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Department": "IT",
      "kms:EncryptionContext:Project": "Alpha"
    }
  }
}
```

若需要一个加密上下文对或另一个加密上下文对，请将每个条件键放在单独的策略语句中。以下示例密钥策略需要 Department=IT 或 Project=Alpha 对，或此两者。允许其他加密上下文对，但它们不是必需的。

```
{
 "Effect": "Allow",
 "Principal": {
  "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
 },
 "Action": "kms:GenerateDataKey",
 "Resource": "*",
 "Condition": {
  "StringEquals": {
   "kms:EncryptionContext:Department": "IT"
  }
 }
},
{
 "Effect": "Allow",
 "Principal": {
  "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
 },
```

```
 "Action": "kms:GenerateDataKey",
 "Resource": "*",
 "Condition": {
  "StringEquals": {
   "kms:EncryptionContext:Project": "Alpha"
  }
 }
}
```

要要求特定的加密对并排除所有其他加密上下文对，请同时使用 kms:EncryptionContext：上下文密钥和策略kms:EncryptionContextKeys声明中。以下密钥策略声明使用 kms:EncryptionContext:conte xt-key 条件来要求同时包含Department=IT和Project=Alpha对的加密上下文。它结合使用 kms:EncryptionContextKeys 条件键与 ForAllValues 集合运算符以仅允许 Department 和 Project 加密上下文键。

ForAllValues 集合运算符将请求中的加密上下文键限制为 Department 和 Project。如果在条件中单独使用它，则此集合运算符将允许没有加密上下文的请求，但是在此配置中，此条件下的 kms:EncryptionContext:conte xt-key 将失败。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Department": "IT",
      "kms:EncryptionContext:Project": "Alpha"
    },
    "ForAllValues:StringEquals": {
      "kms:EncryptionContextKeys": [
        "Department",
        "Project"
      ]
    }
  }
}
```

您还可以禁止多个加密上下文对。如果请求中的加密上下文包含 Stage=Restricted 或 Stage=Production 对，以下示例密钥策略语句使用 Deny 效果，以禁止委托人使用 KMS 密钥。

相同键 (kms:EncryptionContext:Stage) 的多个值（Restricted 和 Production）用 OR 隐式连接。有关详细信息，请参阅 IAM 用户指南中的具有多个键或值的条件的评估逻辑。

```
{
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Stage": [
         "Restricted",
         "Production"
      ]
    }
  }
}
```

加密上下文条件区分大小写

在解密操作中指定的加密上下文，与加密操作中指定的加密上下文必须是区分大小写的精确匹配。只有当加密上下文中有多个上下文对时，上下文对的顺序可以改变。

但是，在策略条件中，条件键不区分大小写。条件值是否区分大小写由您使用的策略条件运算符确定，例如 StringEquals 或 StringEqualsIgnoreCase。

因此，由 kms:EncryptionContext: 前缀和 *context-key* 替换组成的条件键不区分大小写。使用此条件的策略不检查条件键任何元素的大小写。条件值（即 *context-value* 替换）是否区分大小写由策略条件运算符确定。

例如，以下策略语句在加密上下文包含 Appname 键时允许操作，不论其大小写如何。StringEquals 条件要求 ExampleApp 按照其指定的大小写。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:Decrypt",
  "Resource": "*",
```

```
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:Appname": "ExampleApp"
    }
  }
}
```

要要求使用区分大小写的加密上下文密钥，请使用 k ms: EncryptionContextKeys 策略条件和区分大小写的条件运算符，例如。StringEquals在此策略条件中，由于加密上下文键是此策略条件中的值，它是否区分大小写由条件运算符确定。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "kms:EncryptionContextKeys": "AppName"
    }
  }
}
```

要要求对加密上下文密钥和值进行区分大小写的评估，请在同一个策略声明中同时使用 kms:EncryptionContextKeys an kms: EncryptionContext d: conte xt-key 策略条件。区分大小写的条件运算符（例如 StringEquals）始终适用于条件的值。加密上下文键（例如 AppName）是 kms:EncryptionContextKeys 条件的值。加密上下文值（例如ExampleApp）是 kms:EncryptionContext:上下文密钥条件的值。

例如，在以下示例密钥策略语句中，由于 StringEquals 运算符区分大小写，加密上下文键和加密上下文值均区分大小写。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
```

```
  "Condition": {
    "ForAnyValue:StringEquals": {
      "kms:EncryptionContextKeys": "AppName"
    },
    "StringEquals": {
      "kms:EncryptionContext:AppName": "ExampleApp"
    }
  }
}
```

在加密上下文条件中使用变量

加密上下文对中的键和值必须是简单的文本字符串。它们不能是整数或对象，也不能是任何未完全解析的类型。如果您使用其他类型，例如整数或浮点数，则将其 Amazon KMS 解释为文字字符串。

```
"encryptionContext": {
    "department": "10103.0"
}
```

不过，kms:EncryptionContext:*context-key* 条件键的值可以是 IAM policy 变量。在运行时根据请求中的值解析这些策略变量。例如，aws:CurrentTime 解析为请求的时间，aws:username 解析为调用方的友好名称。

您可以使用这些策略变量来创建一个策略语句，该语句包含的条件需要加密上下文中非常具体的信息，例如调用方的用户名。由于它包含一个变量，因此，您可以对所有能够代入该角色的用户使用相同的策略语句。您无需为每个用户编写单独的策略语句。

请考虑这样一种情况：您希望所有能够代入角色的用户都使用同一 KMS 密钥来加密和解密其数据。但是，您希望仅允许这些用户解密其加密的数据。首先，要求每个请求都 Amazon KMS 包含一个加密上下文，其中密钥是user，值是调用者的 Amazon 用户名，例如以下内容。

```
"encryptionContext": {
    "user": "bob"
}
```

然后，要强制执行此要求，您可以使用与以下示例中的策略语句类似的策略语句。此策略语句向 TestTeam 角色授予使用 KMS 密钥加密和解密数据的权限。不过，此权限仅在请求中的加密上下文包含 "user": "*<username>*" 对时有效。为了表示用户名，条件使用 aws:username 策略变量。

在评估请求时，调用方的用户名将替换条件中的变量。同样地，条件要求 "user": "bob"（对于“bob”）和 "user": "alice"（对于“alice”）的加密上下文。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/TestTeam"
  },
  "Action": [
    "kms:Decrypt",
    "kms:Encrypt"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:user": "${aws:username}"
    }
  }
}
```

您只能在 kms:EncryptionContext:*context-key* 条件键的值中使用 IAM policy 变量。不能在键中使用变量。

也可以在变量中使用提供程序特定的上下文键。这些上下文密钥唯一标识使用 Web 联合身份验证登录 Amazon 的用户。

与所有变量一样，这些变量只能用于 kms:EncryptionContext:*context-key* 策略条件，而不能用于实际加密上下文。此外，它们只能用于条件的值，而不能用于条件的键。

例如，以下键策略语句与上一个类似。不过，条件需要加密上下文，其中键为 sub，并且值唯一标识已登录 Amazon Cognito 用户池的用户。有关识别 Amazon Cognito 中的用户和角色的详细信息，请参阅 Amazon Cognito 开发人员指南中的 IAM 角色。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/TestTeam"
  },
  "Action": [
    "kms:Decrypt",
    "kms:Encrypt"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
```

```
        "kms:EncryptionContext:sub": "${cognito-identity.amazonaws.com:sub}"
      }
    }
}
```

另请参阅

- the section called “kms: EncryptionContextKeys”
- the section called “kms: GrantConstraintType”

kms: EncryptionContextKeys

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:EncryptionContextKeys	字符串（列表）	多值	CreateGrant Decrypt Encrypt GenerateDataKey GenerateDataKeyPair GenerateDataKeyPairWithoutPlaintext GenerateDataKeyWithoutPlaintext ReEncrypt	密钥策略和 IAM policy

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
			RetireGrant	

您可以使用 kms:EncryptionContextKeys 条件键，根据加密操作请求中的加密上下文，控制对对称加密 KMS 密钥的访问。使用此条件键仅评估各个加密上下文对中的键。要同时评估加密上下文中的键和值，请使用 kms:EncryptionContext:*context-key* 条件键。

您不能在使用非对称 KMS 密钥或 HMAC KMS 密钥的加密操作中指定加密上下文。非对称算法和 MAC 算法不支持加密上下文。

> **Note**
>
> 条件密钥值（包括加密上下文密钥）必须符合密 Amazon KMS 钥策略的字符和编码规则。您可能无法使用此条件键来表示所有有效的加密上下文键。有关密钥策略文档规则的详细信息，请参阅 密钥策略格式。有关 IAM policy 文档规则的详细信息，请参阅《IAM 用户指南》中的 IAM 名称要求。

这是一个多值条件键。您可以在每个 API 请求中指定多个加密上下文对。kms:EncryptionContextKeys 会将请求中的加密上下文键与策略中的加密上下文键集进行比较。要确定如何比较这些集，您必须在策略条件中提供 ForAnyValue 或 ForAllValues 集合运算符。有关集合运算符的详细信息，请参阅 IAM 用户指南中的使用多个键和值。

- ForAnyValue：请求中的至少一个加密上下文键必须匹配策略条件中的加密上下文键。允许其他加密上下文键。如果请求中没有加密上下文，则不满足此条件。
- ForAllValues：请求中的每个加密上下文键必须匹配策略条件中的加密上下文键。此集合运算符将加密上下文键限制为策略条件中的键。它不需要任何加密上下文键，但禁止未指定的加密上下文键。

以下示例密钥策略语句结合使用 kms:EncryptionContextKeys 条件键与 ForAnyValue 集合运算符。此策略语句允许对指定运算使用 KMS 密钥，但仅当请求中至少有一个加密上下文对包括 AppName 键（忽视其值）时，才允许为指定运算使用 KMS 密钥。

例如，此密钥策略语句允许包含两个加密上下文对 AppName=Helper 和 Project=Alpha 的 GenerateDataKey 请求，因为第一个加密上下文对满足此条件。只有 Project=Alpha 或没有加密上下文的请求都会失败。

由于StringEquals条件操作区分大小写，因此此策略声明要求加密上下文密钥的拼写和大小写。不过您可以使用忽略键大小写的条件运算符，例如 StringEqualsIgnoreCase。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": [
    "kms:Encrypt",
    "kms:GenerateDataKey*"
  ],
  "Resource": "*",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "kms:EncryptionContextKeys": "AppName"
    }
  }
}
```

您也可以使用 kms:EncryptionContextKeys 条件键在使用 KMS 密钥的加密操作中要求加密上下文（任何加密上下文）。

以下示例密钥策略语句结合使用 kms:EncryptionContextKeys 条件键和 Null 条件运算符，仅允许在 API 请求中的加密上下文不为 null 时允许访问 KMS 密钥。此条件不检查加密上下文的键或值。它只验证加密上下文是否存在。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": [
    "kms:Encrypt",
    "kms:GenerateDataKey*"
  ],
  "Resource": "*",
  "Condition": {
```

```
    "Null": {
      "kms:EncryptionContextKeys": false
    }
  }
}
```

另请参阅

- kms:EncryptionContext: 上下文密钥
- kms: GrantConstraintType

kms: ExpirationModel

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:ExpirationModel	字符串	单值	ImportKeyMaterial	密钥策略和 IAM policy

kms:ExpirationModel条件键根据请求中ExpirationModel参数的值控制对ImportKeyMaterial操作的访问权限。

ExpirationModel 是可选参数，用于确定导入的密钥材料是否过期。有效值为 KEY_MATERIAL_EXPIRES 和 KEY_MATERIAL_DOES_NOT_EXPIRE。默认值为 KEY_MATERIAL_EXPIRES。

到期日期和时间由ValidTo参数的值决定。除非 ExpirationModel 参数的值为 KEY_MATERIAL_DOES_NOT_EXPIRE，否则 ValidTo 参数是必需的。您也可以使用 k ms: ValidTo 条件密钥要求特定的到期日期作为访问条件。

以下示例策略语句使用 kms:ExpirationModel 条件键，以仅允许用户在请求包含 ExpirationModel 参数且其值为 KEY_MATERIAL_DOES_NOT_EXPIRE 时将密钥材料导入 KMS 密钥。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
```

```
  },
  "Action": "kms:ImportKeyMaterial",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ExpirationModel": "KEY_MATERIAL_DOES_NOT_EXPIRE"
    }
  }
}
```

您也可以使用 kms:ExpirationModel 条件键，以仅允许用户在密钥材料过期时导入密钥材料。以下示例密钥策略语句结合使用 kms:ExpirationModel 条件键和 空值条件运算符，以仅允许用户在请求没有 ExpirationModel 参数时导入密钥材料。的默认值 ExpirationModel 为KEY_MATERIAL_EXPIRES。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:ImportKeyMaterial",
  "Resource": "*",
  "Condition": {
    "Null": {
      "kms:ExpirationModel": true
    }
  }
}
```

另请参阅

- kms: ValidTo
- kms: WrappingAlgorithm
- kms: WrappingKeySpec

kms: GrantConstraintType

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:GrantConstraintType	字符串	单值	CreateGrant RetireGrant	密钥策略和 IAM policy

您可以使用此条件键根据请求中的授权约束类型来控制对CreateGrant操作的访问权限。

创建授权时，您可以选择性地指定授权约束，以仅在存在特定加密上下文时允许授予操作权限。授权约束可以是以下两种类型之一：EncryptionContextEquals 或 EncryptionContextSubset。您可以使用此条件键来检查请求中包含哪种类型。

> **Important**
>
> 不要在此字段中包含机密或敏感信息。此字段可能会以纯文本形式显示在 CloudTrail 日志和其他输出中。

以下示例密钥策略语句使用 kms:GrantConstraintType 条件键，以仅允许用户在请求中包含 EncryptionContextEquals 授权约束时创建授权。以下示例显示了密钥策略中的策略语句。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:GrantConstraintType": "EncryptionContextEquals"
    }
  }
}
```

另请参阅

- kms:EncryptionContext: 上下文密钥
- kms: EncryptionContextKeys
- kms: GrantIsFor AWSResource
- kms: GrantOperations
- kms: GranteePrincipal
- kms: RetiringPrincipal

kms: GrantIsFor AWSResource

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型

`kms:GrantIsForAWSResource`	布尔值	单值	`CreateGrant`
			`ListGrants`
			`RevokeGrant`
				密钥策略和 IAM policy

仅当与之集成的Amazon 服务代表用户 Amazon KMS调用RevokeGrant操作时 CreateGrant，才允许或拒绝对ListGrants、或操作的权限。此策略条件不允许用户直接调用这些授权操作。

以下示例密钥策略语句使用 `kms:GrantIsForAWSResource` 条件键。它允许与 Amazon KMS之集成的 Amazon 服务（例如 Amazon EBS）代表指定的委托人针对此 KMS 密钥创建授权。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:GrantIsForAWSResource": true
    }
  }
}
```

另请参阅

- kms: GrantConstraintType
- kms: GrantOperations
- kms: GranteePrincipal
- kms: RetiringPrincipal

kms: GrantOperations

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:GrantOperations	字符串	多值	CreateGrant	密钥策略和 IAM policy

您可以使用此条件密钥根据请求中的授权CreateGrant操作来控制对操作的访问权限。例如，您可以允许用户创建委托加密权限（但不委托解密权限）的授权。有关授权的更多信息，请参阅使用授权。

这是一个多值条件键。kms:GrantOperations 将比较 CreateGrant 请求中的授权操作集合与策略中的授权操作集合。要确定如何比较这些集，您必须在策略条件中提供 ForAnyValue 或 ForAllValues 集合运算符。有关集合运算符的详细信息，请参阅 IAM 用户指南中的使用多个键和值。

- ForAnyValue：请求中的至少一个授权操作必须匹配策略条件中的授权操作之一。允许其他授权操作。
- ForAllValues：请求中的每个授予操作都必须与策略条件中的授权操作相匹配。此集合运算符将授权操作限制为策略条件中指定的操作。它不需要任何授权操作，但它禁止未指定的授权操作。

  ForAllValues 当请求中没有授予操作但CreateGrant不允许时，也会返回 true。如果 Operations 参数缺失或具有 null 值，则 CreateGrant 请求失败。

以下示例密钥策略语句使用 kms:GrantOperations 条件键，以仅允许在授权操作为 Encrypt、ReEncryptTo 或此两者时创建授权。如果授权包括任何其他操作，则 CreateGrant 请求失败。

```
{
```

```
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "ForAllValues:StringEquals": {
      "kms:GrantOperations": [
        "Encrypt",
        "ReEncryptTo"
      ]
    }
  }
}
```

如果将策略条件中的集合运算符更改为 `ForAnyValue`，则策略语句将要求授权中至少有一个授权操作是 `Encrypt` 或 `ReEncryptTo`，但它允许其他授权操作，例如 `Decrypt` 或 `ReEncryptFrom`。

另请参阅

- kms: GrantConstraintType
- kms: GrantIsFor AWSResource
- kms: GranteePrincipal
- kms: RetiringPrincipal

kms: GranteePrincipal

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
`kms:GranteePrincipal`	字符串	单值	`CreateGrant`	IAM 和密钥策略

您可以使用此条件键根据请求中GranteePrincipal参数的值来控制对CreateGrant操作的访问权限。例如，您可以仅允许在 `CreateGrant` 请求中的被授权主体与条件语句中指定的主体匹配时创建使用 KMS 密钥的授权。

要指定被授权者主体，请使用 Amazon 主体的 Amazon 资源名称（ARN）。有效的委托人包括 Amazon Web Services 账户 IAM 用户、IAM 角色、联合用户和代入角色用户。有关委托人的 ARN 语法的帮助，请参阅 IAM 用户指南 ARNs中的 IAM。

以下示例密钥策略语句使用 `kms:GranteePrincipal` 条件键，以仅允许在授权中的被授权主体为 `LimitedAdminRole` 时创建 KMS 密钥授权。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:GranteePrincipal": "arn:aws:iam::111122223333:role/LimitedAdminRole"
    }
  }
}
```

另请参阅

- kms: GrantConstraintType
- kms: GrantIsFor AWSResource
- kms: GrantOperations
- kms: RetiringPrincipal

kms: KeyAgreementAlgorithm

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
`kms:KeyAgreementAlgorithm`	字符串	单值	`DeriveSharedSecret`	密钥策略和 IAM policy

您可以使用kms:KeyAgreementAlgorithm条件键根据请求中KeyAgreementAlgorithm参数的值来控制对DeriveSharedSecret操作的访问权限。对于 KeyAgreementAlgorithm，唯一的有效值为 ECDH。

例如，以下密钥策略声明使用kms:KeyAgreementAlgorithm条件密钥拒绝所有访问权限，DeriveSharedSecret 除非KeyAgreementAlgorithm是ECDH。

```
{
       "Effect": "Deny",
       "Principal": {
         "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
       },
       "Action": "kms:DeriveSharedSecret",
       "Resource": "*",
       "Condition": {
            "StringNotEquals": {
               "kms:KeyAgreementAlgorithm": "ECDH"
         }
       }
}
```

另请参阅

- the section called “kms: KeyUsage”

kms: KeyOrigin

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:KeyOrigin	字符串	单值	CreateKey KMS 密钥资源操作	IAM 策略 密钥策略和 IAM 策略

kms:KeyOrigin 条件键根据操作创建的或操作中使用的 KMS 密钥的 Origin 属性值，控制对操作的访问。它作为资源条件或请求条件工作。

您可以使用此条件键根据请求中 O rigin 参数的值来控制对CreateKey操作的访问权限。Origin 的有效值为 AWS_KMS、AWS_CLOUDHSM 和 EXTERNAL。

例如，只有在 () 中生成密钥材料、仅当密钥材料是在与自定义密钥存储库关联的 Amazon CloudHSM 集群 Amazon KMS (AWS_KMS) 中生成密钥材料时，或者仅当密钥材料是从外部来源 (AWS_CLOUDHSMEXTERNAL) 导入时，才能创建 KMS 密钥。

以下示例密钥策略声明仅在创建密钥材料时才使用kms:KeyOrigin条件密钥创 Amazon KMS 建 KMS 密钥。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
      },
      "Action": "kms:CreateKey",
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:KeyOrigin": "AWS_KMS"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
      },
      "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:GenerateDataKey",
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:GenerateDataKeyPair",
        "kms:GenerateDataKeyPairWithoutPlaintext",
        "kms:ReEncrypt*"
      ],
```

```
      "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "kms:KeyOrigin": "AWS_CLOUDHSM"
        }
      }
    }
  ]
}
```

还可以使用 kms:KeyOrigin 条件键，根据用于操作的 KMS 密钥的 Origin 属性，控制对使用或管理 KMS 密钥的操作的访问。该操作必须是 KMS 密钥资源操作，即为特定 KMS 密钥授权的操作。若要标识 KMS 密钥资源操作，请在操作和资源表中，在操作的 Resources 列中查找的 KMS key 的值。

例如，以下 IAM policy 允许委托人执行指定的 KMS 密钥资源操作，但只能使用账户中在自定义密钥存储中创建的 KMS 密钥。

```
{
  "Effect": "Allow",
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:GenerateDataKey",
    "kms:GenerateDataKeyWithoutPlaintext",
    "kms:GenerateDataKeyPair",
    "kms:GenerateDataKeyPairWithoutPlaintext",
    "kms:ReEncrypt*"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
  "Condition": {
    "StringEquals": {
      "kms:KeyOrigin": "AWS_CLOUDHSM"
    }
  }
}
```

另请参阅

- kms: BypassPolicyLockoutSafetyCheck
- kms: KeySpec

- kms: KeyUsage

kms: KeySpec

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:KeySpec	字符串	单值	CreateKey KMS 密钥资源操作	IAM 策略 密钥策略和 IAM 策略

kms:KeySpec 条件键根据操作创建的或操作中使用的 KMS 密钥的 KeySpec 属性值，控制对操作的访问。

您可以在 IAM 策略中使用此条件密钥，根据CreateKey请求中KeySpec参数的值来控制对CreateKey操作的访问权限。例如，您可以使用此条件允许用户仅创建对称加密 KMS 密钥或仅创建 HMAC KMS 密钥。

以下示例 IAM policy 语句使用 kms:KeySpec 条件键，允许主体仅创建 RSA 非对称 KMS 密钥。权限仅在请求中的 KeySpec 以 RSA_ 开头时有效。

```
{
  "Effect": "Allow",
  "Action": "kms:CreateKey",
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:KeySpec": "RSA_*"
    }
  }
}
```

还可以使用 kms:KeySpec 条件键，根据用于操作的 KMS 密钥的 KeySpec 属性，控制对使用或管理 KMS 密钥的操作的访问。该操作必须是 KMS 密钥资源操作，即为特定 KMS 密钥授权的操作。若要标识 KMS 密钥资源操作，请在操作和资源表中，在操作的 Resources 列中查找的 KMS key 的值。

例如，以下 IAM policy 允许主体执行指定的 KMS 密钥资源操作，但只能使用账户中的对称加密 KMS 密钥。

```
{
  "Effect": "Allow",
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:DescribeKey"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
  "Condition": {
    "StringEquals": {
      "kms:KeySpec": "SYMMETRIC_DEFAULT"
    }
  }
}
```

另请参阅

- kms: BypassPolicyLockoutSafetyCheck
- kms:CustomerMasterKeySpec（已弃用）
- kms: DataKeyPairSpec
- kms: KeyOrigin
- kms: KeyUsage

kms: KeyUsage

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:KeyUsage	字符串	单值	CreateKey KMS 密钥资源操作	IAM 策略 密钥策略和 IAM 策略

kms:KeyUsage 条件键根据操作创建的或操作中使用的 KMS 密钥的 KeyUsage 属性值，控制对操作的访问。

您可以使用此条件键根据请求中KeyUsage参数的值来控制对CreateKey操作的访问权限。KeyUsage 的有效值为 ENCRYPT_DECRYPT、SIGN_VERIFY、GENERATE_VERIFY_MAC 和 KEY_AGREEMENT。

例如，您可以仅在 KeyUsage 为 ENCRYPT_DECRYPT 时允许创建 KMS 密钥，或者在 KeyUsage 为 SIGN_VERIFY 时拒绝用户权限。

以下示例 IAM policy 语句使用 kms:KeyUsage 条件键，以仅允许在 KeyUsage 为 ENCRYPT_DECRYPT 时创建 KMS 密钥。

```
{
  "Effect": "Allow",
  "Action": "kms:CreateKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:KeyUsage": "ENCRYPT_DECRYPT"
    }
  }
}
```

还可以使用 kms:KeyUsage 条件键，根据操作中的 KMS 密钥的 KeyUsage 属性，控制对使用或管理 KMS 密钥的操作的访问。该操作必须是 KMS 密钥资源操作，即为特定 KMS 密钥授权的操作。若要标识 KMS 密钥资源操作，请在操作和资源表中，在操作的 Resources 列中查找的 KMS key 的值。

例如，以下 IAM policy 允许委托人执行指定的 KMS 密钥资源操作，但只能使用账户中用于签名和验证的 KMS 密钥。

```
{
  "Effect": "Allow",
  "Action": [
    "kms:CreateGrant",
    "kms:DescribeKey",
    "kms:GetPublicKey",
    "kms:ScheduleKeyDeletion"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
  "Condition": {
    "StringEquals": {
      "kms:KeyUsage": "SIGN_VERIFY"
    }
  }
}
```

另请参阅

- kms: BypassPolicyLockoutSafetyCheck
- kms:CustomerMasterKeyUsage（已弃用）
- kms: KeyOrigin
- kms: KeySpec

kms: MacAlgorithm

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:MacAlgorithm	字符串	单值	GenerateMac VerifyMac	密钥策略和 IAM policy

您可以使用kms:MacAlgorithm条件键根据请求中MacAlgorithm参数的值来控制对GenerateMac和VerifyMac操作的访问权限。

以下示例密钥策略允许可以代入 testers 角色的用户仅在请求中的 MAC 算法为 HMAC_SHA_384 或 HMAC_SHA_512 时使用 HMAC KMS 密钥生成和验证 HMAC 标签。此策略使用两个独立的策略语句，每个语句都有自己的条件。如果在单个条件语句中指定多个 MAC 算法，则该条件需要两种算法，而不是其中一种算法。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/testers"
      },
      "Action": [
        "kms:GenerateMac",
        "kms:VerifyMac"
```

```
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:MacAlgorithm": "HMAC_SHA_384"
        }
      }
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/testers"
      },
      "Action": [
        "kms:GenerateMac",
        "kms:VerifyMac"
      ],
      "Resource": "*",
      "Condition": {
        "StringEquals": {
          "kms:MacAlgorithm": "HMAC_SHA_512"
        }
      }
    }
  ]
}
```

另请参阅

- the section called “kms: EncryptionAlgorithm”
- kms: SigningAlgorithm

kms: MessageType

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:MessageType	字符串	单值	Sign Verify	密钥策略和 IAM 策略

kms:MessageType 条件键基于请求中 MessageType 参数的值控制对 Sign 和 Verify 操作的访问权限。MessageType 的有效值为 RAW 和 DIGEST。

例如，以下密钥策略语句使用 kms:MessageType 条件键，以允许使用非对称 KMS 密钥签署消息，而不是消息摘要。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:Sign",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:MessageType": "RAW"
    }
  }
}
```

另请参阅

- the section called “kms: SigningAlgorithm”

kms: MultiRegion

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:MultiRegion	布尔值	单值	CreateKey KMS 密钥资源操作	密钥策略和 IAM 策略

您可以使用此条件键以允许仅对单区域密钥或仅对多区域密钥进行操作。kms:MultiRegion条件密钥根据 KMS 密钥的MultiRegion属性值控制对 KMS 密钥的CreateKey操作和操作的访问权限。Amazon KMS 有效值为 true（多区域）或 false（单区域）。所有 KMS 密钥都具有 MultiRegion 属性。

例如，以下 IAM policy 语句使用 kms:MultiRegion 条件键，允许委托人仅创建单区域密钥。

```
{
  "Effect": "Allow",
  "Action": "kms:CreateKey",
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:MultiRegion": false
    }
  }
}
```

kms: MultiRegionKeyType

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:MultiRegionKeyType	字符串	单值	CreateKey KMS 密钥资源操作	密钥策略和 IAM 策略

您可以使用此条件键以允许仅对多区域主键或仅对多区域副本密钥执行操作。kms:MultiRegionKeyType条件密钥根据 KMS 密钥的MultiRegionKeyType属性控制对 KMS 密钥的CreateKey操作和操作的访问权限。 Amazon KMS 有效值为 PRIMARY 和 REPLICA。只有多区域密钥具有 MultiRegionKeyType 属性。

通常情况下，您可以使用 IAM policy 中的 kms:MultiRegionKeyType 条件键以控制对多个 KMS 密钥的访问。但是，由于给定的多区域密钥可以更改为主密钥或副本密钥，因此您可能希望在密钥策略中使用此条件，以便仅当特定的多区域密钥为主密钥或副本密钥时才允许操作。

IAM policy 语句使用 kms:MultiRegionKeyType 条件键，以允许委托人仅对 Amazon Web Services 账户中指定的多区域副本密钥计划和取消删除密钥。

```
{
  "Effect": "Allow",
  "Action": [
    "kms:ScheduleKeyDeletion",
```

```
    "kms:CancelKeyDeletion"
  ],
  "Resource": "arn:aws:kms:*:111122223333:key/*",
  "Condition": {
    "StringEquals": {
      "kms:MultiRegionKeyType": "REPLICA"
    }
  }
}
```

要允许或拒绝访问所有多区域密钥，您可以将这两个值或 null 值用于 kms:MultiRegionKeyType。但是，为此，建议使用 k ms: MultiRegion 条件密钥。

kms: PrimaryRegion

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:PrimaryRegion	字符串（列表）	单值	UpdatePrimaryRegion	密钥策略和 IAM policy

您可以使用此条件键来限制UpdatePrimaryRegion操作中的目标区域。它们 Amazon Web Services 区域 可以托管您的多区域主密钥。

kms:PrimaryRegion条件键根据PrimaryRegion参数的值控制对UpdatePrimaryRegion操作的访问权限。该PrimaryRegion参数指定要提升 Amazon Web Services 区域 为主密钥的多区域副本密钥。条件的值是一个或多个 Amazon Web Services 区域 名称，例如us-east-1或ap-southeast-2，或者区域名称模式，例如 eu-*

例如，以下密钥策略语句使用 kms:PrimaryRegion 条件键，以允许委托人将多区域密钥的主区域更新为四个指定的区域之一。

```
{
  "Effect": "Allow",
  "Action": "kms:UpdatePrimaryRegion",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/Developer"
  },
  "Resource": "*",
```

```
  "Condition": {
    "StringEquals": {
      "kms:PrimaryRegion": [
         "us-east-1",
         "us-west-2",
         "eu-west-3",
         "ap-southeast-2"
      ]
    }
  }
}
```

kms: ReEncryptOnSameKey

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:ReEncryptOnSameKey	布尔值	单值	ReEncrypt	密钥策略和 IAM policy

您可以使用此条件密钥来控制对ReEncrypt操作的访问权限，具体取决于请求指定的目标 KMS 密钥是否与用于原始加密的目标密钥相同。

例如，以下密钥策略语句使用 kms:ReEncryptOnSameKey 条件键，以仅允许在目标 KMS 密钥与用于原始加密的 KMS 密钥相同时进行重新加密。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:ReEncrypt*",
  "Resource": "*",
  "Condition": {
    "Bool": {
      "kms:ReEncryptOnSameKey": true
    }
  }
}
```

kms: RequestAlias

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:RequestAlias	字符串（列表）	单值	加密操作 DescribeKey GetPublicKey	密钥策略和 IAM policy

您可以使用此条件键，仅在请求使用特定别名来标识 KMS 密钥时允许操作。kms:RequestAlias 条件键基于标识请求中的 KMS 密钥的别名控制对加密操作中使用的 KMS 密钥 GetPublicKey 或 DescribeKey 的访问。（此策略条件对GenerateRandom操作没有影响，因为该操作不使用 KMS 密钥或别名。）

此条件支持中的基于属性的访问控制 (ABAC) Amazon KMS，允许您根据 KMS 密钥的标签和别名控制对 KMS 密钥的访问。您可以使用标签和别名，在不更改策略或授权的情况下，允许或拒绝对 KMS 密钥的访问权限。有关更多信息，请参阅 ABAC for Amazon KMS。

要在此策略条件下指定别名，请使用别名名称，例如 alias/project-alpha，或别名名称模式，例如 alias/*test*。您无法在此条件键的值中指定别名 ARN。

为了满足此条件，请求中的 KeyId 参数的值必须是匹配的别名名称或别名 ARN。如果请求使用不同的密钥标识符，即使标识相同的 KMS 密钥，它也不能满足条件。

例如，以下密钥策略声明允许委托人对 KMS 密钥调用GenerateDataKey操作。但是，仅当请求中的 KeyId 参数的值为 alias/finance-key 或具有该别名名称（例如 arn:aws:kms:us-west-2:111122223333:alias/finance-key）的别名 ARN 时才允许此操作。

```
{
  "Sid": "Key policy using a request alias condition",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/developer"
  },
  "Action": "kms:GenerateDataKey",
  "Resource": "*",
  "Condition": {
```

```
    "StringEquals": {
      "kms:RequestAlias": "alias/finance-key"
    }
  }
}
```

您不能使用此条件键来控制对别名操作（例如CreateAlias或）的访问权限DeleteAlias。有关控制对别名操作的访问的信息，请参阅 控制对别名的访问。

kms: ResourceAliases

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:ResourceAliases	字符串（列表）	多值	KMS 密钥资源操作	仅限 IAM policy

使用此条件键可根据与 KMS 密钥关联的别名来控制对 KMS 密钥的访问。该操作必须是 KMS 密钥资源操作，即为特定 KMS 密钥授权的操作。若要标识 KMS 密钥资源操作，请在操作和资源表中，在操作的 Resources 列中查找的 KMS key 的值。

此条件在 Amazon KMS中支持基于属性的访问控制 (ABAC)。使用 ABAC，您可以根据分配给 KMS 密钥的标签以及与 KMS 密钥关联的别名来控制对 KMS 密钥的访问。您可以使用标签和别名，在不更改策略或授权的情况下，允许或拒绝对 KMS 密钥的访问权限。有关更多信息，请参阅 ABAC for Amazon KMS。

别名在 Amazon Web Services 账户 和区域中必须是唯一的，但此条件允许您控制对同一区域中的多个 KMS 密钥（使用StringLike比较运算符）或每个账户中不同的 KMS 密钥 Amazon Web Services 区域 的访问权限。

> **Note**
>
> 仅当 KMS 密钥符合每个 KMS 密钥配额的别名时，km s: ResourceAliases 条件才有效。如果 KMS 密钥超出此配额，则由 kms:ResourceAliases 条件授权使用 KMS 密钥的委托人将被拒绝访问 KMS 密钥。

要在此策略条件下指定别名，请使用别名名称，例如 alias/project-alpha，或别名名称模式，例如 alias/*test*。您无法在此条件键的值中指定别名 ARN。要满足条件，操作中使用的 KMS 密钥必须具有指定的别名。是否或者如何在操作请求中标识 KMS 密钥并不重要。

这是一个多值条件键，用于将与 KMS 密钥关联的别名集与策略中的别名集进行比较。要确定如何比较这些集，您必须在策略条件中提供 ForAnyValue 或 ForAllValues 集合运算符。有关集合运算符的详细信息，请参阅 IAM 用户指南中的使用多个键和值。

- ForAnyValue：至少有一个与 KMS 密钥关联的别名必须与策略条件中的别名匹配。允许使用其他别名。如果 KMS 密钥没有别名，则不满足条件。
- ForAllValues：与 KMS 密钥关联的每个别名都必须与策略中的别名相匹配。此集合运算符将与 KMS 密钥关联的别名限制为策略条件中的别名。它不需要任何别名，但它会禁止未指定的别名。

例如，以下 IAM 策略声明允许委托人对指定 Amazon Web Services 账户 的 KMS 密钥中与finance-key别名关联的任何 KMS 密钥调用该GenerateDataKey操作。（受影响的 KMS 密钥的密钥策略还必须允许委托人的账户将它们用于此操作。）为了指示条件在可能与 KMS 密钥关联的很多别名中的一个为 alias/finance-key 时得到满足，条件使用 ForAnyValue 集合运算符。

由于 kms:ResourceAliases 条件基于资源，而不是请求，因此对于与 finance-key 别名关联的任何 KMS 密钥，对 GenerateDataKey 的调用将会成功，即使请求使用密钥 ID 或密钥 ARN 来标识 KMS 密钥。

```
{
  "Sid": "AliasBasedIAMPolicy",
  "Effect": "Allow",
  "Action": "kms:GenerateDataKey",
  "Resource": [
    "arn:aws:kms:*:111122223333:key/*",
    "arn:aws:kms:*:444455556666:key/*"
  ],
  "Condition": {
    "ForAnyValue:StringEquals": {
      "kms:ResourceAliases": "alias/finance-key"
    }
  }
}
```

以下示例 IAM policy 语句允许委托人启用和禁用 KMS 密钥，但仅当 KMS 密钥的所有别名都包含“Test”时。此策略语句使用两个条件。带有 ForAllValues 集合运算符的条件要求与 KMS 密钥关联的所有别名都包括“Test”。带有 ForAnyValue 集合运算符的条件要求 KMS 密钥至少具有一个带

有“Test”的别名。如果不使用 ForAnyValue 条件，此策略语句将允许委托人使用没有别名的 KMS 密钥。

```
{
  "Sid": "AliasBasedIAMPolicy",
  "Effect": "Allow",
  "Action": [
    "kms:EnableKey",
    "kms:DisableKey"
  ],
  "Resource": "arn:aws:kms:*:111122223333:key/*",
  "Condition": {
    "ForAllValues:StringLike": {
      "kms:ResourceAliases": [
        "alias/*Test*"
      ]
    },
    "ForAnyValue:StringLike": {
      "kms:ResourceAliases": [
        "alias/*Test*"
      ]
    }
  }
}
```

kms: ReplicaRegion

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:ReplicaRegion	字符串（列表）	单值	ReplicateKey	密钥策略和 IAM policy

您可以使用此条件密钥来限制委托人 Amazon Web Services 区域 可以复制多区域密钥的范围。kms:ReplicaRegion条件键根据请求中ReplicaRegion参数的值控制对ReplicateKey操作的访问权限。此参数为新的副本密钥指定 Amazon Web Services 区域 。

条件的值是一个或多个 Amazon Web Services 区域 名称，例如us-east-1或ap-southeast-2，或者名称模式，例如eu-*。有关 Amazon KMS 支持的名称列表 Amazon Web Services 区域 ，请参阅中的Amazon Key Management Service 终端节点和配额 Amazon Web Services 一般参考。

例如，以下密钥策略声明使用kms:ReplicaRegion条件密钥，仅当ReplicaRegion参数的值为指定区域之一时，委托人才能调用该ReplicateKey操作。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/Administrator"
  },
  "Action": "kms:ReplicateKey"
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ReplicaRegion": [
         "us-east-1",
         "eu-west-3",
         "ap-southeast-2"
      ]
    }
  }
}
```

此条件键仅控制对ReplicateKey操作的访问权限。要控制对UpdatePrimaryRegion操作的访问权限，请使用 k ms: PrimaryRegion 条件密钥。

kms: RetiringPrincipal

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:RetiringPrincipal	字符串（列表）	单值	CreateGrant	密钥策略和 IAM policy

您可以使用此条件键根据请求中RetiringPrincipal参数的值来控制对CreateGrant操作的访问权限。例如，您可以仅在 CreateGrant 请求中的 RetiringPrincipal 与条件语句中的 RetiringPrincipal 匹配时，允许创建使用 KMS 密钥的授权。

要指定停用主体，请使用 Amazon 主体的 Amazon 资源名称（ARN）。有效的委托人包括 Amazon Web Services 账户 IAM 用户、IAM 角色、联合用户和代入角色用户。有关委托人的 ARN 语法的帮助，请参阅 IAM 用户指南 ARNs中的 IAM。

以下示例密钥政策语句允许用户为 KMS 密钥创建授权。kms:RetiringPrincipal 条件键将权限限制为 CreateGrant 请求，其中授权的停用主体是 LimitedAdminRole。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:RetiringPrincipal": "arn:aws:iam::111122223333:role/LimitedAdminRole"
    }
  }
}
```

另请参阅

- kms: GrantConstraintType
- kms: GrantIsFor AWSResource
- kms: GrantOperations
- kms: GranteePrincipal

kms: RotationPeriodInDays

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:RotationPeriodInDays	数值	单值	EnableKeyRotation	密钥策略和 IAM policy

您可以使用此条件键来限制委托人可以在EnableKeyRotation请求RotationPeriodInDays参数中指定的值。

RotationPeriodInDays指定每个自动密钥轮换日期之间的天数。 Amazon KMS 允许您指定介于 90 到 2560 天之间的轮换周期，但您可以使用kms:RotationPeriodInDays条件键进一步限制轮换周期，例如在有效范围内强制规定最短轮换周期。

例如，以下密钥政策语句使用 kms:RotationPeriodInDays 条件键，以防止主体在轮换时间小于或等于 180 天时启用密钥轮换。

```
{
  "Effect": "Deny",
  "Action": "kms:EnableKeyRotation",
  "Principal": "*",
  "Resource": "*",
  "Condition" : {
      "NumericLessThanEquals" : {
        "kms:RotationPeriodInDays" : "180"
      }
  }
}
```

kms: ScheduleKeyDeletionPendingWindowInDays

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:ScheduleKeyDeletionPendingWindowInDays	数值	单值	ScheduleKeyDeletion	密钥策略和 IAM policy

您可以使用此条件键来限制委托人可以在ScheduleKeyDeletion请求PendingWindowInDays参数中指定的值。

PendingWindowInDays指定删除密钥之前 Amazon KMS 要等待的天数。 Amazon KMS 允许您指定 7 到 30 天之间的等待期，但您可以使用kms:ScheduleKeyDeletionPendingWindowInDays条件键进一步限制等待时间，例如在有效范围内强制规定最短等待时间。

例如，以下密钥政策语句使用 kms:ScheduleKeyDeletionPendingWindowInDays 条件键，以防止主体安排在等待时间小于或等于 21 天时删除密钥。

```
{
  "Effect": "Deny",
  "Action": "kms:ScheduleKeyDeletion",
  "Principal": "*",
  "Resource": "*",
  "Condition" : {
      "NumericLessThanEquals" : {
        "kms:ScheduleKeyDeletionPendingWindowInDays" : "21"
      }
  }
}
```

kms: SigningAlgorithm

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:SigningAlgorithm	字符串	单值	Sign Verify	密钥策略和 IAM policy

您可以使用kms:SigningAlgorithm条件键根据请求中SigningAlgorithm参数的值来控制对 “签名” 和 “验证” 操作的访问权限。此条件密钥对以外执行的操作没有影响 Amazon KMS，例如使用外部非对称 KMS 密钥对中的公钥验证签名。 Amazon KMS

以下示例密钥策略允许可担任 testers 角色的用户，仅当用于请求的签名算法为 RSASSA_PSS 算法（如 RSASSA_PSS_SHA512）时，才能使用 KMS 密钥签署消息。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/testers"
  },
  "Action": "kms:Sign",
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:SigningAlgorithm": "RSASSA_PSS*"
    }
```

```
  }
}
```

另请参阅

- kms: EncryptionAlgorithm
- the section called “kms: MacAlgorithm”
- the section called “kms: MessageType”

kms: ValidTo

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:ValidTo	Timestamp	单值	ImportKeyMaterial	密钥策略和 IAM policy

kms:ValidTo条件密钥根据请求中ValidTo参数的值控制对ImportKeyMaterial操作的访问权限，该值决定了导入的密钥材料何时过期。此值用 Unix 时间表示。

默认情况下，ImportKeyMaterial 请求中需要 ValidTo 参数。但是，如果ExpirationModel参数的值为KEY_MATERIAL_DOES_NOT_EXPIRE，则该ValidTo参数无效。您也可以使用 kms: ExpirationModel 条件键来要求ExpirationModel参数或特定的参数值。

以下示例策略语句允许用户在 KMS 密钥中导入密钥材料。kms:ValidTo 条件键将权限限制为 ImportKeyMaterial 请求，其中 ValidTo 值小于或等于 1546257599.0（2018 年 12 月 31 日晚上 11:59:59）。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:ImportKeyMaterial",
  "Resource": "*",
  "Condition": {
    "NumericLessThanEquals": {
      "kms:ValidTo": "1546257599.0"
```

```
    }
  }
}
```

另请参阅

- kms: ExpirationModel
- kms: WrappingAlgorithm
- kms: WrappingKeySpec

kms: ViaService

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:ViaService	字符串	单值	KMS 密钥资源操作	密钥策略和 IAM policy

kms:ViaService条件密钥将 KMS 密钥的使用限制为来自指定的请求 Amazon Web Services 服务。此条件密钥仅适用于转发访问会话。您可以在每个 kms:ViaService 条件键中指定一个或多个服务。该操作必须是 KMS 密钥资源操作，即为特定 KMS 密钥授权的操作。若要标识 KMS 密钥资源操作，请在操作和资源表中，在操作的 Resources 列中查找的 KMS key 的值。

例如，以下密钥政策声明使用kms:ViaService条件密钥，仅当请求来自美国西部（俄勒冈）地区的 Amazon EC2 或 Amazon RDS 代表时，才允许将客户托管密钥用于指定操作ExampleRole。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:CreateGrant",
    "kms:ListGrants",
```

```
    "kms:DescribeKey"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ViaService": [
        "ec2.us-west-2.amazonaws.com",
        "rds.us-west-2.amazonaws.com"
      ]
    }
  }
}
```

您也可以使用 kms:ViaService 条件键以在请求来自特定服务时拒绝使用 KMS 密钥的权限。例如，密钥策略中的以下策略语句使用 kms:ViaService 条件键以防止在代表 ExampleRole 发出来自 Amazon Lambda 的请求时将客户托管密钥用于 Encrypt 操作。

```
{
  "Effect": "Deny",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": [
    "kms:Encrypt"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ViaService": [
          "lambda.us-west-2.amazonaws.com"
      ]
    }
  }
}
```

Important

在使用 kms:ViaService 条件键时，服务代表 Amazon Web Services 账户中的委托人发出请求。这些委托人必须具有以下权限：

- 使用 KMS 密钥的权限。委托人需要向集成服务授予这些权限，这样此服务才能代表委托人使用客户托管的密钥。有关更多信息，请参阅 对 Amazon 服务使用 Amazon KMS 加密。

- 使用集成服务的权限。有关向用户提供与集成的 Amazon 服务的访问权限的详细信息 Amazon KMS，请参阅集成服务的文档。

所有 Amazon 托管式密钥 都使用其密钥策略文档中的 kms:ViaService 条件键。此条件允许 KMS 密钥仅用于来自创建 KMS 密钥的服务的请求。要查看的密钥策略 Amazon 托管式密钥，请使用GetKeyPolicy操作。

kms:ViaService 条件键在 IAM 和密钥策略语句中有效。您指定的服务必须与 Amazon KMS集成并支持 kms:ViaService 条件键。

支持 **kms:ViaService** 条件键的服务

下表列出了 Amazon 与客户托管密钥集成 Amazon KMS 并支持在客户托管密钥中使用kms:ViaService条件密钥的服务。此表中的服务可能并非在所有地区都可用。在所有 Amazon 分区中使用 Amazon KMS ViaService 名称的.amazonaws.com后缀。

> **Note**
>
> 您可能需要水平或垂直滚动才能查看此表中的所有数据。

服务名称	Amazon KMS ViaService 名字
Amazon AI 操作	aiops.*AWS_region* .amazonaws.com
Amazon App Runner	apprunner. *AWS_region* .amazonaws.com
Amazon AppFabric	appfabric. *AWS_region* .amazonaws.com
Amazon AppFlow	appflow.*AWS_region* .amazonaws.com
Amazon Application Migration Service	mgn.*AWS_region* .amazonaws.com
Amazon Athena	athena.*AWS_region* .amazonaws.com

服务名称	Amazon KMS ViaService 名字
Amazon Audit Manager	auditmanager. *AWS_region* .amazonaws.com
Amazon Aurora	rds.*AWS_region* .amazonaws.com
Amazon Backup	backup.*AWS_region* .amazonaws.com
Amazon Backup 网关	backup-gateway. *AWS_region* .amazonaws.com
Amazon Bedrock Model Copy	bedrock.*AWS_region* .amazonaws.com
Amazon Chime SDK	chimevoiceconnector. *AWS_region* .amazonaws.com
Amazon Clean Rooms ML	cleanrooms-ml. *AWS_region* .amazonaws.com
Amazon CodeArtifact	codeartifact. *AWS_region* .amazonaws.com
Amazon CodeGuru Reviewer	codeguru-reviewer. *AWS_region* .amazonaws.com
Amazon Comprehend	comprehend. *AWS_region* .amazonaws.com
Amazon Connect	connect.*AWS_region* .amazonaws.com
Amazon Connect Customer Profiles	profile.*AWS_region* .amazonaws.com
Amazon Q in Connect	wisdom.*AWS_region* .amazonaws.com
Amazon Database Migration Service (Amazon DMS)	dms.*AWS_region* .amazonaws.com
Amazon DeepRacer	deepracer. *AWS_region* .amazonaws.com

服务名称	Amazon KMS ViaService 名字
Amazon Directory Service	directoryservice. *AWS_region* .amazonaws.com
Amazon DocumentDB	docdb-elastic. *AWS_region* .amazonaws.com
Amazon DynamoDB	dynamodb. *AWS_region* .amazonaws.com
亚马逊 S EC2 ystems Manager (SSM)	ssm.*AWS_region* .amazonaws.com
Amazon Elastic Block Store（Amazon EBS）	ec2.*AWS_region* .amazonaws.com （仅限 EBS）
Amazon Elastic Container Registry (Amazon ECR)	ecr.*AWS_region* .amazonaws.com
Amazon Elastic File System (Amazon EFS)	elasticfilesystem. *AWS_region* .amazonaws.com
Amazon ElastiCache	在条件键值中包含两个 ViaService 名称： • elasticache. *AWS_region* .amazonaws.com • dax.*AWS_region* .amazonaws.com
AWS Elemental MediaTailor	mediatailor. *AWS_region* .amazonaws.com
Amazon 实体分辨率	entityresolution. *AWS_region* .amazonaws.com
Amazon EventBridge	events.*AWS_region* .amazonaws.com
Amazon FinSpace	finspace. *AWS_region* .amazonaws.com

服务名称	Amazon KMS ViaService 名字
Amazon Forecast	forecast. *AWS_region* .amazonaws.com
Amazon FSx	fsx.*AWS_region* .amazonaws.com
Amazon Glue	glue.*AWS_region* .amazonaws.com
Amazon Ground Station	groundstation. *AWS_region* .amazonaws.com
Amazon GuardDuty	malware-protection. *AWS_region* .amazonaws.com
Amazon HealthLake	healthlake. *AWS_region* .amazonaws.com
Amazon IoT SiteWise	iotsitewise. *AWS_region* .amazonaws.com
Amazon Kendra	kendra.*AWS_region* .amazonaws.com
Amazon Keyspaces (for Apache Cassandra)	cassandra. *AWS_region* .amazonaws.com
Amazon Kinesis	kinesis.*AWS_region* .amazonaws.com
Amazon Data Firehose	firehose. *AWS_region* .amazonaws.com
Amazon Kinesis Video Streams	kinesisvideo. *AWS_region* .amazonaws.com
Amazon Lambda	lambda.*AWS_region* .amazonaws.com
Amazon Lex	lex.*AWS_region* .amazonaws.com
Amazon License Manager	license-manager. *AWS_region* .amazonaws.com

服务名称	Amazon KMS ViaService 名字
Amazon Location Service	geo.*AWS_region* .amazonaws.com
Amazon Lookout for Equipment	lookoutequipment. *AWS_region* .amazonaws.com
Amazon Lookout for Metrics	lookoutmetrics. *AWS_region* .amazonaws.com
Amazon Lookout for Vision	lookoutvision. *AWS_region* .amazonaws.com
Amazon Macie	macie.*AWS_region* .amazonaws.com
Amazon Mainframe Modernization	m2.*AWS_region* .amazonaws.com
Amazon Mainframe Modernization 应用程序测试	apptest.*AWS_region* .amazonaws.com
Amazon Managed Blockchain	managedblockchain. *AWS_region* .amazonaws.com
Amazon Managed Streaming for Apache Kafka (Amazon MSK)	kafka.*AWS_region* .amazonaws.com
Amazon Managed Workflows for Apache Airflow (MWAA)	airflow.*AWS_region* .amazonaws.com
Amazon MemoryDB	memorydb. *AWS_region* .amazonaws.com
Amazon Monitron	monitron. *AWS_region* .amazonaws.com
Amazon MQ	mq.*AWS_region* .amazonaws.com
Amazon Neptune	rds.*AWS_region* .amazonaws.com
Amazon Nimble Studio	nimble.*AWS_region* .amazonaws.com

服务名称	Amazon KMS ViaService 名字
Amazon HealthOmics	omics.*AWS_region* .amazonaws.com
亚马逊 OpenSearch 服务	es.*AWS_region* .amazonaws.com , aoss.*AWS_region* .amazonaws.com
Amazon OpenSearch 定制套餐	custom-packages. *AWS_region* .amazonaws.com
Amazon Proton	proton.*AWS_region* .amazonaws.com
Amazon Quantum Ledger Database (Amazon QLDB)	qldb.*AWS_region* .amazonaws.com
Amazon RDS 性能详情	rds.*AWS_region* .amazonaws.com
Amazon Redshift	redshift. *AWS_region* .amazonaws.com
Amazon Redshift 查询编辑器 V2	sqlworkbench. *AWS_region* .amazonaws.com
Amazon Redshift Serverless	redshift-serverless. *AWS_region* .amazonaws.com
Amazon Rekognition	rekognition. *AWS_region* .amazonaws.com
Amazon Relational Database Service (Amazon RDS)	rds.*AWS_region* .amazonaws.com
Amazon 复制的数据存储	ards.*AWS_region* .amazonaws.com
亚马逊 SageMaker AI	sagemaker. *AWS_region* .amazonaws.com
Amazon Secrets Manager	secretsmanager. *AWS_region* .amazonaws.com

服务名称	Amazon KMS ViaService 名字
Amazon Security Lake	securitylake.*AWS_region*.amazonaws.com
Amazon Simple Email Service（Amazon SES）	ses.*AWS_region*.amazonaws.com
Amazon Simple Notification Service（Amazon SNS）	sns.*AWS_region*.amazonaws.com
Amazon Simple Queue Service(Amazon SQS)	sqs.*AWS_region*.amazonaws.com
Amazon Simple Storage Service（Amazon S3）	s3.*AWS_region*.amazonaws.com
Amazon S3 表	s3tables.*AWS_region*.amazonaws.com
Amazon Snowball Edge	importexport.*AWS_region*.amazonaws.com
Amazon Step Functions	states.*AWS_region*.amazonaws.com
Amazon Storage Gateway	storagegateway.*AWS_region*.amazonaws.com
Amazon Systems Manager Incident Manager	ssm-incidents.*AWS_region*.amazonaws.com
Amazon Systems Manager Incident Manager 联系人	ssm-contacts.*AWS_region*.amazonaws.com
Amazon Timestream	timestream.*AWS_region*.amazonaws.com
Amazon Translate	translate.*AWS_region*.amazonaws.com

服务名称	Amazon KMS ViaService 名字
Amazon Verified Access	`verified-access.` *`AWS_region`* `.amazonaws.com`
Amazon WorkMail	`workmail.` *`AWS_region`* `.amazonaws.com`
Amazon WorkSpaces	`workspaces.` *`AWS_region`* `.amazonaws.com`
Amazon WorkSpaces 瘦客户机	`thinclient.` *`AWS_region`* `.amazonaws.com`
Amazon WorkSpaces Web	`workspaces-web.` *`AWS_region`* `.amazonaws.com`
Amazon X-Ray	`xray.`*`AWS_region`* `.amazonaws.com`

kms: WrappingAlgorithm

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
`kms:WrappingAlgorithm`	字符串	单值	`GetParametersForImport`	密钥策略和 IAM policy

此条件键根据请求中WrappingAlgorithm参数的值控制对GetParametersForImport操作的访问权限。您可以使用此条件要求委托人在导入过程中使用特定算法来加密密钥材料。当对所需公有密钥和导入令牌的请求指定不同的包装算法时，它们会失败。

以下示例密钥策略语句使用 `kms:WrappingAlgorithm` 条件键为示例用户提供调用 `GetParametersForImport` 操作的权限，但阻止他们使用 `RSAES_OAEP_SHA_1` 包装算法。当 `GetParametersForImport` 请求中的 `WrappingAlgorithm` 是 `RSAES_OAEP_SHA_1` 时，操作会失败。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:GetParametersForImport",
  "Resource": "*",
  "Condition": {
    "StringNotEquals": {
      "kms:WrappingAlgorithm": "RSAES_OAEP_SHA_1"
    }
  }
}
```

另请参阅

- kms: ExpirationModel
- kms: ValidTo
- kms: WrappingKeySpec

kms: WrappingKeySpec

Amazon KMS 条件键	条件类型	值类型	API 操作	策略类型
kms:WrappingKeySpec	字符串	单值	GetParametersForImport	密钥策略和 IAM policy

此条件键根据请求中WrappingKeySpec参数的值控制对GetParametersForImport操作的访问权限。您可以使用此条件，要求委托人在导入过程中使用特定的公有密钥类型。如果请求指定了不同密钥类型，它会失败。

由于 WrappingKeySpec 参数值的唯一有效值为 RSA_2048，阻止用户使用此值将有效阻止它们使用 GetParametersForImport 操作。

以下示例策略语句使用 kms:WrappingAlgorithm 条件键要求请求中的 WrappingKeySpec 为 RSA_4096。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/ExampleRole"
  },
  "Action": "kms:GetParametersForImport",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:WrappingKeySpec": "RSA_4096"
    }
  }
}
```

另请参阅

- kms: ExpirationModel
- kms: ValidTo
- kms: WrappingAlgorithm

Amazon KMS 经过验证的平台的条件密钥

Amazon KMS 提供条件密钥以支持 Nitro Enclaves 和 Amazon Nitro TPM 的加密认证。 Amazon Nitro Enclaves 是 Amazon 的一项 EC2 功能，它允许您创建称为飞地的隔离计算环境，以保护和处理高度敏感的数据。NitroTPM 将类似的认证功能扩展到实例。 EC2

当您使用已签名的证明文档调用 Decrypt DeriveSharedSecretGenerateDataKeyGenerateDataKeyPair、、、或 GenerateRandomAPI 操作时，这些操作会使用认证文档中的公钥对响应中的明文 APIs 进行加密，并返回密文而不是纯文本。此加密文字只能使用 Enclave 中的私有密钥进行解密。有关更多信息，请参阅 Amazon KMS 中的密码认证支持。

> **Note**
>
> 如果您在创建密钥时未提供密钥策略，请为您 Amazon 创建一个 Amazon KMS 密钥策略。此默认密钥策略将为拥有 KMS 密钥的 Amazon Web Services 账户 授予对该密钥的完全访问权限，并允许该账户使用 IAM 策略来允许访问该密钥。该策略将允许所有操作，例如

Decrypt。 Amazon 建议将 最低权限许可 的主体应用于您的 KMS 密钥策略。您也可以通过将 KMS 密钥策略操作 kms:* 修改为 NotAction:kms:Decrypt，从而限制访问权限。

通过以下条件键，您可以根据签名证明文档的内容限制这些操作的权限。在允许操作之前，请将证明文档与这些 Amazon KMS 条件键中的值进行 Amazon KMS 比较。

Nitro Enclaves 的条件键

以下条件键是特定于 Nitro Enclaves 认证：

kms:RecipientAttestation: ImageSha 384

Amazon KMS 条件密钥	条件类型	值类型	API 操作	策略类型
kms:RecipientAttestation:ImageSha384	字符串	单值	Decrypt DeriveSharedSecret GenerateDataKey GenerateDataKeyPair GenerateRandom	密钥策略和 IAM policy

当请求中的已签名证明文档中的映像摘要与条件键中的值相匹配时，kms:RecipientAttestation:ImageSha384 条件键使用 KMS 密钥控制对 Decrypt、DeriveSharedSecret、GenerateDataKey、GenerateDataKeyPair 和 GenerateRandom 的访问。ImageSha384 值对应于证明文档中的 PCR0。仅当请求中的Recipient参数为 Amazon Nitro 飞地指定了已签名的认证文档时，此条件密钥才有效。

此值也包含在请求获得 Nitro 飞地CloudTrail Amazon KMS 的事件中。

例如，以下密钥策略声明允许该data-processing角色使用 KMS 密钥进行解密、、DeriveSharedSecretGenerateDataKeyGenerateDataKeyPair、和GenerateRandom操作。kms:RecipientAttestation:ImageSha384 条件键仅允许在请求中的证明文档的映像摘要值（PCR0）与条件中的映像摘要值匹配时执行操作。仅当请求中的Recipient参数为 Amazon Nitro 飞地指定了已签名的认证文档时，此条件密钥才有效。

如果请求中不包括来自 Amazon Nitro 飞地的有效证明文件，则会因为不满足此条件而拒绝许可。

```
{
  "Sid" : "Enable enclave data processing",
  "Effect" : "Allow",
  "Principal" : {
    "AWS" : "arn:aws:iam::111122223333:role/data-processing"
  },
  "Action": [
    "kms:Decrypt",
    "kms:DeriveSharedSecret",
    "kms:GenerateDataKey",
    "kms:GenerateDataKeyPair",
    "kms:GenerateRandom"
  ],
  "Resource" : "*",
  "Condition": {
    "StringEqualsIgnoreCase": {
      "kms:RecipientAttestation:ImageSha384":
 "9fedcba8abcdef7abcdef6abcdef5abcdef4abcdef3abcdef2abcdef1abcdef1abcdef0abcdef1abcdef2abcdef3a
    }
  }
}
```

kms:: RecipientAttestation 聚合酶链反应 <PCR_ID>

Amazon KMS 条件密钥	条件类型	值类型	API 操作	策略类型
kms:RecipientAttestation:PCR<PCR_ID>	字符串	单值	Decrypt DeriveSharedSecret	密钥策略和 IAM policy

Amazon KMS 条件密钥	条件类型	值类型	API 操作	策略类型
			GenerateDataKey GenerateDataKeyPair GenerateRandom	

只有当平台配置从请求中的已签名证明文档中注册的 (PCRs) GenerateRandom 与kms:RecipientAttestation:PCR<PCR_ID>条件密钥中的匹配时，条件密钥才使用KMS 密钥控制对、、、和 PCRs 的访问。Decrypt DeriveSharedSecret GenerateDataKey GenerateDataKeyPair仅当请求中的Recipient参数指定来自 Amazon Nitro 飞地的已签名认证文档时，此条件密钥才有效。

此值也包含在代表对 Nitro 飞地 Amazon KMS 的请求CloudTrail的事件中。

要指定 PCR 值，请使用以下格式。将 PCR ID 连接到条件键名称。您可以指定标识六个飞地指标之一的 PCR ID，也可以指定您为特定使用案例定义的自定义 PCR ID。PCR 值必须是最多 96 个字节的小写十六进制字符串。

```
"kms:RecipientAttestation:PCRPCR_ID": "PCR_value"
```

例如，以下条件键为指定了一个特定的值 PCR1，该值对应于用于安全区和引导过程的内核哈希值。

```
kms:RecipientAttestation:PCR1:
 "abc1abcdef2abcdef3abcdef4abcdef5abcdef6abcdef7abcdef8abcdef9abcdef8abcdef7abcdef6abcdef5abcde
```

以下示例密钥政策语句允许 data-processing 角色将 KMS 密钥用于 Decrypt 操作。

只有当请求中已签名的证明文档中的值与kms:RecipientAttestation:PCR条件中的 PCR1 kms:RecipientAttestation:PCR1值相匹配时，此语句中的条件键才允许该操作。使用 StringEqualsIgnoreCase 策略运算符来要求对 PCR 值进行不区分大小写的比较。

如果请求不包含证明文档，则权限将被拒绝，因为不满足此条件。

```
{
  "Sid" : "Enable enclave data processing",
  "Effect" : "Allow",
  "Principal" : {
    "AWS" : "arn:aws:iam::111122223333:role/data-processing"
  },
  "Action": "kms:Decrypt",
  "Resource" : "*",
  "Condition": {
    "StringEqualsIgnoreCase": {
      "kms:RecipientAttestation:PCR1":
 "abc1de4f2dcf774f6e3b679f62e5f120065b2e408dcea327bd1c9dddaea6664e7af79355814748447674530 82c6f1
    }
  }
}
```

NitroTPM 的条件键

以下条件键是特定于 NitroTPM 认证：

kms:: RecipientAttestation nitrotpmPCR <PCR_ID>

Amazon KMS 条件密钥	条件类型	值类型	API 操作	策略类型
kms:RecipientAttestation:NitroTPMPCR <PCR_ID>	字符串	单值	Decrypt DeriveSharedSecret GenerateDataKey GenerateDataKeyPair GenerateRandom	密钥策略和 IAM policy

只有当平台配置从请求中的已签名证明文档中注册的 (PCRs) GenerateRandom 与kms:RecipientAttestation:NitroTPMPCR<PCR_ID>条件密钥中的匹配时，条件密钥才使用KMS密钥控制对、、、和 PCRs 的访问。Decrypt DeriveSharedSecret GenerateDataKey GenerateDataKeyPair仅当请求中的 Recipient 参数指定来自 NitroTPM 的已签名认证文档时，该条件键才有效。

此值也包含在代表对 NitrotPM Amazon KMS 的请求CloudTrail的事件中。

要指定 PCR 值，请使用以下格式。将 PCR ID 连接到条件键名称。PCR 值必须是最多 96 个字节的小写十六进制字符串。

```
"kms:RecipientAttestation:NitroTPMPCRPCR_ID": "PCR_value"
```

例如，以下条件键为指定了特定的值 PCR4：

```
kms:RecipientAttestation:NitroTPMPCR4:
 "abc1abcdef2abcdef3abcdef4abcdef5abcdef6abcdef7abcdef8abcdef9abcdef8abcdef7abcdef6abcdef5abcde
```

以下示例密钥政策语句允许 data-processing 角色将 KMS 密钥用于 Decrypt 操作。

只有当请求中已签名的证明文档中的值与kms:RecipientAttestation:NitroTPMPCR条件中的 PCR4 kms:RecipientAttestation:NitroTPMPCR4值相匹配时，此语句中的条件键才允许该操作。使用 StringEqualsIgnoreCase 策略运算符来要求对 PCR 值进行不区分大小写的比较。

如果请求不包含证明文档，则权限将被拒绝，因为不满足此条件。

```
{
  "Sid" : "Enable NitroTPM data processing",
  "Effect" : "Allow",
  "Principal" : {
    "AWS" : "arn:aws:iam::111122223333:role/data-processing"
  },
  "Action": "kms:Decrypt",
  "Resource" : "*",
  "Condition": {
    "StringEqualsIgnoreCase": {
      "kms:RecipientAttestation:NitroTPMPCR4":
 "abc1de4f2dcf774f6e3b679f62e5f120065b2e408dcea327bd1c9dddaea6664e7af7935581474844767453082c6f1
    }
  }
}
```

最低权限许可

由于您的 KMS 密钥可以保护敏感信息，因此我们建议遵循最低权限访问原则。在定义密钥政策时，请委派执行任务所需的最低权限。仅当您计划使用其他 IAM 策略进一步限制权限时，才允许对 KMS 密钥政策执行所有操作 (`kms:*`)。如果您计划使用 IAM 策略管理权限，请限制哪些人员能够创建 IAM 策略并将其附加到 IAM 主体，以及监控策略更改。

如果您允许在密钥政策和 IAM 策略中执行所有操作 (`kms:*`)，则主体同时拥有 KMS 密钥的管理和使用权限。建议仅将这些权限委托给特定主体，这是最佳安全实践。您可以通过在密钥政策中明确指明主体，或通过限制 IAM 策略附加到哪些主体来实现。您也可以使用条件键来限制权限。例如，如果进行 API 调用的主体具有条件规则中指定的标签，则可以使用 `aws:PrincipalTag` 来允许所有操作。

要帮助了解中如何评估策略声明 Amazon，请参阅 IAM 用户指南中的策略评估逻辑。我们建议您在撰写策略前先阅读此主题，减少策略产生意外影响的可能，例如向本不应该拥有访问权限的主体提供访问权限。

> **Tip**
>
> 在非生产环境中测试应用程序时，请使用 IAM Access Analyzer 来帮助您将最低权限应用于 IAM 策略。

如果您使用 IAM 用户而不是 IAM 角色，我们强烈建议您启用 Amazon 多因素身份验证 (MFA)，以缓解长期凭证的漏洞。您可使用 MFA 执行以下操作：

- 要求用户在执行特权操作（例如安排密钥删除）前先使用 MFA 验证其凭证。
- 将管理员账户密码和 MFA 设备的所有权分配给不同个人，进行授权拆分。

了解更多

- Amazon 工作职能的托管策略
- 用于编写最低权限 IAM 策略的方法

实施最低权限

当您向 Amazon 服务授予使用 KMS 密钥的权限时，请确保该权限仅对服务必须代表您访问的资源有效。这种最低权限策略有助于防止在 Amazon 服务之间传递请求时未经授权使用 KMS 密钥。

要实施最低权限策略，我们建议使用 Amazon KMS 加密上下文条件密钥和全局来源 ARN 或源账户条件密钥。

使用加密上下文条件键

在使用 Amazon KMS 资源时，实现最低特权权限的最有效方法是在允许委托人调用 Amazon KMS 加密操作的策略中包含kms:EncryptionContext:context-key或kms:EncryptionContextKeys条件密钥。这些条件键特别有效，因为它们将权限与在加密资源时绑定到密文的加密上下文相关联。

仅当策略语句中的操作为或采用EncryptionContext参数的 Amazon KMS 对称加密操作（例如CreateGrant或 Decrypt 之类GenerateDataKey的操作）时，才使用加密上下文条件密钥。（有关支持的操作列表，请参阅 kms:EncryptionContext:context-key 或 kms:EncryptionContextKeys）。如果您使用这些条件键来允许其他操作（例如）DescribeKey，则权限将被拒绝。

将值设置为服务在加密资源时使用的加密上下文。此信息通常可在服务文档的“安全性”章节中找到。例如，Amazon Proton 的加密上下文标识 P Amazon roton 资源及其关联的模板。Amazon Secrets Manager 加密上下文标识密钥及其版本。Amazon Location 的加密上下文标识跟踪器或集合。

以下示例是密钥策略语句允许 Amazon Location Service 代表授权用户创建授权。本政策声明通过使用 k ms: ViaService、kms: 和kms:EncryptionContext:*context-key*条件密钥将权限绑定到特定的跟踪器资源来限制权限。CallerAccount

```
{
  "Sid": "Allow Amazon Location to create grants on behalf of authorized users",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/LocationTeam"
  },
  "Action": "kms:CreateGrant",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ViaService": "geo.us-west-2.amazonaws.com",
      "kms:CallerAccount": "111122223333",
      "kms:EncryptionContext:aws:geo:arn": "arn:aws:geo:us-west-2:111122223333:tracker/
SAMPLE-Tracker"
    }
  }
}
```

使用 aws:SourceArn 或 aws:SourceAccount 条件键

密钥策略语句中的主体是 Amazon 服务主体时，除了 kms:EncryptionContext:*context-key* 条件键外，我们强烈建议您使用 aws:SourceArn 或 aws:SourceAccount 全局条件键。只有当请求 Amazon KMS 来自其他 Amazon 服务时，ARN 和账户值才会包含在授权上下文中。这种条件的组合实施最低权限，避免了潜在的混淆代理情况。在密钥策略中，服务委托人通常不用作委托人，但某些 Amazon 服务（例如）需要 Amazon CloudTrail它。

要使用 aws:SourceArn 或 aws:SourceAccount 全局条件键，将值设置为正在加密的资源的 Amazon Resource Name (ARN) 或账户。例如，在提供 Amazon CloudTrail 权限加密跟踪记录的密钥策略语句中，将 aws:SourceArn 的值设置为跟踪记录的 ARN。请尽可能使用更具体的 aws:SourceArn。将值设置为 ARN 或带通配符的 ARN 模式。如果您不知道资源的 ARN，请改用 aws:SourceAccount。

> **Note**
>
> 如果资源 ARN 包含 Amazon KMS 密钥策略中不允许的字符，则不能在条件密钥的值中使用该资源 ARN。aws:SourceArn改为使用 aws:SourceAccount 条件键。有关密钥策略文档规则的详细信息，请参阅 密钥策略格式。

在以下示例密钥策略中，获得权限的主体是 Amazon CloudTrail 服务主体 cloudtrail.amazonaws.com。为实施最低权限，本策略使用 aws:SourceArn 和 kms:EncryptionContext:*context-key* 条件键。该策略声明 CloudTrail 允许使用 KMS 密钥生成用于加密跟踪的数据密钥。aws:SourceArn 和 kms:EncryptionContext:*context-key* 条件会得到独立评估。使用 KMS 密钥进行指定操作的任何请求都必须满足这两个条件。

为了限制服务对示例账户 (111122223333) 和 us-west-2 区域中 finance 跟踪记录的权限，此策略语句将 aws:SourceArn 条件键设置为特定跟踪记录的 ARN。条件语句使用ArnEquals运算符来确保在匹配时独立评估 ARN 中的每个元素。此示例还使用 kms:EncryptionContext:*context-key* 条件键来限制对特定账户和区域中跟踪记录的权限。

在使用此密钥策略之前，请将示例账户 ID、区域和跟踪记录名称替换为您账户中的有效值。

JSON

```
{
  "Version":"2012-10-17",
```

```
  "Statement": [
    {
      "Sid": "AllowCloudTrailToEncryptLogs",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudtrail.amazonaws.com"
      },
      "Action": "kms:GenerateDataKey",
      "Resource": "*",
      "Condition": {
        "ArnEquals": {
          "aws:SourceArn": [
            "arn:aws:cloudtrail:us-west-2:111122223333:trail/finance"
          ]
        },
        "StringLike": {
          "kms:EncryptionContext:aws:cloudtrail:arn": [
            "arn:aws:cloudtrail:*:111122223333:trail/*"
          ]
        }
      }
    }
  ]
}
```

ABAC for Amazon KMS

基于属性的访问控制 (ABAC) 是一种基于属性定义权限的授权策略。 Amazon KMS 支持 ABAC，允许您根据与 KMS 密钥关联的标签和别名来控制对客户托管密钥的访问。启用 ABAC 的标签和别名条件密钥 Amazon KMS 提供了一种强大而灵活的方式来授权委托人使用 KMS 密钥，而无需编辑策略或管理授权。但是，您应该小心使用这些功能，以免委托人无意中允许或拒绝访问。

如果您使用 ABAC，请注意管理标签和别名的权限现在是访问控制权限。在部署依赖于标签或别名的策略之前，请确保您知道所有 KMS 密钥上的现有标签和别名。添加、删除和更新别名，以及标记和取消标记密钥时，采取合理的预防措施。仅授予需要管理标签和别名权限的委托人该权限，并限制他们可以管理的标签和别名。

> **注意**
>
> 使用 ABAC 时 Amazon KMS，请谨慎行事，不要授予委托人管理标签和别名的权限。更改标签或别名可以允许或拒绝对 KMS 密钥的权限。不具有更改密钥策略或创建授权权限的密钥管理员可以控制对 KMS 密钥的访问，前提是他们有权管理标签或别名。
> 标签和别名的更改最多可能需要 5 分钟的时间才能影响 KMS 密钥授权。最近的更改可能会在 API 操作中显示，然后才会影响授权。
> 要根据 KMS 密钥别名控制对它的访问权限，必须使用条件键。您不能使用别名来表示策略语句的 `Resource` 元素中的 KMS 密钥元素。当别名出现在 `Resource` 元素时，策略语句将应用于别名，而不是关联的 KMS 密钥。

了解更多

- 有关 Amazon KMS 支持 ABAC 的详细信息（包括示例），请参阅使用别名控制对 KMS 密钥的访问和。使用标签控制对 KMS 密钥的访问
- 有关使用标签控制 Amazon 资源访问的更多一般信息，请参阅 ABAC 有什么用 Amazon？ 以及使用 IAM 用户指南中的资源标签控制对资源的访问权限。 Amazon

ABAC 条件键适用于 Amazon KMS

要根据 KMS 密钥的标签和别名授权访问 KMS 密钥，请在密钥政策或 IAM policy 中使用以下条件键。

ABAC 条件键	描述	策略类型	Amazon KMS 操作
aws：ResourceTag	KMS 密钥上的标签（键和值）与策略中的标签（键和值）或标签模式匹配	仅限 IAM policy	KMS 密钥资源操作 [2]
aws:RequestTag/tag-key	请求中的标签（键和值）与策略中的标签（键和值）或标签模式匹配	密钥政策和 IAM policy [1]	TagResource, UntagResource
aws：TagKeys	请求中的标签键与策略中的标签键匹配	密钥政策和 IAM policy [1]	TagResource, UntagResource

ABAC 条件键	描述	策略类型	Amazon KMS 操作
kms: ResourceAliases	与 KMS 密钥关联的别名与策略中的别名或别名模式匹配	仅限 IAM policy	KMS 密钥资源操作 [2]
kms: RequestAlias	表示请求中的 KMS 密钥的别名与策略中的别名或别名模式匹配。	密钥政策和 IAM policy [1]	加密操作，，DescribeKeyGetPublicKey

[1] 任何可在密钥政策中使用的条件键也可以在 IAM policy 中使用，但只有在密钥政策允许它时。

[2] KMS 密钥资源操作是特定 KMS 密钥授权的操作。若要标识 KMS 密钥资源操作，请在 Amazon KMS 权限表中，在操作的 Resources 列中查找的 KMS 密钥的值。

例如，您可以使用这些条件键创建以下策略。

- 具有 kms:ResourceAliases 的 IAM policy，可授予将 KMS 密钥与特定别名或别名模式结合使用的权限。这与依赖标签的策略略有不同：尽管您可以在策略中使用别名模式，但每个别名在 Amazon Web Services 账户 和区域中都必须是唯一的。这允许您对一组选定的 KMS 密钥应用策略，而无需在策略声明中列出 KMS 密钥 ARNs 的密钥。要从集中添加或删除 KMS 密钥，请更改 KMS 密钥的别名。
- 带有 kms:RequestAlias 的密钥策略，可允许委托人在 Encrypt 操作中使用 KMS 密钥，但前提是仅当 Encrypt 请求使用该别名标识 KMS 密钥时。
- 带有 aws:ResourceTag/*tag-key* 的 IAM policy，可拒绝将 KMS 密钥与特定标签键和标签值结合使用的权限。这使您可以将策略应用于一组选定的 KMS 密钥，而无需在策略声明中列出 KMS 密钥的密钥。 ARNs 要在集中添加或删除 KMS 密钥，请标记或取消标记 KMS 密钥。
- 带有 aws:RequestTag/*tag-key* 的 IAM policy，可允许委托人仅删除 "Purpose"="Test" KMS 密钥标签。
- 带有 aws:TagKeys 的 IAM policy，可拒绝使用 Restricted 标签键标记或取消标记 KMS 密钥的权限。

ABAC 使访问管理具有灵活性和可扩展性。例如，您可以使用 aws:ResourceTag/*tag-key* 条件键创建 IAM policy，该策略允许委托人仅在 KMS 密钥具有 Purpose=Test 标签时将 KMS 密钥用于特定操作。该策略适用于 Amazon Web Services 账户的所有区域中的所有 KMS 密钥。

当附加到用户或角色时，以下 IAM policy 允许委托人将带有 Purpose=Test 标签的所有现有 KMS 密钥用于指定操作。要提供对新的或现有 KMS 密钥的访问权限，您不需要更改策略。只需将 Purpose=Test 标签附加到 KMS 密钥。同样，要从具有 Purpose=Test 标签的 KMS 密钥中删除此访问权限，请编辑或删除标签。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AliasBasedIAMPolicy",
      "Effect": "Allow",
      "Action": [
        "kms:Decrypt",
        "kms:Encrypt",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Purpose": "Test"
        }
      }
    }
  ]
}
```

但是，如果您使用此功能，在管理标记和别名时要加小心。添加、更改或删除标签或别名可能会无意中允许或拒绝对 KMS 密钥的访问。不具有更改密钥策略或创建授权权限的密钥管理员可以控制对 KMS 密钥的访问，前提是他们有权管理标签和别名。为了减轻这种风险，请考虑限制管理标签的权限和别名。例如，您可能想要仅允许特色级委托人管理 Purpose=Test 标签。有关详细信息，请参阅 使用别名控制对 KMS 密钥的访问 和 使用标签控制对 KMS 密钥的访问。

标签还是别名？

Amazon KMS 支持带有标签和别名的 ABAC。这两种选项都提供了灵活、可扩展的访问控制策略，但它们彼此略有不同。

您可以根据自己的特定使用模式决定使用标签或 Amazon 使用别名。例如，如果您已经向大多数管理员授予了标记权限，则基于别名控制授权策略可能会更容易。或者，如果您接近每个 KMS 密钥的别名数量配额，您可能更喜欢基于标签的授权策略。

以下益处是大家都感兴趣的。

基于标签的访问控制的益处

- 对不同类型的 Amazon 资源使用相同的授权机制。

  您可以使用相同的标签或标签键来控制对多种资源类型的访问，例如 Amazon Relational Database Service（Amazon RDS）集群、Amazon Elastic Block Store（Amazon EBS）卷和 KMS 密钥。此功能支持多种不同的授权模型，这些模型比传统的基于角色的访问控制更加灵活。
- 授权访问一组 KMS 密钥。

  您可以使用标签来管理对同一 Amazon Web Services 账户 和区域中的一组 KMS 密钥的访问权限。将相同的标签或标签键分配给您选择的 KMS 密钥。然后创建一个基于标签或标签密钥的简单 easy-to-maintain策略声明。要在授权组中添加或删除 KMS 密钥，请添加或删除标签；您无需编辑策略。

基于别名的访问控制的益处

- 根据别名授权对加密操作的访问。

  大多数基于请求的属性策略条件，包括 a ws:RequestTag/tag-key，仅影响添加、编辑或删除属性的操作。但是 k ms: RequestAlias 条件密钥根据用于在请求中识别 KMS 密钥的别名来控制对加密操作的访问。例如，您可以授予委托人在 `Encrypt` 操作中使用 KMS 密钥的权限，但只有当 `KeyId` 参数的值为 `alias/restricted-key-1` 时。要满足此条件，需要以下所有条件：

  - KMS 密钥必须与该别名相关联。
  - 请求必须使用别名来标识 KMS 密钥。
  - 委托人必须拥有使用受 `kms:RequestAlias` 条件约束的 KMS 密钥的权限。

  如果您的应用程序通常使用别名或别名来引用 KMS 密钥 ARNs ，则此功能特别有用。
- 提供非常有限的权限。

  在 Amazon Web Services 账户 和区域中，别名必须是唯一的。因此，基于别名授予委托人访问 KMS 密钥的权限可能比基于标签授予他们访问权限更严格。与别名不同，标签可分配给同一账户和区域中的多个 KMS 密钥。如果选择，则可以使用别名模式（例如 `alias/test*`）为委托人授予对

同一账户和区域中一组 KMS 密钥的访问权限。但是，允许或拒绝访问特定别名允许对 KMS 密钥进行非常严格的控制。

对 ABAC 进行故障排除 Amazon KMS

基于 KMS 密钥的标签和别名控制对 KMS 密钥的访问非常方便，且功能强大。但是，它容易出现一些您希望防止的可预测错误。

由于标签更改而更改访问权限

如果某个标签被删除或其值被更改，则仅能基于该标签访问 KMS 密钥的委托人将被拒绝访问 KMS 密钥。当拒绝策略语句中包含的标签添加到 KMS 密钥时，也会发生这种情况。向 KMS 密钥添加与策略相关的标签可以允许访问应被拒绝访问 KMS 密钥的委托人。

例如，假设委托人可以基于 `Project=Alpha` 标签访问 KMS 密钥，例如以下示例 IAM policy 语句提供的权限。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPolicyWithResourceTag",
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:ap-southeast-1:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Project": "Alpha"
        }
      }
    }
  ]
}
```

如果该标签已从该 KMS 密钥中删除或标签值发生更改，则委托人不再具有将 KMS 密钥用于指定操作的权限。当委托人尝试在使用客户托管密钥的 Amazon 服务中读取或写入数据时，这一点可能会变得显而易见。要跟踪标签的更改，请查看您的 CloudTrail日志TagResource或UntagResource 条目。

要在不更新策略的情况下恢复访问，请更改 KMS 密钥上的标签。这一措施除了短时间在整个 Amazon KMS中生效之后，产生的影响极小。为了防止这样的错误，请仅向需要它的委托人授予标记和取消标记权限，并将其标记权限限制到他们需要管理的标签。更改标签之前，请搜索策略以检测依赖于标签的访问权限，并在具有该标签的所有区域中获取 KMS 密钥。您可以考虑在更改特定标签时创建 Amazon CloudWatch 警报。

由于别名更改而导致的访问权限更改

如果别名被删除或与其他 KMS 密钥关联，则仅能基于该别名访问 KMS 密钥的委托人将被拒绝访问 KMS 密钥。当拒绝策略语句中包含与 KMS 密钥关联的别名时，也会发生这种情况。向 KMS 密钥添加与策略相关的别名也可以允许访问应被拒绝访问 KMS 密钥的委托人。

例如，以下 IAM 策略声明使用 k ms: ResourceAliases 条件密钥允许使用任意指定别名访问账户中不同区域的 KMS 密钥。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AliasBasedIAMPolicy",
      "Effect": "Allow",
      "Action": [
        "kms:List*",
        "kms:Describe*",
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "kms:ResourceAliases": [
            "alias/ProjectAlpha",
            "alias/ProjectAlpha_Test",
            "alias/ProjectAlpha_Dev"
          ]
        }
```

```
      }
    }
  ]
}
```

要跟踪别名更改，请查看 CloudTrail 日志中是否有CreateAliasUpdateAlias、和DeleteAlias条目。

要在不更新策略的情况下恢复访问，请更改与 KMS 密钥关联的别名。由于每个别名只能与账户和区域中的一个 KMS 密钥相关联，因此管理别名比管理标签要困难一些。恢复一些委托人对一个 KMS 密钥的访问权限可能会拒绝相同或其他委托人对其他 KMS 密钥的访问。

为了防止出现此错误，请仅向需要它的委托人授予别名管理权限，并将其别名管理权限限制到他们需要管理的别名。在更新或删除别名之前，请搜索策略以检测依赖于别名的访问权限，并在与别名关联的所有区域中查找 KMS 密钥。

由于别名配额而拒绝访问

如果 KMS 密钥超过该账户和地区每个 KMS 密钥配额的默认别名，则获得 kms: ResourceAliases 条件授权使用 KMS 密钥的用户将获得`AccessDenied`例外。

要恢复访问，请删除与 KMS 密钥关联的别名，使其符合配额。或者使用备用机制授予用户访问 KMS 密钥的权限。

延迟的授权更改

您对标签和别名的更改最多可能需要 5 分钟的时间才能影响 KMS 密钥授权。因此，标签或别名更改可能会在 API 操作影响授权之前反映在响应中。这种延迟可能比影响大多数 Amazon KMS 操作的短暂的最终一致性延迟要长。

例如，您可能拥有一个 IAM policy，允许某些委托人使用带有 `"Purpose"="Test"` 标签的任何 KMS 密钥。然后，您将 `"Purpose"="Test"` 标签添加到 KMS 密钥。尽管TagResource操作已完成且ListResourceTags响应确认标签已分配给 KMS 密钥，但委托人可能在长达五分钟内无法访问 KMS 密钥。

为了防止出现错误，请将此预期延迟构建到您的代码中。

由于别名更新而失败的请求

当您更新别名时，您将现有别名与另一个 KMS 密钥关联。

解密和指定别名或别名 ARN 的ReEncrypt请求可能会失败，因为该别名现在与未加密密文的 KMS 密钥相关联。这种情况通常会返回 IncorrectKeyException 或者 NotFoundException。或者如果请求中没有 KeyId 或 DestinationKeyId 参数，则操作可能会失败，并出现 AccessDenied 异常，因为调用方不再具有对加密密文的 KMS 密钥的访问权限。

您可以通过查看CreateAliasUpdateAlias、和 CloudTrail 日志条目的DeleteAlias日志来跟踪更改。您还可以在ListAliases响应中使用该LastUpdatedDate字段的值来检测更改。

例如，以下ListAliases示例响应显示kms:ResourceAliases条件中的ProjectAlpha_Test别名已更新。因此，基于别名具有访问权限的委托人将失去对先前关联的 KMS 密钥的访问权限。相反，他们可以访问新关联的 KMS 密钥。

```
$ aws kms list-aliases --query 'Aliases[?starts_with(AliasName, `alias/ProjectAlpha`)]'

{
    "Aliases": [
        {
            "AliasName": "alias/ProjectAlpha_Test",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/ProjectAlpha_Test",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1566518783.394,
            "LastUpdatedDate": 1605308931.903
        },
        {
            "AliasName": "alias/ProjectAlpha_Restricted",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/
ProjectAlpha_Restricted",
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1553410800.010,
            "LastUpdatedDate": 1553410800.010
        }
    ]
}
```

这种变化的补救办法并不简单。您可以再次更新别名以将其与原始 KMS 密钥关联。但是，在执行操作之前，您需要考虑该更改对当前关联的 KMS 密钥的影响。如果委托人在加密操作中使用后一个 KMS 密钥，他们可能需要继续访问该密钥。在这种情况下，您可能需要更新策略以确保委托人有权使用这两个 KMS 密钥。

您可以防止出现这样的错误：在更新别名之前，搜索策略以检测依赖于别名的访问。然后，在与别名关联的所有区域中获取 KMS 密钥。请仅向需要它的委托人授予别名管理权限，并将其别名管理权限限制到他们需要管理的别名。

Amazon KMS 中的 RBAC

基于角色的访问控制（RBAC）是一种授权策略，仅为用户提供履行其工作职责所需的权限，仅此而已。Amazon KMS 支持 RBAC，允许您通过在密钥政策中精细指定密钥用法权限来控制对密钥的访问。密钥政策指定授予密钥访问权限的资源、操作、效果、主体和可选条件。

要在 Amazon KMS 中实施 RBAC，我们建议将密钥用户和密钥管理员权限分开。

Key users

以下密钥政策示例允许 `ExampleUserRole` IAM 角色使用 KMS 密钥。

```
{
            "Sid": "Allow use of the key",
            "Effect": "Allow",
            "Principal": {
                "Amazon": "arn:aws-cn:iam::111122223333:role/ExampleUserRole"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncrypt*",
                "kms:GenerateDataKey*",
                "kms:DescribeKey"
            ],
            "Resource": "*"
  }
```

您的密钥用户需要的权限可能少于本示例中的用户。请仅分配用户需要的权限。使用以下问题帮助您确定并进一步优化权限。

- 哪些 IAM 主体（角色或用户）需要访问密钥？
- 每位主体需要使用密钥来执行哪些操作？ 例如，主体是否只需要加密和签名权限？
- 用户是人还是 Amazon 服务？ 如果是 Amazon 服务，可以使用条件密钥将密钥用法限制为特定 Amazon 服务。

Key administrators

以下密钥政策示例允许 ExampleAdminRole IAM 角色管理 KMS 密钥。

```
{
            "Sid": "Allow access for Key Administrators",
            "Effect": "Allow",
            "Principal": {
                "Amazon": "arn:aws-cn:iam::111122223333:role/ExampleAdminRole"
            },
            "Action": [
                "kms:Create*",
                "kms:Describe*",
                "kms:Enable*",
                "kms:List*",
                "kms:Put*",
                "kms:Update*",
                "kms:Revoke*",
                "kms:Disable*",
                "kms:Get*",
                "kms:Delete*",
                "kms:TagResource",
                "kms:UntagResource",
                "kms:ScheduleKeyDeletion",
                "kms:CancelKeyDeletion"
            ],
            "Resource": "*"
    }
```

在本示例中，您的密钥管理员需要的权限可能少于管理员的权限。仅分配密钥管理员所需的权限。

仅向用户授予履行其角色所需的权限。用户的权限可能有所不同，具体取决于密钥是在测试环境中使用还是在生产环境中使用。如果您在某些非生产环境中使用限制较少的权限，请先实施流程来测试策略，然后再将其发布到生产环境中。

了解更多

- IAM 身份（用户、用户组和角色）
- 访问控制的类型

允许其他账户中的用户使用 KMS 密钥

您可以允许其他 Amazon Web Services 账户中的用户或角色使用您账户中的 KMS 密钥。跨账户访问需要在 KMS 密钥的密钥策略和外部用户账户的 IAM policy 中拥有权限。

跨账户权限仅对以下操作有效：

- 加密操作
- CreateGrant
- DescribeKey
- GetKeyRotationStatus
- GetPublicKey
- ListGrants
- RetireGrant
- RevokeGrant

如果您向另一个账户中的用户授予对其他操作的权限，则这些权限将不起作用。例如，如果您向另一个账户中的委托人授予对 IAM policy 的 kms:ListKeys 权限，或者对密钥策略中的 KMS 密钥的 kms:ScheduleKeyDeletion 权限，则用户对资源调用这些操作的尝试仍会失败。

有关在不同账户中使用 KMS 密钥进行 Amazon KMS 操作的详细信息，请参阅 Amazon KMS 权限 和 在其他账户中使用 KMS 密钥 中的跨账户使用列。Amazon Key Management Service API 参考中每个 API 描述中还有一个跨账户使用部分。

> **Warning**
>
> 请谨慎授予委托人使用 KMS 密钥的权限。只要有可能，请按照最小特权原则。仅针对用户所需的操作为他们授予对所需的 KMS 密钥的访问权。
> 此外，请谨慎使用任何不熟悉的 KMS 密钥，尤其是其他账户中的 KMS 密钥。恶意用户可能会授予您使用其 KMS 密钥获取有关您或您账户的信息的权限。
> 有关使用策略保护您账户中的资源的更多信息，请参阅 IAM policy 的最佳实践。

要将使用 KMS 密钥的权限授予其他账户中的用户和角色，您必须使用两种不同类型的策略：

- KMS 密钥的密钥策略必须向外部账户（或外部账户中的用户和角色）授予使用 KMS 密钥的权限。密钥策略在拥有 KMS 密钥的账户中。

- 外部账户中的 IAM policy 必须将密钥策略权限委托给其用户和角色。这些策略在外部账户中设置，并向该账户中的用户和角色授予权限。

密钥策略决定谁可以访问 KMS 密钥。IAM policy 决定谁确实能够访问 KMS 密钥。单独的密钥策略和 IAM policy 都不足够，您必须更改此两者。

要编辑密钥策略，您可以使用 Amazon Web Services 管理控制台中的策略视图，也可以使用 CreateKey 或 PutKeyPolicy 操作。

有关编辑 IAM policy 时的帮助信息，请参阅。将 IAM 策略与配合使用 Amazon KMS

有关说明密钥策略和 IAM policy 如何协同工作以允许在其他账户中使用 KMS 密钥的示例，请参阅。示例 2：用户扮演角色并有权在其他地方使用 KMS 密钥 Amazon Web Services 账户

您可以在您的 Amazon CloudTrail 日志中查看对 KMS 密钥产生的跨账户 Amazon KMS 操作。在其他账户中使用 KMS 密钥的操作将记录在调用方的账户和 KMS 密钥所有者账户中。

主题

> **Note**
>
> 本主题中的示例展示了如何结合使用密钥策略和 IAM policy 来提供和限制对 KMS 密钥的访问。这些通用示例不是为了表示任何特定的 Amazon Web Services 服务 对于 KMS 密钥需要的权限。有关 Amazon Web Services 服务 需要的权限的信息，请参阅服务文档中的加密主题。

步骤 1：在本地账户中添加密钥策略语句

KMS 密钥的密钥策略是谁可以访问 KMS 密钥以及他们可以执行哪些操作的主要决定因素。密钥策略始终在拥有 KMS 密钥的账户中。与 IAM policy 不同，密钥策略不指定资源。资源是指与密钥策略关联的 KMS 密钥。在提供跨账户权限时，KMS 密钥的密钥策略必须向外部账户（或外部账户中的用户和角色）授予使用 KMS 密钥的权限。

要向外部账户授予使用 KMS 密钥的权限，请向密钥策略添加一条语句用于指定此外部账户。在密钥策略的 Principal 元素中，输入外部账户的 Amazon Resource Name (ARN)。

当您在密钥策略中指定外部账户时，外部账户中的 IAM 管理员可以使用 IAM policy 来将这些权限委派给外部账户中的任何用户和角色。他们还可以决定用户和角色可以执行在密钥策略中指定的哪些操作。

只有在托管 KMS 密钥及其密钥策略的区域中启用了外部账户时，授予给外部账户及其委托人的权限才有效。有关默认情况下未启用的区域（“选择加入区域”）的信息，请参阅《Amazon Web Services 一般参考》 中的 Managing Amazon Web Services 区域。

例如，假定您希望允许账户 444455556666 使用账户 111122223333 中的对称加密 KMS 密钥。为此，请将与以下示例中的策略语句类似的策略语句添加到账户 111122223333 中 KMS 密钥的密钥策略。此策略语句授权外部账户 444455556666 在加密操作中使用对称加密 KMS 密钥的 KMS 密钥。

> **Note**
>
> 以下示例演示了与其他账户共享 KMS 密钥的示例密钥策略。请将示例 Sid、Principal 和 Action 的值替换为 KMS 密钥预期用途的有效值。

```
{
    "Sid": "Allow an external account to use this KMS key",
    "Effect": "Allow",
    "Principal": {
        "AWS": [
            "arn:aws:iam::444455556666:root"
        ]
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*"
}
```

您可以在密钥策略中指定特定的外部用户和角色，而不是向外部账户授予权限。但是，这些用户和角色无法使用该 KMS 密钥，直到外部账户中的 IAM 管理员将适当的 IAM policy 附加到其身份。IAM policy

可以向在密钥策略中指定的所有或部分外部用户和角色授予权限。并且，它们可以允许执行在密钥策略中指定的所有或部分操作。

在密钥策略中指定身份会限制外部账户中的 IAM 管理员可以提供的权限。但是，它使两个账户的策略管理变得更加复杂。例如，假定您需要添加用户或角色。您必须将该身份添加到拥有 KMS 密钥的账户中的密钥策略，并在此身份的账户中创建 IAM policy。

要在密钥策略中指定特定外部用户或角色，请在 `Principal` 元素中输入外部账户中的用户或角色的 Amazon Resource Name (ARN)。

例如，以下示例密钥策略语句允许账户 `444455556666` 中的 `ExampleRole` 使用账户 `111122223333` 中的 KMS 密钥。此密钥策略语句授权外部账户 `444455556666` 在加密操作中使用对称加密 KMS 密钥的 KMS 密钥。

> **Note**
>
> 以下示例演示了与其他账户共享 KMS 密钥的示例密钥策略。请将示例 `Sid`、`Principal` 和 `Action` 的值替换为 KMS 密钥预期用途的有效值。

```
{
    "Sid": "Allow an external account to use this KMS key",
    "Effect": "Allow",
    "Principal": {
        "AWS": "arn:aws:iam::444455556666:role/ExampleRole"
    },
    "Action": [
        "kms:Encrypt",
        "kms:Decrypt",
        "kms:ReEncrypt*",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
    ],
    "Resource": "*"
}
```

> **Note**
>
> 除非您使用条件限制密钥政策，否则不要在允许权限的任何密钥政策语句将主体设置为星号（*）。星号表示允许每个 Amazon Web Services 账户 权限中的每个身份使用 KMS 密钥，除

非另一个策略语句明确拒绝它。其他 Amazon Web Services 账户 中的用户只要在自己的账户中拥有相应的权限，就可以使用您的 KMS 密钥。

您还需要确定您想要向外部账户授予哪些权限。例如，您可能希望授予用户解密但不加密的权限，或者查看 KMS 密钥但不使用 KMS 密钥的权限。有关针对 KMS 密钥的权限的列表，请参阅 Amazon KMS 权限。

创建 KMS 密钥时设置密钥政策

当您使用 CreateKey 操作创建 KMS 密钥时，您可以使用其 `Policy` 参数来指定密钥策略，该策略向外部账户或外部用户和角色授予使用 KMS 密钥的权限。

当您在 Amazon Web Services 管理控制台 中创建 KMS 密钥时，还会创建其密钥策略。当您在密钥管理员和密钥用户部分中选择身份时，Amazon KMS 会将这些身份的策略语句添加到 KMS 密钥的密钥策略中。密钥用户部分还允许您将外部账户添加为密钥用户。

当您输入外部账户的账户 ID 时，Amazon KMS 向密钥策略添加两个语句。此操作仅影响密钥策略。外部账户中的用户和角色无法使用该 KMS 密钥，直到您附加 IAM policy 以向他们授予部分或所有这些权限。

第一个密钥策略语句向外部账户授予在加密操作中使用 KMS 密钥的权限。第二个密钥策略语句允许外部账户在 KMS 上创建、查看和撤消授权，但仅限请求来自与 Amazon KMS 集成的 Amazon 服务的情况。这些权限允许加密用户数据的其他 Amazon 服务使用 KMS 密钥。这些权限是为加密 Amazon 服务中用户数据的 KMS 密钥设计的

步骤 2：在外部账户中添加 IAM policy

拥有 KMS 密钥的账户中的密钥策略设置权限的有效范围。但是，在附加委派这些权限的 IAM policy 或使用授权来管理对 KMS 密钥的访问之前，外部账户中的用户和角色无法使用此 KMS 密钥。IAM policy 在外部账户中设置。

如果密钥策略向外部账户授予权限，则可以将 IAM policy 附加到账户中的任何用户或角色。但是，如果密钥策略向指定的用户或角色授予权限，则 IAM policy 只能将这些权限授予全部或部分指定用户和角色。如果 IAM policy 向其他外部用户或角色授予使用 KMS 密钥的权限，则它不起作用。

密钥策略还限制 IAM policy 中的操作。IAM policy 可以委派在密钥策略中指定的所有或部分操作。如果 IAM policy 列出未在密钥策略中指定的操作，则这些权限无效。

以下示例 IAM policy 允许委托人使用账户 111122223333 中的 KMS 密钥执行加密操作。要向账户 444455556666 中的用户和角色授予此权限，请将策略附加到账户 444455556666 中的用户或角色。

> **Note**
>
> 以下示例演示了与其他账户共享 KMS 密钥的示例 IAM policy。请将示例 `Sid`、`Resource` 和 `Action` 的值替换为 KMS 密钥预期用途的有效值。

```
{
    "Sid": "AllowUseOfKeyInAccount111122223333",
    "Effect": "Allow",
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt",
      "kms:ReEncrypt*",
      "kms:GenerateDataKey*",
      "kms:DescribeKey"
    ],
    "Resource": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

请注意有关该策略的以下详细信息：

- 与密钥策略不同，IAM policy 语句不包含 `Principal` 元素。在 IAM policy 中，委托人是附加了策略的身份。
- IAM policy 中的 `Resource` 元素标识委托人可以使用的 KMS 密钥。要指定 KMS 密钥，请将其密钥 ARN 添加到 `Resource` 元素中。
- 您可以在 `Resource` 元素中指定多个 KMS 密钥。但是，如果您没有在 `Resource` 元素中指定特定的 KMS 密钥，您可能会无意中比预期授予针对更多 KMS 密钥的权限。
- 要允许外部用户将 KMS 密钥和与 Amazon KMS 集成的 Amazon 服务一起使用，您可能需要向密钥策略或 IAM policy 添加权限。有关更多信息，请参阅 允许将外部 KMS 密钥与 Amazon Web Services 服务结合使用。

有关使用 IAM policy 的更多信息，请参阅 IAM 策略。

允许将外部 KMS 密钥与 Amazon Web Services 服务结合使用

您可以向不同账户中的用户授予将您的 KMS 密钥和与 Amazon KMS 集成的服务一起使用的权限。例如，外部账户中的用户可以使用您的 KMS 密钥加密 Amazon S3 存储桶中的对象或加密存储在 Amazon Secrets Manager 中的密钥。

密钥策略必须向外部用户或外部用户的账户授予使用 KMS 密钥的权限。此外，您需要将 IAM policy 附加到某个身份，以向用户授予使用 Amazon Web Services 服务的权限。该服务还可能要求用户在密钥策略或 IAM policy 中具有其他权限。有关 Amazon Web Services 服务需要对客户管理型密钥拥有的权限列表，请参阅相关服务的用户指南或开发人员指南中“安全”章节的“数据保护”主题。

在其他账户中使用 KMS 密钥

如果您有权在另一个 Amazon Web Services 账户 中使用 KMS 密钥，您可以在 Amazon Web Services 管理控制台、Amazon 开发工具包、Amazon CLI 和 Amazon Tools for PowerShell 中使用 KMS 密钥。

要在 shell 命令或 API 请求中标识其他账户中的 KMS 密钥，请使用以下密钥标识符。

- 对于密码操作、DescribeKey 和 GetPublicKey，请使用 KMS 密钥的密钥 ARN 或别名 ARN。
- 对于 CreateGrant、GetKeyRotationStatus、ListGrants 和 RevokeGrant，请使用 KMS 密钥的密钥 ARN。

如果您只输入密钥 ID 或别名名称，Amazon 会假定 KMS 密钥位于您的账户中。

Amazon KMS 控制台不会在其他账户中显示 KMS 密钥，即使您有权使用它们。此外，在其他 Amazon 服务的控制台中显示的 KMS 密钥的列表不包括其他账户中的 KMS 密钥。

要在 Amazon 服务的控制台中指定不同账户中的 KMS 密钥，您必须输入 KMS 密钥的密钥 ARN。所需的密钥标识符因服务而异，并且在服务控制台及其 API 操作之间可能会有所不同。有关详细信息，请参阅服务文档。

控制对多区域密钥的访问

您可以在合规性、灾难恢复和备份场景中使用多区域密钥，这些场景在使用单区域密钥的情况下更为复杂。但是，由于多区域密钥的安全属性与单区域密钥的安全属性明显不同，我们建议在授权创建、管理和使用多区域密钥时务必谨慎。

> **Note**
>
> 在 `Resource` 字段中包通配符的现有 IAM policy 语句现在同时应用于单区域密钥和多区域密钥。要将其限制为单区域 KMS 密钥或多区域密钥，请使用 k ms: MultiRegion 条件密钥。

使用您的授权工具防止在单区域足够的任何情况下创建和使用多区域密钥。仅允许委托人将多区域密钥复制到需要他们的密钥 Amazon Web Services 区域 中。仅对需要多区域密钥的主体授予权限，并且仅对需要多区域密钥的任务授予权限。

您可以使用密钥策略、IAM 策略和授权，允许 IAM 委托人管理和使用您的多区域密钥。 Amazon Web Services 账户每个多区域密钥都是一个独立的资源，具有唯一的密钥 ARN 和密钥策略。您需要为每个密钥建立和维护密钥政策，并确保新的和现有的 IAM policy 实施您的授权策略。

要支持多区域密钥，请 Amazon KMS 使用 IAM 服务关联角色。这个角色为 Amazon KMS 授予同步共享属性所需的权限。有关更多信息，请参阅 授权同步多 Amazon KMS 区域密钥。

主题

- 多区域密钥的授权基础知识
- 授权多区域密钥管理员和用户

多区域密钥的授权基础知识

为多区域密钥设计密钥政策和 IAM policy 时，请考虑以下原则。

- 密钥策略 — 每个多区域密钥都是一个独立的 KMS 密钥资源，具有自己的密钥策略。您可以将相同或不同的密钥策略应用于相关多区域密钥集中的每个密钥。密钥策略不是多区域密钥的共享属性。Amazon KMS 不会在相关的多区域密钥之间复制或同步密钥策略。

  在 Amazon KMS 控制台中创建副本密钥时，为了方便起见，控制台会显示主密钥的当前密钥策略。您可以使用此密钥策略、对其进行编辑或删除和替换。但是，即使您接受未更改的主密钥策略，也 Amazon KMS 不会同步这些策略。例如，如果您更改主密钥的密钥策略，则副本密钥的密钥策略将保持不变。
- 默认密钥策略-使用CreateKey和`ReplicateKey`操作创建多区域密钥时，除非您在请求中指定密钥策略，否则将应用默认密钥策略。这与应用于单区域密钥的默认密钥策略相同。
- IAM policy — 与所有 KMS 密钥一样，只要当密钥政策允许时，您才可以使用 IAM policy 来控制对多区域密钥的访问。默认情况下，IAM 策略适用于所有 Amazon Web Services 区域 人。但是，您可以使用条件键（例如 a ws: RequestedRegion ）来限制对特定区域的权限。

要创建主密钥和副本密钥，委托人必须对应用于创建密钥的区域的 IAM policy 具有 `kms:CreateKey` 权限。

- 补助金 — Amazon KMS 补助金是区域性的。每个授权都允许对一个 KMS 密钥的权限。您可以使用授权来允许对多区域主密钥或副本密钥的权限。但是，您不能使用单个授权来允许对多个 KMS 密钥的权限，即使它们是相关的多区域密钥。
- 密钥 ARN — 每个多区域密钥都有一个唯一的密钥 ARN。相关的多区域密钥 ARNs 的密钥具有相同的分区、账户和密钥 ID，但区域不同。

  要将 IAM policy 语句应用于特定的多区域密钥，请使用其密钥 ARN 或包含该区域的密钥 ARN 模式。要将 IAM policy 语句应用于所有相关多区域密钥，请在 ARN 的区域元素中使用通配符 (*)，如下例所示。

```
{
  "Effect": "Allow",
  "Action": [
    "kms:Describe*",
    "kms:List*"
  ],
  "Resource": {
      "arn:aws:kms:*::111122223333:key/mrk-1234abcd12ab34cd56ef1234567890ab"
  }
}
```

  要将策略声明应用于您的所有多区域密钥 Amazon Web Services 账户，您可以使用 kms:MultiRegion 策略条件或包含独特`mrk-`前缀的密钥 ID 模式。
- 服务相关角色-创建多区域主键的委托人必须拥有 iam: 权限。CreateServiceLinkedRole

  要同步相关多区域密钥的共享属性，请担任 I Amazon KMS AM 服务相关角色。 Amazon KMS 每当您创建多区域主键 Amazon Web Services 账户 时，都会在中创建服务相关角色。（如果角色存在， Amazon KMS 将重新创建它，这没有任何有害影响。）该角色在所有区域中都有效。Amazon KMS 要允许创建（或重新创建）服务相关角色，创建多区域主键的委托人必须拥有 iam: 权限。CreateServiceLinkedRole

授权多区域密钥管理员和用户

创建和管理多区域密钥的委托人需要在主区域和副本区域中具有以下权限：

- `kms:CreateKey`

- kms:ReplicateKey
- kms:UpdatePrimaryRegion
- iam:CreateServiceLinkedRole

创建主密钥

要创建多区域主密钥，委托人需要在主密钥所在区域有效的 IAM 策略中的 kms: CreateKey 和 iam: CreateServiceLinkedRole 权限。具有这些权限的委托人可以创建单区域和多区域密钥，除非您限制其权限。

该iam:CreateServiceLinkedRole权限允许 Amazon KMS 创建AWSServiceRoleForKeyManagementServiceMultiRegionKeys角色以同步相关多区域密钥的共享属性。

例如，此 IAM 策略将允许主体创建多区域密钥，为这些密钥附加策略，以及为多区域密钥创建服务相关角色。

JSON

```
{
  "Version":"2012-10-17",
  "Statement":{
      "Action": [
        "kms:CreateKey",
        "iam:CreateServiceLinkedRole"
      ],
      "Effect":"Allow",
      "Resource":"*"
  }
}
```

要允许或拒绝创建多区域主密钥的权限，请使用 k ms: MultiRegion 条件密钥。有效值为 true（多区域密钥）或 false（单区域密钥）。例如，以下 IAM policy 语句使用 Deny 操作与 kms:MultiRegion 条件键来防止委托人创建多区域密钥。

JSON

```
{
```

```
  "Version":"2012-10-17",
  "Statement":{
      "Action":"kms:CreateKey",
      "Effect":"Deny",
      "Resource":"*",
      "Condition": {
          "Bool": {
            "kms:MultiRegion": true
          }
      }
  }
}
```

复制密钥

要创建多区域副本密钥，委托人需要以下权限：

- km ReplicateKey s：主密钥的密钥策略中的权限。
- kms：在副本密钥区域有效的 IAM 策略中的CreateKey权限。

允许这些权限时请谨慎。它们允许委托人创建 KMS 密钥和授权其使用的密钥策略。kms:ReplicateKey 权限还授权跨 Amazon KMS内的区域边界传输密钥材料。

要限制可以复制多区域密钥的范围，请使用 k ms: ReplicaRegion 条件密钥。 Amazon Web Services 区域 它只限制 kms:ReplicateKey 权限。否则，会没有效果。例如，以下密钥策略允许委托人复制该主密钥，但仅限在指定区域中进行。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/Administrator"
  },
  "Action": "kms:ReplicateKey",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:ReplicaRegion": [
         "us-east-1",
         "eu-west-3",
         "ap-southeast-2"
      ]
```

```
    }
  }
}
```

更新主区域

授权的委托人可以将副本密钥转换为主密钥，从而将以前的主密钥更改为副本密钥。该操作称为更新主区域。要更新主区域，委托人需要两个区域的 kms: UpdatePrimaryRegion 权限。您可以在密钥政策或 IAM policy 中提供这些权限。

- 主密钥上的 `kms:UpdatePrimaryRegion`。此权限必须在主密钥区域中有效。
- 副本密钥上的 `kms:UpdatePrimaryRegion`。此权限必须在副本密钥区域中有效。

例如，以下密钥策略向可以担任管理员角色的用户授予更新 KMS 密钥的主区域的权限。此 KMS 密钥可以是此操作中的主密钥或副本密钥。

```
{
  "Effect": "Allow",
  "Resource": "*",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/Administrator"
  },
  "Action": "kms:UpdatePrimaryRegion"
}
```

要限制 Amazon Web Services 区域 可以托管主密钥的，请使用 k ms: PrimaryRegion 条件密钥。例如，以下 IAM 政策声明允许委托人更新中多区域密钥的主区域 Amazon Web Services 账户，但前提是新的主区域是指定区域之一。

```
{
  "Effect": "Allow",
  "Action": "kms:UpdatePrimaryRegion",
  "Resource": {
      "arn:aws:kms:*:111122223333:key/*"
  },
  "Condition": {
    "StringEquals": {
      "kms:PrimaryRegion": [
         "us-west-2",
         "sa-east-1",
```

```
          "ap-southeast-1"
        ]
      }
    }
}
```

使用和管理多区域密钥

默认情况下，有权在 Amazon Web Services 账户 和区域中使用和管理 KMS 密钥的委托人也有权使用和管理多区域密钥。但是，您可以使用 k ms: MultiRegion 条件密钥来仅允许单区域密钥或仅允许多区域密钥。或者使用 k ms: MultiRegionKeyType 条件密钥仅允许多区域主键或仅允许副本密钥。两个条件密钥都控制对CreateKey操作和使用现有 KMS 密钥的任何操作（例如 Enc rypt 或）的访问权限EnableKey。

以下示例 IAM policy 语句使用 `kms:MultiRegion` 条件键，以防委托人使用或管理任何多区域密钥。

```
{
  "Effect": "Deny",
  "Action": "kms:*",
  "Resource": "*",
  "Condition": {
    "Bool": "kms:MultiRegion": true
  }
}
```

此示例 IAM policy 语句使用 `kms:MultiRegionKeyType` 条件，以允许委托人计划和取消删除密钥，但仅限于多区域副本密钥。

```
{
  "Effect": "Allow",
  "Action": [
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": {
      "arn:aws:kms:us-west-2:111122223333:key/*"
  },
  "Condition": {
    "StringEquals": "kms:MultiRegionKeyType": "REPLICA"
  }
}
```

确定对 Amazon KMS keys 的访问权限

要确定当前有权访问 Amazon KMS key 的所有对象，您必须检查 KMS 密钥的密钥策略、应用于 KMS 密钥的所有授权，以及所有潜在 Amazon Identity and Access Management (IAM) 策略。您可以执行该操作来确定 KMS 密钥的潜在使用范围或帮助您满足合规性或审计要求。以下主题有助于您生成当前有权访问 KMS 密钥的 Amazon 委托人（身份）的完整列表。

主题

检查密钥策略

密钥策略是控制对 KMS 密钥访问的主要方法。每个 KMS 密钥都有且只有一个密钥策略。

如果密钥策略由默认密钥策略组成或包含默认密钥策略，则密钥策略允许账户中的 IAM 管理员使用 IAM policy 控制对 KMS 密钥的访问。此外，如果密钥策略赋予其他 Amazon Web Services 账户 使用 KMS 密钥的权限，则外部账户中的 IAM 管理员可以使用 IAM policy 委派这些权限。要确定可访问 KMS 密钥的委托人的完整列表，请检查 IAM policy。

要查看您账户中的 Amazon KMS 客户自主管理型密钥或 Amazon 托管式密钥的密钥策略，请使用 Amazon Web Services 管理控制台或 Amazon KMS API 中的 GetKeyPolicy 操作。要查看密钥策略，必须对 KMS 密钥具备 `kms:GetKeyPolicy` 权限。有关查看 KMS 密钥的密钥策略的说明，请参阅 the section called “查看密钥政策”。

检查密钥策略文档，并记下每个策略语句的 `Principal` 元素中指定的所有委托人。在具有 `Allow` 后果的策略语句中，`Principal` 元素中的 IAM 用户、IAM 角色 和 Amazon Web Services 账户将拥有对此 KMS 密钥的访问权限。

> **Note**
>
> 除非您使用条件限制密钥政策，否则不要在允许权限的任何密钥政策语句将主体设置为星号（*）。星号表示允许每个 Amazon Web Services 账户 权限中的每个身份使用 KMS 密钥，除非另一个策略语句明确拒绝它。其他 Amazon Web Services 账户 中的用户只要在自己的账户中拥有相应的权限，就可以使用您的 KMS 密钥。

以下示例使用默认密钥策略中的策略语句来演示如何执行该操作。

Example 策略语句 1

```
{
  "Sid": "Enable IAM User Permissions",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:root"},
  "Action": "kms:*",
  "Resource": "*"
}
```

在策略语句 1 中，arn:aws:iam::111122223333:root 是指向 Amazon Web Services 账户 111122223333 的 Amazon 账户主体。（这不是账户的根用户。）默认情况下，当您使用 Amazon Web Services 管理控制台创建新的 KMS 密钥，或以编程方式创建新的 KMS 密钥但未提供密钥策略时，密钥策略文档中将包含与此类似的策略语句。

如果密钥策略文档中具有允许访问 Amazon Web Services 账户的语句，则将使该账户中的 IAM policy 允许访问 KMS 密钥。这意味着，即使账户中的用户和角色未在密钥策略文档中显式列为主体，也可以访问 KMS 密钥。请务必检查所有列为委托人的 Amazon Web Services 账户 中的所有 IAM policy，以确定它们是否允许访问此 KMS 密钥。

Example 策略语句 2

```
{
  "Sid": "Allow access for Key Administrators",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/KMSKeyAdmins"},
  "Action": [
    "kms:Describe*",
    "kms:Put*",
    "kms:Create*",
    "kms:Update*",
    "kms:Enable*",
    "kms:Revoke*",
    "kms:List*",
    "kms:Disable*",
    "kms:Get*",
    "kms:Delete*",
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
```

```
  "Resource": "*"
}
```

在策略语句 2 中，arn:aws:iam::111122223333:role/KMSKeyAdmins 是指 Amazon Web Services 账户 111122223333 中名为 KMSKeyAdmins 的 IAM 角色。被授权代入该角色的用户被允许执行策略语句中列出的操作，即用于管理 KMS 密钥的管理操作。

Example 策略语句 3

```
{
  "Sid": "Allow use of the key",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/EncryptionApp"},
  "Action": [
    "kms:DescribeKey",
    "kms:GenerateDataKey*",
    "kms:Encrypt",
    "kms:ReEncrypt*",
    "kms:Decrypt"
  ],
  "Resource": "*"
}
```

在策略语句 3 中，arn:aws:iam::111122223333:role/EncryptionApp 指的是 Amazon Web Services 账户 111122223333 中名为 EncryptionApp 的 IAM 角色。被授权代入此角色的主体被允许执行策略语句中列出的操作，这包括对称加密 KMS 密钥的 加密操作。

Example 策略语句 4

```
{
  "Sid": "Allow attachment of persistent resources",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:role/EncryptionApp"},
  "Action": [
    "kms:ListGrants",
    "kms:CreateGrant",
    "kms:RevokeGrant"
  ],
  "Resource": "*",
  "Condition": {"Bool": {"kms:GrantIsForAWSResource": true}}
}
```

在策略语句 4 中，`arn:aws:iam::111122223333:role/EncryptionApp` 指的是 Amazon Web Services 账户 111122223333 中名为 EncryptionApp 的 IAM 角色。被授权代入此角色的主体被允许执行策略语句中列出的操作。向大多数与 Amazon KMS 集成的 Amazon 服务（特别是使用授权的服务）委托 KMS 密钥使用权限时，都将需要将这些操作以及示例策略语句 3 中允许的操作结合使用。`Condition` 元素中的 kms:GrantIsForAWSResource 值确保仅在被委托方是与 Amazon KMS 集成，并使用授权进行授权的 Amazon 服务时才允许委托。

要了解可以在密钥策略文档中指定委托人的所有不同方法，请参阅 IAM 用户指南中的指定委托人。

要了解有关 Amazon KMS 密钥策略的更多信息，请参阅中的关键政策 Amazon KMS。

检查 IAM policy

除密钥策略和授权外，您还可以使用 IAM policy 来允许对 KMS 密钥的访问。有关 IAM policy 和密钥策略如何协同工作的更多信息，请参阅 Amazon KMS 权限疑难解答。

要确定当前可通过 IAM policy 访问 KMS 密钥的委托人，可以使用基于浏览器的 IAM policy simulator 工具，也可以向 IAM API 发出请求。

检查 IAM policy 的方法

- 使用 IAM policy simulator 检查 IAM policy
- 使用 IAM API 检查 IAM policy

使用 IAM policy simulator 检查 IAM policy

IAM policy simulator 有助于您了解哪些委托人可以通过 IAM policy 访问 KMS 密钥。

使用 IAM policy simulator 确定对 KMS 密钥的访问权限

1. 登录 Amazon Web Services 管理控制台，然后打开位于 https://policysim.aws.amazon.com/ 的 IAM policy simulator。
2. 在用户、组和角色窗格中，选择您要模拟其策略的用户、组或角色。
3. (可选) 清除您要从模拟中忽略的任何策略旁边的复选框。要模拟所有策略，则将所有策略保持选中状态。
4. 在策略模拟器窗格中，执行以下操作：
   a. 对于选择服务，请选择 Key Management Service。
   b. 要模拟特定 Amazon KMS 操作，请针对选择操作选择要模拟的操作。要模拟所有 Amazon KMS 操作，请选择全选。

5. （可选）默认情况下，策略模拟器会模拟对所有 KMS 密钥的访问。要模拟对特定 KMS 密钥的访问，请选择 Simulation Settings（模拟设置），然后键入要模拟的 KMS 密钥的 Amazon Resource Name (ARN)。
6. 选择 Run Simulation (运行模拟)。

您可以在结果部分查看模拟的结果。对 Amazon Web Services 账户 中的每个用户、组和角色重复第 2 至 6 步。

使用 IAM API 检查 IAM policy

您可以使用 IAM API 以编程方式检查 IAM policy。以下步骤提供了如何执行该操作的一般概述：

1. 对于密钥策略中列为主体的每个 Amazon Web Services 账户（即以 `"Principal": {"AWS": "arn:aws:iam::111122223333:root"}` 格式列出的每个 Amazon 账户主体），请使用 IAM API 中的 ListUsers 和 ListRoles 操作来获取该账户中的所有用户和角色。
2. 对于列表中的每个用户和角色，使用 IAM API 中的 SimulatePrincipalPolicy 操作传递以下参数：
   - 对于 `PolicySourceArn`，指定列表中用户或角色的 Amazon Resource Name (ARN)。您只能为每个 `SimulatePrincipalPolicy` 请求指定一个 `PolicySourceArn`，因此必须多次调用此操作，针对列表中的每个用户和角色分别调用一次。
   - 对于 `ActionNames` 列表，指定要模拟的每个 Amazon KMS API 操作。要模拟所有 Amazon KMS API 操作，请使用 `kms:*`。要测试单独的 Amazon KMS API 操作，请在每个 API 操作之前添加“`kms:`”，例如“`kms:ListKeys`”。有关 Amazon KMS API 操作的完整列表，请参阅《Amazon Key Management Service API 参考》中的 操作。
   - （可选）要确定用户或角色是否有权访问特定的 KMS 密钥，请使用 `ResourceArns` 参数指定 KMS 密钥的 Amazon 资源名称（ARN）列表。要确定用户或角色是否有权访问任何 KMS 密钥，请忽略 `ResourceArns` 参数。

IAM 通过以下评估决策来响应每个 `SimulatePrincipalPolicy` 请求：`allowed`、`explicitDeny` 或 `implicitDeny`。对于每个包含评估决策 `allowed` 的响应，其中包含允许的特定 Amazon KMS API 操作的名称。它还包括评估中使用的 KMS 密钥的 ARN（如果有）。

检查授予

授权是一种高级机制，用于指定您或与 Amazon KMS 集成的 Amazon 服务可用于指定使用 KMS 密钥的方式和时间的权限。授权将附加到 KMS 密钥，每个授权都包含一位委托人，此委托人可获得使用

KMS 密钥和所允许的操作列表的权限。授权是密钥策略的替代方案，对特定的使用案例很有用。有关更多信息，请参阅 Amazon KMS 中的授权。

要获取 KMS 密钥的授权列表，请使用 Amazon KMS ListGrants 操作。您可以检查 KMS 密钥的授权来确定当前有权通过这些授权使用 KMS 密钥的对象。例如，下面是某个授权的 JSON 表示形式，此授权使用 中的 list-grantsAmazon CLI 命令获取。

```
{"Grants": [{
  "Operations": ["Decrypt"],
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "Name": "0d8aa621-43ef-4657-b29c-3752c41dc132",
  "RetiringPrincipal": "arn:aws:iam::123456789012:root",
  "GranteePrincipal": "arn:aws:sts::111122223333:assumed-role/aws:ec2-infrastructure/
i-5d476fab",
  "GrantId": "dc716f53c93acacf291b1540de3e5a232b76256c83b2ecb22cdefa26576a2d3e",
  "IssuingAccount": "arn:aws:iam::111122223333:root",
  "CreationDate": 1.444151834E9,
  "Constraints": {"EncryptionContextSubset": {"aws:ebs:id": "vol-5cccfb4e"}}
}]}
```

要了解有权使用 KMS 密钥的对象，可查找 "GranteePrincipal" 元素。在上述示例中，被授权委托人是与 EC2 实例 i-5d476fab 关联的假设用户角色。EC2 基础设施使用此角色将加密的 EBS 卷 vol-5cccfb4e 附加到此实例。在此情况下，EC2 基础设施角色有权使用 KMS 密钥，因为您之前创建了受此 KMS 密钥保护的加密的 EBS 卷。之后，您将此卷附加到了 EC2 实例。

下面是 JSON 表示形式的另一个授权示例，此授权使用 中的 list-grantsAmazon CLI 命令获取。在以下示例中，被授权者委托人是另一个 Amazon Web Services 账户。

```
{"Grants": [{
  "Operations": ["Encrypt"],
  "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
  "Name": "",
  "GranteePrincipal": "arn:aws:iam::444455556666:root",
  "GrantId": "f271e8328717f8bde5d03f4981f06a6b3fc18bcae2da12ac38bd9186e7925d11",
  "IssuingAccount": "arn:aws:iam::111122223333:root",
  "CreationDate": 1.444151269E9
}]}
```

加密上下文

> **Note**
>
> 您不能在使用非对称 KMS 密钥或 HMAC KMS 密钥的加密操作中指定加密上下文。非对称算法和 MAC 算法不支持加密上下文。

所有具有对称加密 KMS 密钥的 Amazon KMS 加密操作都接受加密上下文，即一组可选的非机密键值对，其中包含有关数据的其他上下文信息。您可以在 Amazon KMS 中的 `Encrypt` 操作中插入加密上下文，从而增强 Amazon KMS API 解密调用的授权和可审计性。Amazon KMS 将加密上下文作为额外验证数据（AAD）来支持经过身份验证的加密。加密上下文以加密方式绑定到加密文字，以便需要使用相同的加密上下文解密数据。

加密上下文不是密钥，且没有加密。它以明文显示在 Amazon CloudTrail 日志中，以便您可以使用它来标识和分类加密操作。您的加密上下文不应包含敏感信息。我们建议您的加密上下文描述正在加密或解密的数据。例如，在加密文件时，您可以将文件路径的一部分用作加密上下文。

```
"encryptionContext": {
    "department": "10103.0"
}
```

例如，当加密使用 Amazon Elastic Block Store (Amazon EBS) CreateSnapshot 操作创建的卷和快照时，Amazon EBS 使用卷 ID 作为加密上下文值。

```
"encryptionContext": {
  "aws:ebs:id": "vol-abcde12345abc1234"
}
```

您还可以使用加密上下文来细化或限制对您账户中 Amazon KMS keys 的访问。您可以使用加密上下文作为授权中的约束，以及作为策略语句中的条件。加密上下文密钥及其值可以是包含 `aws` 的任意字符串。这些值与 Amazon 生成的标签（例如 aws:cloudformation:stack-name）截然不同。有关更多信息，请参阅 kms:EncryptionContext: 上下文密钥。

要了解如何使用加密上下文来保护加密数据的完整性，请参阅 安全博客上的博文 Amazon Key Management ServiceHow to Protect the Integrity of Your Encrypted Data by Using and EncryptionContextAmazon。

加密上下文规则

Amazon KMS 对加密上下文密钥和值强制执行以下规则。

- 加密上下文对中的键和值必须是简单的文本字符串。如果您使用其他类型（例如整数或浮点），则 Amazon KMS 会将它解释为字符串。
- 加密上下文中的密钥和值可以包括 Unicode 字符。如果加密上下文包含密钥策略或 IAM policy 中不允许的字符，则您将无法在策略条件密钥中指定加密上下文，例如 kms:EncryptionContext:*context-key* 和 kms:EncryptionContextKeys。有关密钥策略文档规则的详细信息，请参阅 密钥策略格式。有关 IAM policy 档规则的详细信息，请参阅《IAM 用户指南》中的 IAM 名称要求。

策略中的加密上下文

加密上下文主要用于验证完整性和真实性。但是，您也可以使用加密上下文来控制对密钥策略和 IAM policy 中对称加密 Amazon KMS keys 的访问。

kms:EncryptionContext: 和 kms:EncryptionContextKeys 条件键仅在请求包括特定加密上下文密钥或键值对时允许（或拒绝）权限。

例如，以下密钥策略语句允许 RoleForExampleApp 角色在 Decrypt 操作中使用 KMS 密钥。它使用 kms:EncryptionContext:*context-key* 条件键以仅在请求中的加密上下文包含 AppName:ExampleApp 加密上下文对时允许此权限。

```
{
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/RoleForExampleApp"
  },
  "Action": "kms:Decrypt",
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "kms:EncryptionContext:AppName": "ExampleApp"
    }
  }
}
```

有关这些加密上下文条件键的更多信息，请参阅 的条件密钥 Amazon KMS。

授权中的加密上下文

创建授权时可以包含授权约束，从而设定授予权限的条件。Amazon KMS 支持 `EncryptionContextEquals` 和 `EncryptionContextSubset` 这两种授权约束，并且这两种约束都涉及密码操作请求中的加密上下文。在使用这些授权约束时，授权中的权限仅在加密操作请求中的加密上下文满足授权约束的要求时有效。

例如，您可以将 `EncryptionContextEquals` 授权约束添加到允许 GenerateDataKey 操作的授权中。使用此约束时，授权仅在请求中的加密上下文与授权约束中的加密上下文大小写完全匹配时，允许操作。

```
$ aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:user/exampleUser \
    --retiring-principal arn:aws:iam::111122223333:role/adminRole \
    --operations GenerateDataKey \
    --constraints EncryptionContextEquals={Purpose=Test}
```

来自被授予者委托人的以下请求将满足 `EncryptionContextEquals` 约束。

```
$ aws kms generate-data-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --key-spec AES_256 \
    --encryption-context Purpose=Test
```

有关授权约束的详细信息，请参阅 使用授权约束。有关授权的详细信息，请参阅 the section called “授权”。

记录加密上下文

Amazon KMS 使用 Amazon CloudTrail 记录加密上下文，以便您可以确定访问了哪些 KMS 密钥和数据。日志条目会准确显示哪些 KMS 密钥被用来加密或解密了由日志条目中的加密上下文引用的特定数据。

> **Important**
>
> 由于加密上下文会被记录，它不得包含敏感信息。

存储加密上下文

为了简化在调用 Decrypt 或 ReEncrypt 操作时任何加密上下文的使用，可以将加密上下文与加密数据存储在一起。我们建议您仅存储足够的加密上下文，以帮助您在需要用于加密或解密时创建完整的加密上下文。

例如，如果加密上下文是文件的完全限定路径，仅将该路径部分与加密文件内容存储在一起。然后，当您需要完整的加密上下文时，可以从存储的片段重建它。如果有人擅自改动文件，例如重命名或将其移动到其他位置，加密上下文值更改，解密请求将失败。

测试您的权限

要使用 Amazon KMS，您必须拥有 Amazon 可以用来验证您的 API 请求的凭证。此凭证必须包括访问 KMS 密钥和别名的权限。权限由密钥政策、IAM policy、授权和跨账户存取控制决定。除了控制对 KMS 密钥的访问外，您还可以控制对 CloudHSM 和自定义密钥存储的访问权限。

您可以指定 DryRun API 参数来确认您具有使用 Amazon KMS 密钥的所需权限。您还可以使用 DryRun 来验证 Amazon KMS API 调用中的请求参数指定是否正确。

主题

什么是 DryRun 参数？

DryRun 是一个可选的 API 参数，您可以指定该参数来验证 Amazon KMS API 调用是否成功。在实际调用 Amazon KMS 之前，请使用 DryRun 测试您的 API 调用。您可以验证如下内容。

- 您具有使用 Amazon KMS 密钥的所需权限。
- 您已正确指定调用中的参数。

Amazon KMS 支持在某些 API 操作中使用 DryRun 参数：

- CreateGrant
- Decrypt
- DeriveSharedSecret
- Encrypt

- GenerateDataKey
- GenerateDataKeyPair
- GenerateDataKeyPairWithoutPlaintext
- GenerateDataKeyWithoutPlaintext
- GenerateMac
- ReEncrypt
- RetireGrant
- RevokeGrant
- Sign
- Verify
- VerifyMac

使用 DryRun 参数将产生费用，并将按标准 API 请求计费。有关 Amazon KMS 定价的更多信息，请参阅 Amazon Key Management Service 定价。

使用 DryRun 参数的所有 API 请求都适用于 API 的请求限额，如果您超过 API 请求限额，则可能会导致节流异常。例如，无论使用 DryRun 还是不使用 DryRun 调用 Decrypt，都将计入相同的加密操作限额。请参阅限制 Amazon KMS 请求，了解更多信息。

对 Amazon KMS API 操作的每次调用都被捕获为事件并记录在 Amazon CloudTrail 日志中。任何指定 DryRun 参数的操作的输出都会显示在您的 CloudTrail 日志中。有关更多信息，请参阅 使用 Amazon CloudTrail 记录 Amazon KMS API 调用。

使用 API 指定 DryRun

要使用 DryRun，请在支持该参数的 Amazon CLI 命令和 Amazon KMS API 调用中指定 –dry-run 参数。当您这样做时，Amazon KMS 将验证您的调用是否会成功。使用 DryRun 的 Amazon KMS 调用将始终失败并返回一条消息，其中包含有关调用失败原因的信息。消息可能包括以下例外情况：

- DryRunOperationException - 如果 DryRun 未指定，则请求会成功。
- ValidationException - 请求因指定错误的 API 参数而失败。
- AccessDeniedException - 您无权在 KMS 资源上执行指定的 API 操作。

例如，以下命令会使用 CreateGrant 操作并创建一个授权，以允许被授权担任 keyUserRole 角色的用户对指定的对称 KMS 密钥调用解密操作。DryRun 参数已指定。

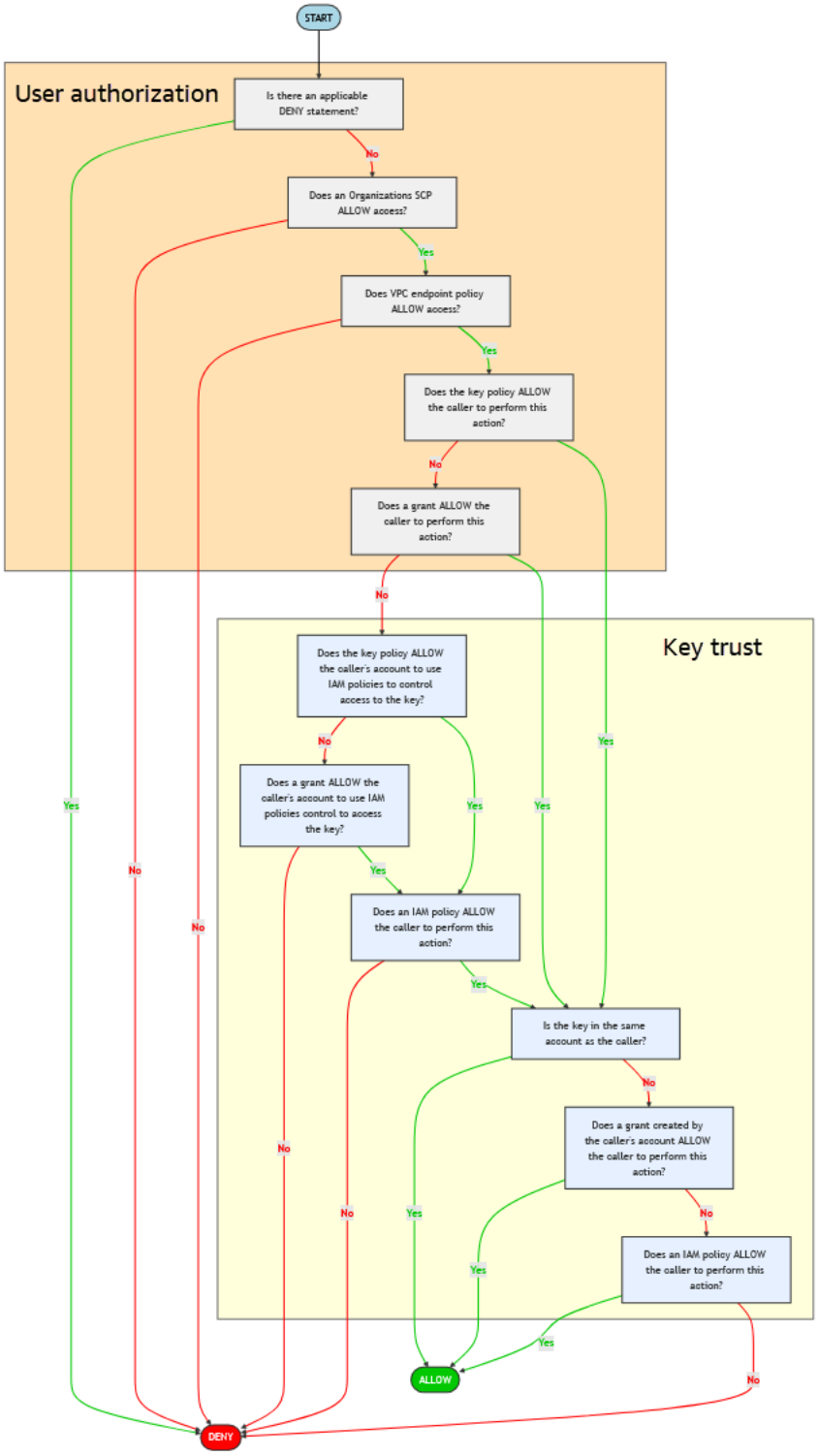
```
$  aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::111122223333:role/keyUserRole \
    --operations Decrypt \
    --dry-run
```

Amazon KMS 权限疑难解答

在授权 KMS 密钥的访问权限时，需要 Amazon KMS 评估以下内容：

- 附加到 KMS 密钥的密钥策略。密钥策略始终在拥有 KMS 密钥的 Amazon Web Services 账户 和区域中定义。
- 附加到发出请求的用户或角色的所有 IAM policy。管理委托人对 KMS 密钥的使用的 IAM policy 始终在委托人的 Amazon Web Services 账户中定义。
- 应用于 KMS 密钥的所有授权。
- 可能应用于请求以使用 KMS 密钥的其他类型的策略，例如 Amazon Organizations 服务控制策略和 VPC 终端节点策略。这些策略是可选的，并且在默认情况下允许执行所有操作，但您可以使用它们限制授予委托人的权限。

Amazon KMS 共同评估这些策略机制，以确定是允许还是拒绝对 KMS 密钥的访问。为此，请 Amazon KMS 使用与以下流程图中描述的过程相似的过程。以下流程图提供策略评估流程的可视化表示。



此流程图分为两个部分。这两个部分按顺序显示，但通常会同时对它们进行评估。

- 使用授权根据其密钥政策、IAM policy、授权和其他适用的策略来确定是否允许您使用 KMS 密钥。

- 密钥信任确定您是否应信任允许您使用的 KMS 密钥。一般而言，您信任自己的资源 Amazon Web Services 账户。但是， Amazon Web Services 账户 如果您的账户中的授权或 IAM 策略允许您使用 KMS 密钥，您也可以放心在其他方式中使用 KMS 密钥。

您可以使用此流程图了解为什么允许或拒绝向发起人授予使用 KMS 密钥的权限。您还可以使用它评估您的策略和授权。例如，此流程图显示某个调用方可能被显式 DENY 语句拒绝访问，或者由于在密钥政策、IAM policy 或授权中没有显式 ALLOW 语句而被拒绝访问。

流程图可以解释一些常见的许可方案。

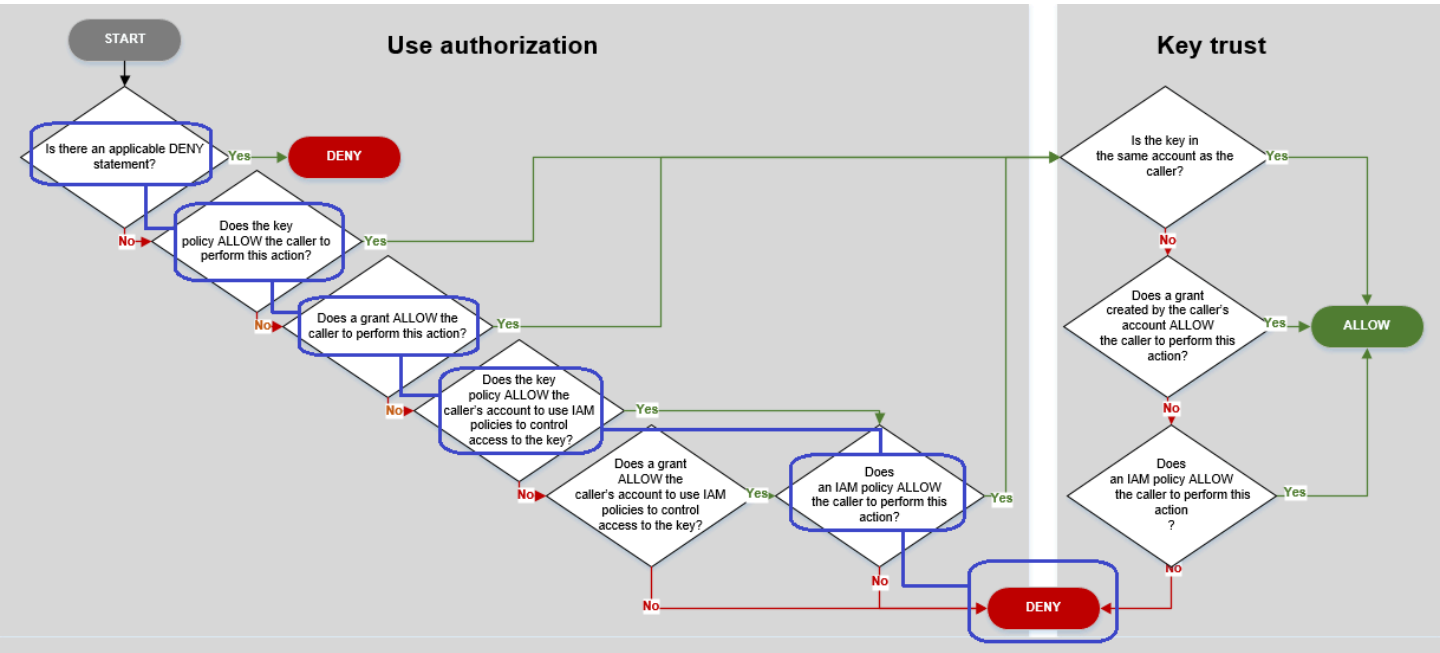
权限示例

- 示例 1：用户被拒绝访问其中的 KMS 密钥 Amazon Web Services 账户
- 示例 2：用户扮演角色并有权在其他地方使用 KMS 密钥 Amazon Web Services 账户

示例 1：用户被拒绝访问其中的 KMS 密钥 Amazon Web Services 账户

Alice 是 11112 Amazon Web Services 账户 2223333 中的 IAM 用户。她被拒绝访问同一 Amazon Web Services 账户中的 KMS 密钥。为什么 Alice 无法使用 KMS 密钥？

在这种情况下，Alice 被拒绝访问该 KMS 密钥，因为没有密钥政策、IAM policy 或为她授予所需权限的授权。KMS 密钥的密钥策略允许使用 IAM 策略 Amazon Web Services 账户 来控制对 KMS 密钥的访问，但是没有 IAM 策略授予 Alice 使用 KMS 密钥的权限。



考虑用于此示例的相关策略。

- Alice 想要使用的 KMS 密钥具有默认密钥策略。此政策允许拥有 KMS 密钥的 Amazon Web Services 账户 使用 IAM policy 控制对 KMS 密钥的访问。此密钥政策满足流程图中的密钥政策是否允许调用方账户使用 IAM policy 来控制对密钥的访问？条件。

  JSON

```
{
  "Version":"2012-10-17",
  "Id" : "key-test-1",
  "Statement" : [ {
    "Sid" : "DelegateToIAMPolicies",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "arn:aws:iam::111122223333:root"
    },
    "Action" : "kms:*",
    "Resource" : "*"
  } ]
}
```

- 但是，没有任何密钥政策、IAM policy 或授权向 Alice 授予 KMS 密钥使用权限。因此，Alice 被拒绝使用 KMS 密钥的权限。

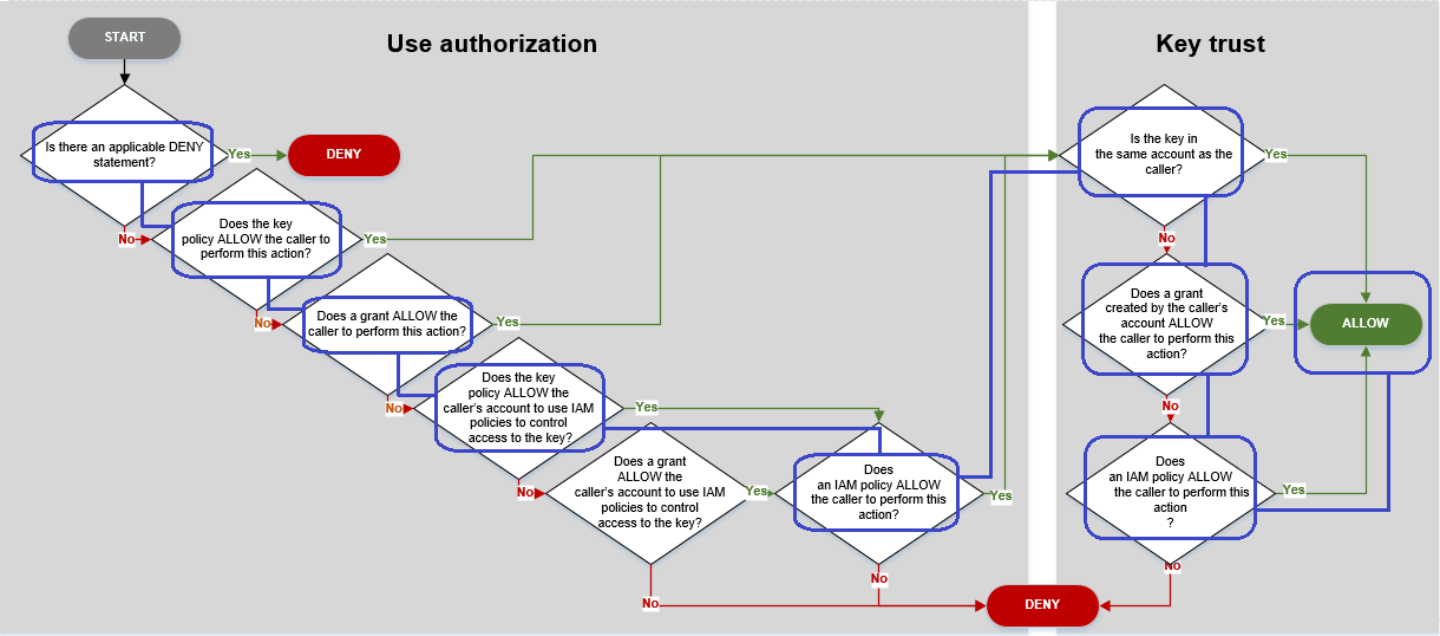
示例 2：用户扮演角色并有权在其他地方使用 KMS 密钥 Amazon Web Services 账户

Bob 是账户 1 (111122223333) 中的一个用户。他可以在加密操作中使用账户 2 (444455556666) 中的 KMS 密钥。如何才能实现？

> **Tip**
>
> 在评估跨账户权限时，请记住，密钥策略在 KMS 密钥的账户中指定。IAM policy 在调用方账户中指定，即使调用方使用的是不同的账户。有关提供对 KMS 密钥的跨账户访问的详细信息，请参阅 允许其他账户中的用户使用 KMS 密钥。

- 账户 2 中 KMS 密钥的密钥政策允许账户 2 使用 IAM policy 来控制对 KMS 密钥的访问。

- 账户 2 中 KMS 密钥的密钥策略允许账户 1 在加密操作中使用 KMS 密钥。但是，账户 1 必须使用 IAM policy 授予其委托人对 KMS 密钥的访问权限。
- 账户 1 中的 IAM policy 允许 `Engineering` 角色将账户 2 中的 KMS 密钥用于加密操作。
- Bob 是账户 1 中的用户，有权限代入 `Engineering` 角色。
- Bob 可以信任此 KMS 密钥，因为即使它不在他的账户中，他账户中的一个 IAM policy 也会向他授予使用此 KMS 密钥的显式权限。



考虑一下 Bob（账户 1 中的用户）使用账户 2 中 KMS 密钥的策略。

- KMS 密钥的密钥政策允许账户 2（444455556666，拥有 KMS 密钥的账户）使用 IAM policy 来控制对 KMS 密钥的访问。此密钥策略还允许账户 1 (111122223333) 在加密操作中使用 KMS 密钥（在策略语句的 `Action` 元素中指定）。但是，账户 1 中的任何人都无法使用账户 2 中的 KMS 密钥，直到账户 1 定义授权委托人访问 KMS 密钥的 IAM policy。

  在流程图中，账户 2 中的此密钥政策满足密钥政策是否允许调用方账户使用 IAM policy 来控制对密钥的访问？条件。

  JSON

```
{
    "Id": "key-policy-acct-2",
    "Version":"2012-10-17",
    "Statement": [
```

```
        {
            "Sid": "PermissionUseIAMpolicies",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::444455556666:root"
            },
            "Action": "kms:*",
            "Resource": "*"
        },
        {
            "Sid": "AllowAccount1UseThisKMSKeys",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::111122223333:root"
            },
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncryptFrom",
                "kms:ReEncryptTo",
                "kms:GenerateDataKey",
                "kms:GenerateDataKeyWithoutPlaintext",
                "kms:DescribeKey"
            ],
            "Resource": "*"
        }
    ]
}
```

- 调用方 Amazon Web Services 账户 （账户 1，111122223333）中的 IAM 策略授予委托人使用账户 2（444455556666）中的 KMS 密钥执行加密操作的权限。`Action` 元素向该委托人委派的权限与账户 2 中的密钥策略向账户 1 授予的权限相同。要将这些权限授予账户 1 中的 `Engineering` 角色，将此内联策略嵌入到 `Engineering` 角色中。

  仅当账户 2 中的 KMS 密钥的密钥政策向账户 1 授予使用此 KMS 密钥的权限时，类似于此策略的跨账户 IAM policy 才有效。此外，账户 1 只能向其委托人授予执行此密钥策略已授予该账户的操作的权限。

  在流程图中，这满足 IAM policy 是否允许调用方执行此操作？条件。

  JSON

```
{
```

```
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "kms:Encrypt",
                "kms:Decrypt",
                "kms:ReEncryptFrom",
                "kms:ReEncryptTo",
                "kms:GenerateDataKey",
                "kms:GenerateDataKeyWithoutPlaintext",
                "kms:DescribeKey"
            ],
            "Resource": [
            "arn:aws:kms:us-
west-2:444455556666:key/1234abcd-12ab-34cd-56ef-1234567890ab"
            ]
        }
    ]
}
```

- 最后一个必需元素是账户 1 中 Engineering 角色的定义。此角色中的 AssumeRolePolicyDocument 允许 Bob 代入 Engineering 角色。

```
{
    "Role": {
        "Arn": "arn:aws:iam::111122223333:role/Engineering",
        "CreateDate": "2019-05-16T00:09:25Z",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": {
                "Principal": {
                    "AWS": "arn:aws:iam::111122223333:user/bob"
                },
                "Effect": "Allow",
                "Action": "sts:AssumeRole"
            }
        },
        "Path": "/",
        "RoleName": "Engineering",
        "RoleId": "AROA4KJY2TU23Y7NK62MV"
    }
}
```

Amazon KMS 访问控制词汇表

以下主题描述了 Amazon KMS 访问控制中的重要术语和概念。

身份验证

身份验证是验证您的身份的过程。要向发送请求 Amazon KMS，您必须 Amazon 使用您的 Amazon 凭据登录。

Authorization

授权提供发送创建、管理或使用 Amazon KMS 资源的请求的权限。例如，您必须获得授权才能在加密操作中使用 KMS 密钥。

要控制对 Amazon KMS 资源的访问权限，请使用密钥策略、IAM 策略和授权。每个 KMS 密钥都必须有一个密钥政策。如果密钥策略允许，您还可以使用 IAM policy 和授权来允许相关主体访问 KMS 密钥。要优化授权，您可以使用 条件键，以仅在请求或资源满足您指定的条件时才允许或拒绝访问。您还可以向您信任的 其他 Amazon Web Services 账户 中的主体授予访问权限。

使用身份进行身份验证

身份验证是您 Amazon 使用身份凭证登录的方式。您必须以 IAM 用户身份进行身份验证 Amazon Web Services 账户根用户，或者通过担任 IAM 角色进行身份验证。

对于编程访问， Amazon 提供 SDK 和 CLI 来对请求进行加密签名。有关更多信息，请参阅《IAM 用户指南》中的适用于 API 请求的Amazon 签名版本 4。

Amazon Web Services 账户 root 用户

创建时 Amazon Web Services 账户，首先会有一个名为 Amazon Web Services 账户 root 用户的登录身份，该身份可以完全访问所有资源 Amazon Web Services 服务 和资源。我们强烈建议不要使用根用户进行日常任务。有关需要根用户凭证的任务，请参阅《IAM 用户指南》中的需要根用户凭证的任务。

联合身份

作为最佳实践，要求人类用户使用与身份提供商的联合身份验证才能 Amazon Web Services 服务 使用临时证书进行访问。

联合身份是指来自您的企业目录、Web 身份提供商的用户 Amazon Directory Service ，或者 Amazon Web Services 服务 使用来自身份源的凭据进行访问的用户。联合身份代入可提供临时凭证的角色。

IAM 用户和群组

IAM 用户是对某个人员或应用程序具有特定权限的一个身份。建议使用临时凭证，而非具有长期凭证的 IAM 用户。有关更多信息，请参阅 IAM 用户指南中的要求人类用户使用身份提供商的联合身份验证才能 Amazon 使用临时证书进行访问。

IAM 组指定一组 IAM 用户，便于更轻松地对大量用户进行权限管理。有关更多信息，请参阅《IAM 用户指南》中的 IAM 用户的使用案例。

IAM 角色

IAM 角色是具有特定权限的身份，可提供临时凭证。您可以通过从用户切换到 IAM 角色（控制台）或调用 Amazon CLI 或 Amazon API 操作来代入角色。有关更多信息，请参阅《IAM 用户指南》中的担任角色的方法。

IAM 角色对于联合用户访问、临时 IAM 用户权限、跨账户访问、跨服务访问以及在 Amazon 上运行的应用程序非常有用。 EC2有关更多信息，请参阅《IAM 用户指南》中的 IAM 中的跨账户资源访问。

使用策略管理访问

您可以 Amazon 通过创建策略并将其附加到 Amazon 身份或资源来控制中的访问权限。策略定义了与身份或资源关联时的权限。 Amazon 在委托人提出请求时评估这些政策。大多数策略都以 JSON 文档的 Amazon 形式存储在中。有关 JSON 策略文档的更多信息，请参阅《IAM 用户指南》中的 JSON 策略概述。

管理员使用策略，通过定义哪个主体可以对什么资源以及在什么条件下执行操作，来指定谁有权访问什么内容。

默认情况下，用户和角色没有权限。IAM 管理员创建 IAM 策略并将其添加到角色中，然后用户可以代入这些角色。IAM 策略定义权限，而不考虑您使用哪种方法来执行操作。

基于身份的策略

基于身份的策略是您附加到身份（用户、组或角色）的 JSON 权限策略文档。这些策略控制身份可在什么条件下对哪些资源执行什么操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的使用客户管理型策略定义自定义 IAM 权限。

基于身份的策略可以是内联策略（直接嵌入到单个身份中）或托管策略（附加到多个身份的独立策略）。要了解如何在托管策略和内联策略之间进行选择，请参阅《IAM 用户指南》中的在托管策略与内联策略之间进行选择。

基于资源的策略

Amazon KMS 密钥策略是一种基于资源的策略，用于控制对 KMS 密钥的访问。每个 KMS 密钥都必须有一个密钥政策。您可以使用其他授权机制来允许对 KMS 密钥的访问，但仅在密钥策略允许时才可以这样操作。（您可以使用 IAM policy 拒绝对 KMS 密钥的访问，即使密钥策略未显式允许。）

基于资源的策略是附加到资源（如 KMS 密钥）的 JSON 策略文档，用于控制对该特定资源的访问。基于资源的策略定义了指定的主体可以对该资源执行的操作以及执行操作的条件。您无需在基于资源的策略中指定资源，但必须指定委托人，例如账户、用户、角色、联合用户或 Amazon Web Services 服务。基于资源的策略是位于管理该资源的服务中的内联策略。您不能在基于资源的策略中使用 Amazon IAM 中的`AWSKeyManagementServicePowerUser`托管策略，例如托管策略。

其他策略类型

Amazon 支持其他策略类型，这些策略类型可以设置更常见的策略类型授予的最大权限：

- 权限边界 – 设置基于身份的策略可以授予 IAM 实体的最大权限。有关更多信息，请参阅《 IAM 用户指南》中的 IAM 实体的权限边界。
- 服务控制策略 (SCPs)-在中指定组织或组织单位的最大权限 Amazon Organizations。有关更多信息，请参阅《Amazon Organizations 用户指南》中的服务控制策略。
- 资源控制策略 (RCPs)-设置账户中资源的最大可用权限。有关更多信息，请参阅《Amazon Organizations 用户指南》中的资源控制策略 (RCPs)。
- 会话策略 – 在为角色或联合用户创建临时会话时，作为参数传递的高级策略。有关更多信息，请参阅《IAM 用户指南》中的会话策略。

多个策略类型

当多个类型的策略应用于一个请求时，生成的权限更加复杂和难以理解。要了解在涉及多种策略类型时如何 Amazon 确定是否允许请求，请参阅 IAM 用户指南中的策略评估逻辑。

Amazon KMS 资源

在中 Amazon KMS，主要资源是 Amazon KMS key。 Amazon KMS 还支持别名，这是一种为 KMS 密钥提供友好名称的独立资源。某些 Amazon KMS 操作允许您使用别名来标识 KMS 密钥。

KMS 密钥或别名的每个实例均具有标准格式的唯一 Amazon Resource Name (ARN)。在 Amazon KMS 资源中， Amazon 服务名称为`kms`。

- Amazon KMS key

ARN 格式：

```
arn:Amazon partition name:Amazon service name:Amazon Web Services ##:Amazon Web Services ## ID:key/key ID
```

示例 ARN:

```
arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

- 别名

  ARN 格式：

  ```
  arn:Amazon partition name:Amazon service name:Amazon Web Services ##:Amazon Web Services ## ID:alias/alias name
  ```

  示例 ARN:

  ```
  arn:aws:kms:us-west-2:111122223333:alias/example-alias
  ```

Amazon KMS 提供了一组 API 操作来使用您的 Amazon KMS 资源。有关在 Amazon Web Services 管理控制台 和 Amazon KMS API 操作中识别 KMS 密钥的更多信息，请参阅密钥标识符 (KeyId)。有关 Amazon KMS 操作列表，请参阅 Amazon Key Management Service API 参考。

创建 KMS 密钥

您可以使用CreateKey操作或AWS::KMS::Key Amazon CloudFormation 资源 Amazon KMS keys在 Amazon Web Services 管理控制台、或中创建。在此过程中，您要设置 KMS 密钥的密钥政策，但您可以随时更改该政策。您还可以选择以下值来定义您创建的 KMS 密钥的类型。创建 KMS 密钥后，这些属性无法更改。

KMS 密钥类型

密钥类型是一个决定创建什么类型的加密密钥的属性。 Amazon KMS 提供了三种保护数据的关键类型：

- 高级加密标准（AES）对称密钥

  256 位密钥在 AES 的 Galois 计数器模式 (GCM) 模式下使用，用于对大小小于 4KB 的数据进行身份验证 encryption/decryption 。这是最常见的密钥类型，用于保护应用程序中使用的其他数据加密密钥 Amazon Web Services 服务 ，从而代表您加密您的数据。

- RSA、椭圆曲线或（仅限 SM2 中国区域）非对称密钥

  这些密钥提供各种大小，并支持多种算法。根据算法选择，其可能用于加密和解密、签名和验证或派生共享密钥操作。

- 用于执行 HMAC 散列消息认证码操作的对称密钥

  这些密钥是用于签名和验证操作的 256 位密钥。

  无法以明文形式从服务中导出 KMS 密钥。它们由服务使用的硬件安全模块 (HSMs) 生成，并且只能在其中使用。这是 Amazon KMS 用于确保密钥不被泄露的基本安全属性。

密钥用法

密钥用法属性决定密钥支持的加密操作。KMS 密钥的密钥用法可以是 ENCRYPT_DECRYPT、SIGN_VERIFY、GENERATE_VERIFY_MAC 或 KEY_AGREEMENT。每个 KMS 密钥都只能有一个密钥用法。这遵循了美国国家标准与技术研究院（NIST）特别出版物 800-57 号《密钥管理建议》第 5.2 节“密钥用法”中有关密钥用法的最佳实践建议。将 KMS 密钥用于多种操作类型，会使两种操作的产物更容易受到攻击。

密钥规范

密钥规范 是一种属性，用于表示密钥的加密配置。密钥规范的含义因密钥类型而异。

对于 KMS 密钥，密钥规范将确定 KMS 密钥是对称密钥还是非对称密钥。它还可确定其密钥材料类型，及其支持的算法。

默认密钥规范 SYMMETRIC_DEFAULT 表示一组 256 位的对称加密密钥。有关所有支持的密钥规范的详细说明，请参阅 密钥规范引用。

密钥材料源

密钥材料源是标识 KMS 密钥中密钥材料的来源的 KMS 密钥属性。您在创建 KMS 密钥时选择密钥材料源，并且无法更改。密钥材料的来源会影响 KMS 密钥的安全性、耐久性、可用性、延迟和吞吐量特性。

每个 KMS 密钥的元数据中都包含对其密钥材料的引用。对称加密 KMS 密钥的密钥材料源可能会有所不同。您可以使用 Amazon KMS 生成的密钥材料、在自定义密钥存储中生成的密钥材料，或者导入自己的密钥材料。

默认情况下，每个 KMS 密钥都具有唯一的密钥材料。但是，您可以使用相同的密钥材料创建一组多区域密钥。

KMS 密钥可以具有以下密钥材料来源值之一：AWS_KMS、EXTERNAL（导入的密钥材料）、AWS_CLOUDHSM（密钥存储中的 KMS Amazon CloudHSM 密钥）或EXTERNAL_KEY_STORE（外部密钥存储中的 KMS 密钥）。

主题

- 创建 KMS 密钥的权限
- 选择要创建的 KMS 密钥的类型
- 创建对称加密 KMS 密钥
- 创建非对称 KMS 密钥
- 创建 HMAC KMS 密钥
- 创建多区域主密钥
- 创建多区域副本密钥
- 删除具有导入密钥材料的 KMS 密钥
- 在密钥库中创建 KMS Amazon CloudHSM 密钥
- 在外部密钥存储中创建 KMS 密钥

创建 KMS 密钥的权限

要在控制台中或使用创建 KMS 密钥 APIs，您必须在 IAM 策略中拥有以下权限。在可能的情况下，使用条件键来限制权限。例如，您可以在 IAM 策略中使用 kms: KeySpec 条件密钥来允许委托人仅创建对称加密密钥。

有关创建密钥的委托人的 IAM policy 的示例，请参阅 允许用户创建 KMS 密钥。

> **Note**
>
> 请谨慎授予委托人管理标签和别名的权限。更改标签或别名可以允许或拒绝对客户托管密钥的权限。有关更多信息，请参阅 ABAC for Amazon KMS。

- km CreateKey s: 为必填项。
- km CreateAlias s: 需要在控制台中创建 KMS 密钥，其中每个新的 KMS 密钥都需要一个别名。
- km TagResource s: 需要在创建 KMS 密钥时添加标签。
- CreateServiceLinkedRolei@@ am: 是创建多区域主密钥所必需的。有关更多信息，请参阅 控制对多区域密钥的访问。

创建 KMS 密钥不需要 kms: PutKeyPolicy 权限。`kms:CreateKey` 权限包括设置初始密钥策略的权限。但是，您必须在创建 KMS 密钥时将此权限添加到密钥策略中，以确保您可以控制对 KMS 密钥的访问。另一种方法是使用BypassLockoutSafetyCheck参数，但不建议这样做。

KMS 密钥属于创建密钥的 Amazon 账户。创建 KMS 密钥的 IAM 用户不会被视为密钥的拥有者，且他们不会自动获得使用或管理自己所创建 KMS 密钥的权限。与任何其他主体一样，密钥创建者需要通过密钥策略、IAM policy 或授权获得权限。但是，拥有 `kms:CreateKey` 权限的主体可以设置初始密钥策略，并授予自己使用或管理密钥的权限。

选择要创建的 KMS 密钥的类型

创建的 KMS 密钥类型在很大程度上取决于计划使用 KMS 密钥的方式，以及安全要求和授权要求。KMS 密钥的密钥类型和密钥用法决定了该密钥可以执行的加密操作。每个 KMS 密钥只有一个密钥用法。将 KMS 密钥用于多种操作类型，会使所有操作的产物更容易受到攻击。

要允许委托人仅针对特定的密钥使用创建 KMS 密钥，请使用 k ms: KeyUsage 条件密钥。还可以使用 `kms:KeyUsage` 条件键，允许委托人根据 KMS 密钥的密钥用法对其调用 API 操作。例如，可以允许仅当 KMS 密钥的密钥用法为 SIGN_VERIFY 时禁用 KMS 密钥的权限。

请根据使用案例，遵照以下指南确定所需的 KMS 密钥类型。

加密和解密数据

对于需要加密和解密数据的大多数使用案例，使用对称 KMS 密钥。 Amazon KMS 使用的对称加密算法快速、高效，并可确保数据的机密性和真实性。它支持具有附加身份验证数据 (AAD) 的身份验证加密，这些数据定义为加密上下文。这种类型的 KMS 密钥要求加密数据的发送者和接收者都具有有效的通话 Amazon 凭证 Amazon KMS。

如果您的用例要求无法呼叫的用户在外部 Amazon 进行加密 Amazon KMS，那么非对称 KMS 密钥是一个不错的选择。您可以分发非对称 KMS 密钥的公有密钥，以允许这些用户对数据进行加密。需要解密该数据的应用程序，可以在 Amazon KMS内部使用非对称 KMS 密钥的私有密钥。

签署消息并验证签名

要签署消息并验证签名，必须使用非对称 KMS 密钥。您可以将 KMS 密钥与表示 RSA 密钥对、椭圆曲线 (ECC) 密钥对、ML-DSA 密钥对或密钥对的密钥规格一起使用（仅限中国区域）。 SM2 选择哪种密钥规范由想要使用的签名算法决定。推荐使用 ECC 密钥对支持的 ECDSA 签名算法，而不是 RSA 签名算法。从 RSA 或 ECC 密钥迁移到后量子密钥时，请使用 ML-DSA 密钥对。不过，您可能需要使用特定的密钥规范和签名算法来支持在 Amazon之外验证签名的用户。

使用非对称密钥对加密

要使用非对称密钥对加密数据，必须使用具有 RSA 密钥规范或密钥规范的非对称 KMS 密SM2 钥（仅限中国区域）。要使用 KMS 密钥对的公有密钥为 Amazon KMS 中的数据进行加密，请使用 Encrypt 操作。您也可以下载公钥并与需要在外部加密数据的各方共享 Amazon KMS。

下载非对称 KMS 密钥的公有密钥后，可以在 Amazon KMS外部使用该密钥。但它不再受保护 KMS 密钥的安全控制措施的约束 Amazon KMS。例如，您不能使用 Amazon KMS 密钥策略或授权来控制公钥的使用。您也无法控制密钥是否仅用于使用支持的加密算法进行加密和解密。 Amazon KMS 有关更多详细信息，请参阅下载公有密钥的特殊注意事项。

要解密使用外部公钥加密的数据，请调用 Dec rypt 操作。 Amazon KMS如果使用 `SIGN_VERIFY` 的密钥用法通过 KMS 密钥中的公有密钥对数据进行加密，则 `Decrypt` 操作会失败。如果使用 Amazon KMS 不支持您选择的密钥规范的算法对其进行加密，它也会失败。有关密钥规范和支持算法的更多信息，请参阅 密钥规范引用。

为避免这些错误，任何在外部使用公钥的人 Amazon KMS 都必须存储密钥配置。 Amazon KMS 控制台和GetPublicKey响应提供了共享公钥时必须包含的信息。

派生共享密钥

要派生共享密钥，请使用带有 NIST 标准椭圆曲线的 KMS 密钥或（仅限SM2中国区域）密钥材料。 Amazon KMS 使用椭圆曲线密码学辅助因子 Diffie-Hellman Priman Primitive (ECDH) 通过从两个对等体的椭圆曲线公私钥对中得出共享密钥来建立密钥协议。您可以使用DeriveSharedSecret操作返回的原始共享密钥来派生对称密钥，该密钥可以加密和解密在双方之间发送的数据，或者生成和验证。 HMACs Amazon KMS 建议您在使用原始共享密钥派生对称密钥时遵循 NIST 关于密钥派生的建议。

生成并验证 HMAC 代码

要生成和验证散列消息认证码，请使用 HMAC 密钥。在中创建 HMAC 密钥时 Amazon KMS，Amazon KMS 会创建和保护您的密钥材料，并确保对密钥使用正确的 MAC 算法。HMAC 代码也可以用作伪随机数，在某些情况下用于对称签名和令牌化。

HMAC KMS 密钥是对称密钥。在 Amazon KMS 控制台中创建 HMAC KMS 密钥时，选择 `Symmetric` 密钥类型。

与 Amazon 服务一起使用

要创建 KMS 密钥以与集成的Amazon 服务一起使用 Amazon KMS，请查阅该服务的文档。Amazon 加密数据的服务需要对称加密 KMS 密钥。

除上述注意事项外，KMS 密钥加密操作的密钥规范不同，其价格和请求限额也不同。有关 Amazon KMS 定价的信息，请参阅 Amazon Key Management Service 定价。有关请求配额的信息，请参阅 请求配额。

创建对称加密 KMS 密钥

本主题介绍如何创建基本 KMS 密钥，这是针对单个区域的对称加密 KMS 密钥，密钥材料来自 Amazon KMS于。您可以在 Amazon Web Services 服务中使用此 KMS 密钥保护您的资源。

您可以在 Amazon KMS 控制台、使用 CreateKeyAPI 或使用AWS::KMS::Key Amazon CloudFormation 模板创建对称加密 KMS 密钥。

默认密钥规范 SYMMETRIC_DEFAULT 是对称加密 KMS 密钥的密钥规范。当您在 Amazon KMS 控制台中选择 Symmetric 密钥类型以及加密和解密密钥用法时，它会选择密钥规范。`SYMMETRIC_DEFAULT`在CreateKey操作中，如果您未指定`KeySpec`值，则会选择 SYMMETRIC_DEFAULT。如果您没有理由使用其他密钥规范，SymMETIC_DEFAULT 是个不错的选择。

有关应用于 KMS 密钥的配额的信息，请参阅 限额。

使用控制 Amazon KMS 台

您可以使用创建 Amazon KMS keys （ KMS 密钥 ）。 Amazon Web Services 管理控制台

> **Important**
>
> 不要在别名、描述或标签中包含机密或敏感信息。这些字段可能以纯文本形式出现在 CloudTrail 日志和其他输出中。

1. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/km s 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择客户托管密钥。
4. 选择创建密钥。
5. 要创建对称加密 KMS 密钥，请为 Key type （ 密钥类型 ）选择 Symmetric （ 对称 ）。
6. 在密钥用法中，系统已为您选择加密和解密选项。
7. 选择下一步。
8. 键入 KMS 密钥的别名。别名名称不能以 **aws/** 开头。该**aws/**前缀由 Amazon Web Services 保留，用于 Amazon 托管式密钥 在您的账户中表示。

   > **Note**
   >
   > 添加、删除或更新别名可以允许或拒绝对 KMS 密钥的权限。有关详细信息，请参阅 ABAC for Amazon KMS 和 使用别名控制对 KMS 密钥的访问。

   别名是一个显示名称，您可以使用它来标识 KMS 密钥。我们建议您选择一个别名，用来指示您计划保护的数据类型或计划与 KMS 密钥搭配使用的应用程序。

   在 Amazon Web Services 管理控制台中创建 KMS 密钥时需要别名。当您使用CreateKey操作时，它们是可选的。
9. （ 可选 ）为 KMS 密钥键入描述。

现在，除非密钥状态为 `Pending Deletion` 或 `Pending Replica Deletion`，否则您可以随时添加描述或更新描述。要添加、更改或删除现有客户托管密钥的描述，请在 Amazon Web Services 管理控制台 或使用UpdateKeyDescription操作中编辑 KMS 密钥的详细信息页面上的描述。

10. （可选）键入标签键和一个可选标签值。要向 KMS 密钥添加多个标签，请选择 Add tag（添加标签）。

> **Note**
>
> 标记或取消标记 KMS 密钥可以允许或拒绝对 KMS 密钥的权限。有关详细信息，请参阅 ABAC for Amazon KMS 和 使用标签控制对 KMS 密钥的访问。

向 Amazon 资源添加标签时， Amazon 会生成一份成本分配报告，其中包含按标签汇总的使用量和成本。标签还可以用来控制对 KMS 密钥的访问。有关轮换 KMS 密钥的信息，请参阅 标签在 Amazon KMS 和 ABAC for Amazon KMS。

11. 选择下一步。
12. 选择可管理 KMS 密钥的 IAM 用户和角色。

> **注意**
>
> 此密钥策略赋予了对此 KMS 密钥的 Amazon Web Services 账户 完全控制权。此控制权允许账户管理员使用 IAM policy 授予其他主体管理 KMS 密钥的权限。有关更多信息，请参阅 the section called “默认密钥策略”。
> IAM 最佳实践不鼓励使用具有长期凭证的 IAM 用户。而应尽可能使用提供临时凭证的 IAM 角色。有关更多信息，请参阅《IAM 用户指南》中的 IAM 安全最佳实践。
> Amazon KMS 控制台在语句标识符下将密钥管理员添加到密钥策略中`"Allow access for Key Administrators"`。如果修改此语句标识符，则可能会影响控制台显示您对该语句所做修改的方式。

13. （可选）要阻止选定 IAM 用户和角色删除此 KMS 密钥，请在页面底部的密钥删除部分中，清除允许密钥管理员删除此密钥复选框。
14. 选择下一步。
15. 选择可在加密操作中使用密钥的 IAM 用户和角色

> **注意**
>
> IAM 最佳实践不鼓励使用具有长期凭证的 IAM 用户。而应尽可能使用提供临时凭证的 IAM 角色。有关更多信息，请参阅《IAM 用户指南》中的 IAM 安全最佳实践。
> Amazon KMS 控制台在语句标识符"Allow use of the key"和下将密钥用户添加到密钥策略中"Allow attachment of persistent resources"。如果修改这些语句标识符，则可能会影响控制台显示您对该语句所做修改的方式。

16. （可选）您可以允许其他人使用 Amazon Web Services 账户 此 KMS 密钥进行加密操作。为此，请在页面底部的 Amazon Web Services 账户“其他” 部分中，选择 “添加另一个”， Amazon Web Services 账户然后输入外部账户的 Amazon Web Services 账户 标识号。要添加多个外部账户，请重复此步骤。

> **Note**
>
> 要允许外部账户中的委托人使用 KMS 密钥，外部账户的管理员必须创建提供这些权限的 IAM policy。有关更多信息，请参阅 允许其他账户中的用户使用 KMS 密钥。

17. 选择下一步。
18. 检查密钥的密钥策略语句。要对密钥策略进行更改，请选择编辑。
19. 选择下一步。
20. 检视您选择的密钥设置。您仍然可以返回并更改所有设置。
21. 选择 Finish（完成）以创建 KMS 密钥。

使用 Amazon KMS API

您可以使用该CreateKey操作来创建 Amazon KMS keys 所有类型。这些示例使用 Amazon Command Line Interface（Amazon CLI）。有关使用多种编程语言的示例，请参阅CreateKey与 Amazon SDK 或 CLI 配合使用。

> **Important**
>
> 不要在 Description 或 Tags 字段中包含机密或敏感信息。这些字段可能以纯文本形式出现在 CloudTrail 日志和其他输出中。

以下操作会在由 Amazon KMS生成的密钥材料提供支持的单个区域中创建对称加密密钥。该操作没有必需参数。不过，您可能还希望使用 Policy 参数指定密钥策略。您可以随时更改密钥策略 (PutKeyPolicy) 并添加可选元素，例如描述和标签。您还可以创建非对称密钥、多区域密钥、具有导入密钥材料的密钥以及自定义密钥存储中的密钥。要创建用于客户端加密的数据密钥，请使用GenerateDataKey操作。

该CreateKey操作不允许您指定别名，但您可以使用该CreateAlias操作为新 KMS 密钥创建别名。

以下示例显示的是在没有任何参数的情况下调用 CreateKey 操作。此命令使用所有默认值。它将创建一个具有 Amazon KMS生成的密钥材料的对称加密 KMS 密钥。

```
$ aws kms create-key
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "KeySpec": "SYMMETRIC_DEFAULT",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1502910355.475,
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "MultiRegion": false
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
    }
}
```

如果您不为新 KMS 密钥指定密钥策略，则 CreateKey 应用的默认密钥策略会不同于在您使用控制台创建新 KMS 密钥时控制台应用的默认密钥策略。

例如，此GetKeyPolicy操作调用将返回CreateKey适用的密钥策略。它授予对 KMS 密钥的 Amazon Web Services 账户 访问权限，并允许其为 KMS 密钥创建 Amazon Identity and Access Management (IAM) 策略。有关 KMS 密钥的 IAM policy 和密钥策略的详细信息，请参阅 KMS 密钥访问权限和权限

```
$ aws kms get-key-policy --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --policy-name
 default --output text
```

JSON

```
{
  "Version":"2012-10-17",
  "Id" : "key-default-1",
  "Statement" : [ {
    "Sid" : "EnableIAMUserPermissions",
    "Effect" : "Allow",
    "Principal" : {
      "AWS" : "arn:aws:iam::111122223333:root"
    },
    "Action" : "kms:*",
    "Resource" : "*"
  } ]
}
```

创建非对称 KMS 密钥

您可以在 Amazon KMS 控制台、使用 CreateKeyAPI 或使用AWS::KMS::Key Amazon CloudFormation 模板创建非对称 KMS 密钥。非对称 KMS 密钥表示可用于加密、签名或派生共享密钥的公有密钥和私有密钥对。私钥保留在里面 Amazon KMS。要下载公钥以供外部使用 Amazon KMS，请参阅下载公有密钥。

创建非对称 KMS 密钥时，必须选择密钥规范。通常，选择哪个密钥规范取决于法规、安全或业务要求。也可能受需要加密或签名的消息大小的影响。一般来说，加密密钥越长，对暴力攻击的抵抗力越强。有关所有支持的密钥规范的详细说明，请参阅密钥规范引用。

Amazon 与集成的服务 Amazon KMS 不支持非对称 KMS 密钥。如果要创建 KMS 密钥来加密您在 Amazon 服务中存储或管理的数据，请创建对称加密 KMS 密钥。

有关创建 KMS 密钥所需权限的信息，请参阅 创建 KMS 密钥的权限。

使用控制 Amazon KMS 台

您可以使用创建非对称 Amazon KMS keys （ KMS 密钥 ）。 Amazon Web Services 管理控制台 每个非对称 KMS 密钥表示一个公有密钥和私有密钥对。

> **Important**
>
> 不要在别名、描述或标签中包含机密或敏感信息。这些字段可能以纯文本形式出现在 CloudTrail 日志和其他输出中。

1. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/km s 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择客户托管密钥。
4. 选择创建密钥。
5. 要创建非对称 KMS 密钥，请在 Key type（密钥类型）中选择 Asymmetric（非对称）。
6. 要创建用于公有密钥加密的非对称 KMS 密钥，请在 Key usage（密钥用法）中选择 Encrypt and decrypt（加密和解密）。

   要创建用于签署消息和验证签名的非对称 KMS 密钥，请在密钥用法中选择签名和验证。

   要创建用于派生共享密钥的非对称 KMS 密钥，请在密钥用法中，选择密钥协议。

   有关选择密钥用法值的帮助信息，请参阅选择要创建的 KMS 密钥的类型。
7. 为非对称 KMS 密钥选择规范（密钥规范）。
8. 选择下一步。
9. 为 KMS 密钥键入别名。别名名称不能以 **aws/** 开头。**aws/** 前缀由 Amazon Web Services 预留，用于在您的账户中表示 Amazon 托管式密钥 。

   别名是一个友好的名称，可用于在控制台和某些控制台中识别 KMS 密钥 Amazon KMS APIs。我们建议您选择一个别名，用来指示您计划保护的数据类型或计划与 KMS 密钥搭配使用的应用程序。

   在 Amazon Web Services 管理控制台中创建 KMS 密钥时需要别名。使用CreateKey操作时无法指定别名，但可以使用控制台或CreateAlias操作为现有 KMS 密钥创建别名。有关更多信息，请参阅 中的别名 Amazon KMS。
10. （可选）为 KMS 密钥键入描述。

    输入一个描述，用来说明您计划保护的数据类型或计划与 KMS 密钥配合使用的应用程序。

现在，除非密钥状态为 `Pending Deletion` 或 `Pending Replica Deletion`，否则您可以随时添加描述或更新描述。要添加、更改或删除现有客户托管密钥的描述，请在 Amazon Web Services 管理控制台 或使用UpdateKeyDescription操作中编辑 KMS 密钥的详细信息页面上的描述。

11. （可选）键入标签键和一个可选标签值。要向 KMS 密钥添加多个标签，请选择 Add tag（添加标签）。

    向 Amazon 资源添加标签时， Amazon 会生成一份成本分配报告，其中包含按标签汇总的使用量和成本。标签还可以用来控制对 KMS 密钥的访问。有关轮换 KMS 密钥的信息，请参阅 标签在 Amazon KMS 和 ABAC for Amazon KMS。

12. 选择下一步。

13. 选择可管理 KMS 密钥的 IAM 用户和角色。

    > **注意**
    >
    > 此密钥策略赋予了对此 KMS 密钥的 Amazon Web Services 账户 完全控制权。此控制权允许账户管理员使用 IAM policy 授予其他主体管理 KMS 密钥的权限。有关更多信息，请参阅 the section called “默认密钥策略”。
    > IAM 最佳实践不鼓励使用具有长期凭证的 IAM 用户。而应尽可能使用提供临时凭证的 IAM 角色。有关更多信息，请参阅《IAM 用户指南》中的 IAM 安全最佳实践。
    > Amazon KMS 控制台在语句标识符下将密钥管理员添加到密钥策略中`"Allow access for Key Administrators"`。如果修改此语句标识符，则可能会影响控制台显示您对该语句所做修改的方式。

14. （可选）要阻止选定 IAM 用户和角色删除此 KMS 密钥，请在页面底部的密钥删除部分中，清除允许密钥管理员删除此密钥复选框。

15. 选择下一步。

16. 选择可将 KMS 密钥用于加密操作的 IAM 用户和角色。

    > **注意**
    >
    > IAM 最佳实践不鼓励使用具有长期凭证的 IAM 用户。而应尽可能使用提供临时凭证的 IAM 角色。有关更多信息，请参阅《IAM 用户指南》中的 IAM 安全最佳实践。

Amazon KMS 控制台在语句标识符"Allow use of the key"和下将密钥用户添加到密钥策略中"Allow attachment of persistent resources"。如果修改这些语句标识符，则可能会影响控制台显示您对该语句所做修改的方式。

17. （可选）您可以允许其他人使用 Amazon Web Services 账户 此 KMS 密钥进行加密操作。为此，请在页面底部的其他 Amazon Web Services 账户部分中选择添加其他 Amazon Web Services 账户，然后输入外部账户的 Amazon Web Services 账户 标识号。要添加多个外部账户，请重复此步骤。

> **Note**
>
> 若要允许外部账户中的主体使用 KMS 密钥，外部账户的管理员必须创建提供这些权限的 IAM policy。有关更多信息，请参阅 允许其他账户中的用户使用 KMS 密钥。

18. 选择下一步。
19. 检查密钥的密钥策略语句。要对密钥策略进行更改，请选择编辑。
20. 选择下一步。
21. 检视您选择的密钥设置。您仍然可以返回并更改所有设置。
22. 选择 Finish（完成）以创建 KMS 密钥。

使用 Amazon KMS API

您可以使用该CreateKey操作来创建非对称 Amazon KMS key的。这些示例使用 Amazon Command Line Interface (Amazon CLI)，但您可以使用任何受支持的编程语言。

创建非对称 KMS 密钥时，必须指定 KeySpec 参数，该参数决定了所创建的密钥类型。此外，还必须指定 KeyUsage 值是 ENCRYPT_DECRYPT、SIGN_VERIFY 还是 KEY_AGREEMENT。创建 KMS 密钥后，这些属性无法更改。

该CreateKey操作不允许您指定别名，但您可以使用该CreateAlias操作为新 KMS 密钥创建别名。

> **Important**
>
> 不要在 Description 或 Tags 字段中包含机密或敏感信息。这些字段可能以纯文本形式出现在 CloudTrail 日志和其他输出中。

创建非对称 KMS 密钥对用于公共加密

以下示例使用 CreateKey 操作，创建一个 4096 位 RSA 密钥的非对称 KMS 密钥，用于公有密钥加密。

```
$ aws kms create-key --key-spec RSA_4096 --key-usage ENCRYPT_DECRYPT
{
    "KeyMetadata": {
        "KeyState": "Enabled",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "Description": "",
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "CreationDate": 1569973196.214,
        "MultiRegion": false,
        "KeySpec": "RSA_4096",
        "CustomerMasterKeySpec": "RSA_4096",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "EncryptionAlgorithms": [
            "RSAES_OAEP_SHA_1",
            "RSAES_OAEP_SHA_256"
        ],
        "AWSAccountId": "111122223333",
        "Origin": "AWS_KMS",
        "Enabled": true
    }
}
```

创建非对称 KMS 密钥对用于签名和验证

以下示例命令创建一个非对称 KMS 密钥，表示一对用于签名和验证的 ECC 密钥。不能创建用于加密和解密的椭圆曲线密钥对。

```
$ aws kms create-key --key-spec ECC_NIST_P521 --key-usage SIGN_VERIFY
{
    "KeyMetadata": {
        "KeyState": "Enabled",
        "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
        "CreationDate": 1570824817.837,
        "Origin": "AWS_KMS",
        "SigningAlgorithms": [
            "ECDSA_SHA_512"
        ],
```

```
        "Arn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
        "AWSAccountId": "111122223333",
        "KeySpec": "ECC_NIST_P521",
        "CustomerMasterKeySpec": "ECC_NIST_P521",
        "KeyManager": "CUSTOMER",
        "Description": "",
        "Enabled": true,
        "MultiRegion": false,
        "KeyUsage": "SIGN_VERIFY"
    }
}
```

创建非对称 KMS 密钥对，用于派生共享密钥

以下示例命令创建一个非对称 KMS 密钥，表示一对用于派生共享密钥的 ECDH 密钥。不能创建用于加密和解密的椭圆曲线密钥对。

```
$ aws kms create-key --key-spec ECC_NIST_P256 --key-usage KEY_AGREEMENT
{
    "KeyMetadata": {
        "AWSAccountId": "111122223333",
        "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
        "Arn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
        "CreationDate": "2023-12-27T19:10:15.063000+00:00",
        "Enabled": true,
        "Description": "",
        "KeyUsage": "KEY_AGREEMENT",
        "KeyState": "Enabled",
        "Origin": "AWS_KMS",
        "KeyManager": "CUSTOMER",
        "CustomerMasterKeySpec": "ECC_NIST_P256",
        "KeySpec": "ECC_NIST_P256",
        "KeyAgreementAlgorithms": [
            "ECDH"
        ],
        "MultiRegion": false
    }
}
```

创建 HMAC KMS 密钥

您可以在 Amazon KMS 控制台、使用 CreateKeyAPI 或使用AWS::KMS::Key Amazon CloudFormation 模板创建 HMAC KMS 密钥。

创建 HMAC KMS 密钥时，必须选择密钥规范。 Amazon KMS 支持 HMAC KMS 密钥的多种密钥规格。您选择的密钥规范可能取决于法规、安全或业务要求。一般来说，密钥越长，对暴力攻击的抵抗力越强。

有关创建 KMS 密钥所需权限的信息，请参阅 创建 KMS 密钥的权限。

使用控制 Amazon KMS 台

您可以使用创建 HMAC KMS 密钥。 Amazon Web Services 管理控制台 HMAC KMS 密钥是对称密钥，其密钥用法为Generate and verify MAC（生成并验证 MAC）。您也可以创建多区域 HMAC 密钥。

1. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/km s 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择客户托管密钥。
4. 选择创建密钥。
5. 对于 Key type (密钥类型)，选择 Symmetric (对称)。

   HMAC KMS 密钥是对称密钥。您可以使用相同的密钥生成和验证 HMAC 标签。
6. 对于 Key usage（密钥用法），请选择 Generate and verify MAC（生成并验证 MAC）。

   生成并验证 MAC 是 HMAC KMS 密钥的唯一有效密钥用法。

   > **Note**
   >
   > Key usage（密钥用法）仅在所选区域支持 HMAC KMS 密钥时才会显示对称密钥。
7. 为 HMAC KMS 密钥选择规范（密钥规范）。

   您选择的密钥规范可取决于法规、安全或业务要求。一般来说，较长的密钥更安全。
8. 要创建多区域 主 HMAC 密钥，在 Advanced options（高级选项）下，选择 Multi-Region key（多区域密钥）。您为此 KMS 密钥定义的共享属性（例如密钥类型和密钥用法）将与其副本密钥共享。

您无法使用该过程创建副本密钥。要创建多区域副本 HMAC 密钥，请按照创建副本密钥的说明。

9. 选择下一步。
10. 为 KMS 密钥键入别名。别名名称不能以 **aws/** 开头。**aws/** 前缀由 Amazon Web Services 预留，用于在您的账户中表示 Amazon 托管式密钥 。

    我们建议您使用将 KMS 密钥标识为 HMAC 密钥的别名，例如 `HMAC/test-key`。这将使您更轻松地在控制台中识别 HMAC 密钥，您可以在 Amazon KMS 控制台中按标签和别名对密钥进行排序和筛选，但不能按密钥规格或密钥使用情况对密钥进行排序和筛选。

    在 Amazon Web Services 管理控制台中创建 KMS 密钥时需要别名。使用CreateKey操作时无法指定别名，但可以使用控制台或CreateAlias操作为现有 KMS 密钥创建别名。有关更多信息，请参阅 中的别名 Amazon KMS。
11. （可选）为 KMS 密钥输入描述。

    输入一个描述，用来说明您计划保护的数据类型或计划与 KMS 密钥配合使用的应用程序。

    现在，除非密钥状态为 `Pending Deletion` 或 `Pending Replica Deletion`，否则您可以随时添加描述或更新描述。要添加、更改或删除现有客户托管密钥的描述，请在 Amazon Web Services 管理控制台 或使用UpdateKeyDescription操作中编辑 KMS 密钥的 Amazon Web Services 管理控制台 详细信息页面上的描述。
12. （可选）输入标签键和一个可选标签值。要向 KMS 密钥添加多个标签，请选择 Add tag（添加标签）。

    考虑添加将密钥识别为 HMAC 密钥的标签，例如 `Type=HMAC`。这将使您更轻松地在控制台中识别 HMAC 密钥，您可以在 Amazon KMS 控制台中按标签和别名对密钥进行排序和筛选，但不能按密钥规格或密钥使用情况对密钥进行排序和筛选。

    向 Amazon 资源添加标签时， Amazon 会生成一份成本分配报告，其中包含按标签汇总的使用量和成本。标签还可以用来控制对 KMS 密钥的访问。有关轮换 KMS 密钥的信息，请参阅 标签在 Amazon KMS 和 ABAC for Amazon KMS。
13. 选择下一步。
14. 选择可管理 KMS 密钥的 IAM 用户和角色。

> **注意**
>
> 此密钥策略赋予了对此 KMS 密钥的 Amazon Web Services 账户 完全控制权。此控制权允许账户管理员使用 IAM policy 授予其他主体管理 KMS 密钥的权限。有关更多信息，请参阅 the section called “默认密钥策略”。
> IAM 最佳实践不鼓励使用具有长期凭证的 IAM 用户。而应尽可能使用提供临时凭证的 IAM 角色。有关更多信息，请参阅《IAM 用户指南》中的 IAM 安全最佳实践。
> Amazon KMS 控制台在语句标识符下将密钥管理员添加到密钥策略中`"Allow access for Key Administrators"`。如果修改此语句标识符，则可能会影响控制台显示您对该语句所做修改的方式。

15. （可选）要阻止选定 IAM 用户和角色删除此 KMS 密钥，请在页面底部的密钥删除部分中，清除允许密钥管理员删除此密钥复选框。
16. 选择下一步。
17. 选择可将 KMS 密钥用于加密操作的 IAM 用户和角色。

> **注意**
>
> IAM 最佳实践不鼓励使用具有长期凭证的 IAM 用户。而应尽可能使用提供临时凭证的 IAM 角色。有关更多信息，请参阅《IAM 用户指南》中的 IAM 安全最佳实践。
> Amazon KMS 控制台在语句标识符`"Allow use of the key"`和下将密钥用户添加到密钥策略中`"Allow attachment of persistent resources"`。如果修改这些语句标识符，则可能会影响控制台显示您对该语句所做修改的方式。

18. （可选）您可以允许其他人使用 Amazon Web Services 账户 此 KMS 密钥进行加密操作。为此，请在页面底部的其他 Amazon Web Services 账户部分中选择添加其他 Amazon Web Services 账户，然后输入外部账户的 Amazon Web Services 账户 标识号。要添加多个外部账户，请重复此步骤。

> **Note**
>
> 要允许外部账户中的委托人使用 KMS 密钥，外部账户的管理员必须创建提供这些权限的 IAM policy。有关更多信息，请参阅 允许其他账户中的用户使用 KMS 密钥。

19. 选择下一步。
20. 检查密钥的密钥策略语句。要对密钥策略进行更改，请选择编辑。

21. 选择下一步。
22. 检视您选择的密钥设置。您仍然可以返回并更改所有设置。
23. 选择 Finish（完成）以创建 HMAC KMS 密钥。

使用 Amazon KMS API

您可以使用该CreateKey操作创建 HMAC KMS 密钥。这些示例使用 Amazon Command Line Interface (Amazon CLI)，但您可以使用任何受支持的编程语言。

创建 HMAC KMS 密钥时，必须指定 KeySpec 参数，该参数决定了 KMS 密钥的类型。另外，您必须指定 GENERATE_VERIFY_MAC 的 KeyUsage 值，尽管它是 HMAC 密钥的唯一有效密钥用法值。要创建多区域 HMAC KMS 密钥，添加值为 true 的 MultiRegion 参数。创建 KMS 密钥后，这些属性无法更改。

该CreateKey操作不允许您指定别名，但您可以使用该CreateAlias操作为新 KMS 密钥创建别名。我们建议您使用将 KMS 密钥标识为 HMAC 密钥的别名，例如 HMAC/test-key。这将使您更容易在控制台中识别 HMAC 密钥，您可以在 Amazon KMS 控制台中按别名对密钥进行排序和筛选，但不能按密钥规格或密钥使用情况进行排序和筛选。

如果您尝试在不支持 HMAC 密钥的情况下创建 HMAC KMS 密钥，则该CreateKey操作会返回 Amazon Web Services 区域 UnsupportedOperationException

以下示例使用 CreateKey 操作来创建 512 位的 HMAC KMS 密钥。

```
$ aws kms create-key --key-spec HMAC_512 --key-usage GENERATE_VERIFY_MAC
{
    "KeyMetadata": {
        "KeyState": "Enabled",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "Description": "",
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "CreationDate": 1669973196.214,
        "MultiRegion": false,
        "KeySpec": "HMAC_512",
        "CustomerMasterKeySpec": "HMAC_512",
        "KeyUsage": "GENERATE_VERIFY_MAC",
        "MacAlgorithms": [
            "HMAC_SHA_512"
```

```
        ],
        "AWSAccountId": "111122223333",
        "Origin": "AWS_KMS",
        "Enabled": true
    }
}
```

创建多区域主密钥

您可以在 Amazon KMS 控制台中或使用 Amazon KMS API 创建多区域主密钥。您可以在任何 Amazon KMS 支持多区域密钥 Amazon Web Services 区域 的地方创建主密钥。

要创建多区域主密钥，委托人需要与创建任何 KMS 密钥相同的权限，包括 IAM 策略中的 kms: CreateKey 权限。委托人还需要 iam: CreateServiceLinkedRole 权限。您可以使用 kms: MultiRegionKeyType 条件密钥来允许或拒绝创建多区域主密钥的权限。

> **Note**
>
> 创建多区域主密钥时，请谨慎考虑您选择用于管理和使用密钥的 IAM 用户和角色。IAM policy 可以向其他 IAM 用户和角色授予管理 KMS 密钥的权限。
> IAM 最佳实践不鼓励使用具有长期凭证的 IAM 用户。而应尽可能使用提供临时凭证的 IAM 角色。有关更多信息，请参阅《IAM 用户指南》中的 IAM 安全最佳实践。

使用控制 Amazon KMS 台

要在 Amazon KMS 控制台中创建多区域主密钥，请使用与创建任何 KMS 密钥相同的过程。您可以在高级选项中选择多区域密钥。有关完整说明，请参阅创建 KMS 密钥。

> **Important**
>
> 不要在别名、描述或标签中包含机密或敏感信息。这些字段可能以纯文本形式出现在 CloudTrail 日志和其他输出中。

1. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/km s 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。

3. 在导航窗格中，选择客户托管密钥。
4. 选择创建密钥。
5. 选择对称或非对称密钥类型。对称密钥为原定设置。

   您可以创建多区域对称密钥和非对称密钥，包括对称的多区域 HMAC KMS 密钥。
6. 选择您的密钥使用方法。Encrypt and decrypt（加密和解密）是原定设置。

   有关帮助信息，请参阅 创建 KMS 密钥、the section called “创建非对称 KMS 密钥” 或 the section called “创建 HMAC KMS 密钥”。
7. 展开 Advanced options (高级选项)。
8. 在密钥材料来源下，要 Amazon KMS 生成您的主密钥和副本密钥将共享的密钥材料，请选择 KMS。如果您 将密钥材料导入 到主密钥和副本密钥中，请选择 External (Import key material) [外部（导入密钥材料）]。
9. 在区域性下，选择多区域密钥。

   创建 KMS 密钥之后，您无法再更改此设置。
10. 为主密钥键入别名。

    别名不是多区域密钥的共享属性。您可以为多区域主密钥及其副本指定相同的别名或不同的别名。Amazon KMS 不会同步多区域密钥的别名。

    > **Note**
    >
    > 添加、删除或更新别名可以允许或拒绝对 KMS 密钥的权限。有关详细信息，请参阅 ABAC for Amazon KMS 和 使用别名控制对 KMS 密钥的访问。
11. （可选）键入主密钥的描述。

    描述不是多区域密钥的共享属性。您可以为多区域主密钥及其副本提供相同的描述或不同的描述。Amazon KMS 不同步多区域密钥的密钥描述。
12. （可选）键入标签键和一个可选标签值。要向主密钥分配多个标签，请选择 Add tag（添加标签）。

    标签不是多区域密钥的共享属性。您可以为多区域主密钥及其副本密钥指定相同的标签或不同的标签。 Amazon KMS 不同步多区域密钥的标签。您可以随时更改 KMS 密钥上的标签。

> **Note**
>
> 标记或取消标记 KMS 密钥可以允许或拒绝对 KMS 密钥的权限。有关详细信息，请参阅 ABAC for Amazon KMS 和 使用标签控制对 KMS 密钥的访问。

13. 选择可管理主密钥的 IAM 用户和角色。

> **注意**
>
> - 此步骤将开始为主密钥创建密钥策略的过程。密钥策略不是多区域密钥的共享属性。您可以为多区域主密钥及其副本提供相同的密钥策略或不同的密钥策略。 Amazon KMS 不会同步多区域密钥的密钥策略。条件密钥值必须遵守密钥策略和 IAM policy 的字符和编码规则。
> - 创建多区域主密钥时，可考虑使用控制台生成的默认密钥策略。如果您修改此策略，则控制台将不会在创建副本密钥时提供选择密钥管理员和用户的步骤，也不会添加相应的策略语句。这样，您将需要手动添加这些语句。
> - Amazon KMS 控制台在语句标识符下将密钥管理员添加到密钥策略中`"Allow access for Key Administrators"`。如果修改此语句标识符，则可能会影响控制台显示您对该语句所做修改的方式。

14. （可选）要阻止选定 IAM 用户和角色删除此 KMS 密钥，请在页面底部的密钥删除部分中，清除允许密钥管理员删除此密钥复选框。
15. 选择下一步。
16. 选择可将 KMS 密钥用于加密操作的 IAM 用户和角色。

> **注意**
>
> Amazon KMS 控制台在语句标识符`"Allow use of the key"`和下将密钥用户添加到密钥策略中`"Allow attachment of persistent resources"`。如果修改这些语句标识符，则可能会影响控制台显示您对该语句所做修改的方式。

17. （可选）您可以允许其他人使用 Amazon Web Services 账户 此 KMS 密钥进行加密操作。为此，请在页面底部的其他 Amazon Web Services 账户部分中选择添加其他 Amazon Web Services 账户，然后输入外部账户的 Amazon Web Services 账户 标识号。要添加多个外部账户，请重复此步骤。

> **Note**
>
> 要允许外部账户中的委托人使用 KMS 密钥，外部账户的管理员必须创建提供这些权限的 IAM policy。有关更多信息，请参阅 允许其他账户中的用户使用 KMS 密钥。

18. 选择下一步。
19. 检查密钥的密钥策略语句。要对密钥策略进行更改，请选择编辑。
20. 选择下一步。
21. 检视您选择的密钥设置。您仍然可以返回并更改所有设置。
22. 选择完成以创建多区域主密钥。

使用 Amazon KMS API

要创建多区域主键，请使用CreateKey操作。使用带 `True` 值的 `MultiRegion` 参数。

例如，以下命令在调用者的主键 Amazon Web Services 区域 (us-east-1) 中创建多区域主键。它接受所有其他属性的默认值，包括密钥策略。多区域主密钥的默认值与所有其他 KMS 密钥的默认值相同，包括默认密钥策略。此过程将创建一个对称加密密钥，即默认 KMS 密钥。

响应包含 `MultiRegion` 元素和 `MultiRegionConfiguration` 元素，其中包含典型的子元素和不含副本密钥的多区域主密钥的值。多区域密钥的密钥 ID 总是以 `mrk-` 开头。

> **Important**
>
> 不要在 `Description` 或 `Tags` 字段中包含机密或敏感信息。这些字段可能以纯文本形式出现在 CloudTrail 日志和其他输出中。

```
$ aws kms create-key --multi-region
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
```

```
        "KeySpec": "SYMMETRIC_DEFAULT",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1606329032.475,
        "Arn": "arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "AWSAccountId": "111122223333",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "MultiRegion": true,
        "MultiRegionConfiguration": {
            "MultiRegionKeyType": "PRIMARY",
            "PrimaryKey": {
                "Arn": "arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                "Region": "us-east-1"
            },
            "ReplicaKeys": [ ]
      }
    }
}
```

创建多区域副本密钥

您可以在 Amazon KMS 控制台、使用ReplicateKey操作或使用AWS::KMS::ReplicaKey Amazon CloudFormation 模板创建多区域副本密钥。您不能使用该CreateKey操作来创建副本密钥。

您可以使用这些步骤复制任何多区域主密钥，包括对称加密 KMS 密钥，非对称 KMS 密钥，或者 HMAC KMS 密钥。

此操作完成后，新的副本密钥具有暂时性的密钥状态 `Creating`。在创建新副本密钥的过程完成几秒后，此密钥状态将变为 `Enabled`（如果您使用导入的密钥材料创建多区域密钥，则变为 `PendingImport`）。当密钥状态为 `Creating` 时，您可以管理该密钥，但不能将其用于加密操作。如果您以编程方式创建和使用副本密钥，请在使用副本密钥之前重试`KMSInvalidStateException`或调`DescribeKey`用检查其`KeyState`值。

如果误删了副本密钥，您可以使用此过程来重新创建副本密钥。如果您在同一区域中复制相同的主密钥，则您创建的新副本密钥将具有与原始副本密钥相同的共享属性。

> **Important**
>
> 不要在别名、描述或标签中包含机密或敏感信息。这些字段可能以纯文本形式出现在 CloudTrail 日志和其他输出中。

要使用 Amazon CloudFormation 模板创建副本密钥，请参阅Amazon CloudFormation 用户指南AWS::KMS::ReplicaKey中的。

步骤 1：选择副本区域

您通常会 Amazon Web Services 区域 根据自己的业务模式和监管要求选择将多区域密钥复制到中。例如，您可以将密钥复制到保留资源的区域中。或者，为了符合灾难恢复要求，您可以将密钥复制到地理位置偏远的区域中。

以下是副本区域的 Amazon KMS 要求。如果您选择的区域不符合这些要求，则尝试复制密钥失败。

- 每个区域一个相关的多区域密钥 — 您不能在与主密钥相同的区域中创建副本密钥，也不能在与主密钥的另一个副本相同的区域中创建副本密钥。

  如果您尝试在已具有该主密钥副本的区域中复制主密钥，则尝试会失败。如果区域中的当前副本密钥处于 `PendingDeletion` 密钥状态，您可以取消副本密钥删除或者等待副本密钥被删除。
- 同一区域中的多个不相关的多区域密钥 — 您可以在同一个区域中拥有多个不相关的多区域密钥。例如，您可以在 `us-east-1` 区域中拥有两个多区域主密钥。每个主密钥在 `us-west-2` 区域中都可以有一个副本密钥。
- 同一分区中的区域 — 副本密钥区域必须位于与主密钥区域相同的 Amazon 分区中。
- 必须启用区域 — 如果某个区域默认已禁用，则无法在该区域中创建任何资源，直到为您的 Amazon Web Services 账户启用该区域为止。

步骤 2：创建副本密钥

> **Note**
>
> 创建副本密钥时，请谨慎考虑您选择用于管理和使用副本密钥的 IAM 用户和角色。IAM policy 可以向其他 IAM 用户和角色授予管理 KMS 密钥的权限。
> IAM 最佳实践不鼓励使用具有长期凭证的 IAM 用户。而应尽可能使用提供临时凭证的 IAM 角色。有关更多信息，请参阅《IAM 用户指南》中的 IAM 安全最佳实践。

使用控制 Amazon KMS 台

在 Amazon KMS 控制台中，您可以在同一个操作中创建多区域主密钥的一个或多个副本。

此过程类似于在控制台中创建标准的单区域 KMS 密钥。但是，由于副本密钥基于主密钥，因此您不能为共享属性选择值，例如密钥规范（对称或非对称）、密钥用法或密钥来源。

您可以指定不共享的属性，包括别名、标签、描述和密钥策略。为方便起见，控制台会显示主密钥的当前属性值，但您可以更改它们。即使您保留了主键值，也 Amazon KMS 不会使这些值保持同步。

> **Important**
>
> 不要在别名、描述或标签中包含机密或敏感信息。这些字段可能以纯文本形式出现在 CloudTrail 日志和其他输出中。

1. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/km s 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择客户托管密钥。
4. 选择多区域主密钥的密钥 ID 或别名。此操作将打开 KMS 密钥的密钥详细信息页面。

   要识别多区域主密钥，请使用右上角的工具图标将 Regionality（区域性）列添加到表中。
5. 选择 Regionality（区域性）选项卡。
6. 在 Related multi-Region keys（相关的多区域密钥）部分中，选择 Create new replica keys（创建新的副本密钥）。

   Related multi-Region keys（相关的多区域密钥）部分显示主密钥及其副本密钥的区域。您可以使用此显示来帮助您为新副本密钥选择区域。
7. 选择一个或多个 Amazon Web Services 区域。此过程将在您选择的每个区域中创建一个副本密钥。

   该菜单仅包含与主键位于同一 Amazon 分区中的区域。已具有相关多区域密钥的区域显示出来，但不可选。您可能没有权限将密钥复制到菜单上的所有区域。

   完成选择 Regions（区域）后，关闭菜单。将显示您选择的区域。要取消复制到某个区域，请选择 Region（区域）名称旁边的 X。
8. 为副本密钥键入别名。

控制台将显示主密钥的当前别名之一，但您可以对其进行更改。您可以为多区域主密钥及其副本密钥指定相同的别名或不同的别名。别名不是多区域密钥的共享属性。 Amazon KMS 不会同步多区域密钥的别名。

添加、删除或更新别名可以允许或拒绝对 KMS 密钥的权限。有关详细信息，请参阅 ABAC for Amazon KMS 和 使用别名控制对 KMS 密钥的访问。

9. （可选）键入副本密钥的描述。

   控制台将显示主密钥的当前描述，但您可以对其进行更改。描述不是多区域密钥的共享属性。您可以为多区域主密钥及其副本提供相同的描述或不同的描述。 Amazon KMS 不同步多区域密钥的密钥描述。

10. （可选）键入标签键和一个可选标签值。要为副本密钥分配多个标签，请选择 Add tag（添加标签）。

    控制台将显示当前附加到主密钥的标签，但您可以更改它们。标签不是多区域密钥的共享属性。您可以为多区域主密钥及其副本指定相同的标签或不同的标签。 Amazon KMS 不会同步多区域密钥的标签。

    标记或取消标记 KMS 密钥可以允许或拒绝对 KMS 密钥的权限。有关详细信息，请参阅 ABAC for Amazon KMS 和 使用标签控制对 KMS 密钥的访问。

11. 选择可管理副本密钥的 IAM 用户和角色。

> **注意**
>
> - 如果您在创建多区域主密钥时修改了默认密钥策略，则控制台将不会在创建副本密钥期间提示您选择密钥管理员或密钥用户（第 11-15 步）。在这种情况下，您需要通过在编辑密钥策略步骤（第 17 步）中选择编辑，手动将密钥管理员和用户所需的权限添加到密钥策略中。
> - 此步骤将开始为副本密钥创建密钥策略的过程。控制台将显示主密钥的当前密钥策略，但您可以对其进行更改。密钥策略不是多区域密钥的共享属性。您可以为多区域主密钥及其副本提供相同的密钥策略或不同的密钥策略。 Amazon KMS 不同步密钥策略。您可以随时更改任何 KMS 密钥的密钥策略。
> - Amazon KMS 控制台在语句标识符下将密钥管理员添加到密钥策略中`"Allow access for Key Administrators"`。如果修改此语句标识符，则可能会影响控制台显示您对该语句所做修改的方式。

12. （可选）要阻止选定 IAM 用户和角色删除此 KMS 密钥，请在页面底部的密钥删除部分中，清除允许密钥管理员删除此密钥复选框。
13. 选择下一步。
14. 选择可将 KMS 密钥用于加密操作的 IAM 用户和角色。

> **备注**
>
> Amazon KMS 控制台在语句标识符"Allow use of the key"和下将密钥用户添加到密钥策略中"Allow attachment of persistent resources"。如果修改这些语句标识符，则可能会影响控制台显示您对该语句所做修改的方式。

15. （可选）您可以允许其他人使用 Amazon Web Services 账户 此 KMS 密钥进行加密操作。为此，请在页面底部的其他 Amazon Web Services 账户部分中选择添加其他 Amazon Web Services 账户，然后输入外部账户的 Amazon Web Services 账户 标识号。要添加多个外部账户，请重复此步骤。

> **Note**
>
> 要允许外部账户中的委托人使用 KMS 密钥，外部账户的管理员必须创建提供这些权限的 IAM policy。有关更多信息，请参阅 允许其他账户中的用户使用 KMS 密钥。

16. 选择下一步。
17. 检查密钥的密钥策略语句。要对密钥策略进行更改，请选择编辑。
18. 选择下一步。
19. 检视您选择的密钥设置。您仍然可以返回并更改所有设置。
20. 选择完成以创建多区域副本密钥。

使用 Amazon KMS API

要创建多区域副本密钥，请使用ReplicateKey操作。您不能使用该CreateKey操作来创建副本密钥。此操作一次创建一个副本密钥。您指定的区域必须符合副本密钥的区域要求。

当您使用 ReplicateKey 操作时，无需为多区域密钥的任何共享属性指定值。共享属性值从主密钥复制并保持同步。但是，您可以为未共享的属性指定值。否则， Amazon KMS 将应用 KMS 密钥的标准默认值，而不是主密钥的值。

> **Note**
>
> 如果您没有为、或Tags参数指定值 DescriptionKeyPolicy，则使用空字符串描述、默认密钥策略且不使用标签 Amazon KMS 创建副本密钥。
> 不要在 Description 或 Tags 字段中包含机密或敏感信息。这些字段可能以纯文本形式出现在 CloudTrail 日志和其他输出中。

例如，以下命令在亚太地区（悉尼）区域 (ap-southeast-2) 中创建一个多区域副本密钥。此副本密钥基于美国东部（弗吉尼亚北部）区域 (us-east-1) 中的主密钥建模，它由 KeyId 参数的值进行标识。此示例接受所有其他属性的默认值，包括密钥策略。

响应描述了新的副本密钥。它包含共享属性的字段，例如 KeyId、KeySpec、KeyUsage 和密钥材料来源 (Origin)。它还包括独立于主密钥的属性，例如 Description、密钥策略 (ReplicaKeyPolicy) 和标签 (ReplicaTags)。

响应还包括主密钥的密钥 ARN 和区域及其所有副本密钥，包括刚刚在 ap-southeast-2 区域中创建的密钥。在此示例中，ReplicaKey 元素表明此主密钥已在欧洲（爱尔兰）区域 (eu-west-1) 复制。

```
$ aws kms replicate-key \
    --key-id arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab \
    --replica-region ap-southeast-2
{
    "ReplicaKeyMetadata": {
        "MultiRegion": true,
        "MultiRegionConfiguration": {
            "MultiRegionKeyType": "REPLICA",
            "PrimaryKey": {
                "Arn": "arn:aws:kms:us-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                "Region": "us-east-1"
            },
            "ReplicaKeys": [
                {
                    "Arn": "arn:aws:kms:ap-southeast-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "ap-southeast-2"
                },
                {
```

```
                    "Arn": "arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "eu-west-1"
                }
            ]
        },
        "AWSAccountId": "111122223333",
        "Arn": "arn:aws:kms:ap-southeast-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "CreationDate": 1607472987.918,
        "Description": "",
        "Enabled": true,
        "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "Origin": "AWS_KMS",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    },
    "ReplicaKeyPolicy": "{\n  \"Version\" : \"2012-10-17\",\n  \"Id\" : \"key-
default-1\",...,
    "ReplicaTags": []
}
```


删除具有导入密钥材料的 KMS 密钥

导入的密钥材料允许您在生成的加密密钥下保护您的 Amazon 资源。以下概述说明了如何将密钥材料导入 Amazon KMS。要了解该过程中每个步骤的更多详细信息，请参阅相应主题。

1. 创建不具有密钥材料的 KMS 密钥 - 源必须是 EXTERNAL。密钥来源为EXTERNAL表示密钥是为导入的密钥材料设计的，因此无法 Amazon KMS 为 KMS 密钥生成密钥材料。在后面的步骤中，您会将自己的密钥材料导入此 KMS 密钥中。

   您导入的密钥材料必须与关联 Amazon KMS 密钥的密钥规范兼容。有关兼容性的更多信息，请参阅the section called “导入密钥材料的要求”。
2. 下载包装公有密钥和导入令牌 – 在完成步骤 1 后，请下载公有密钥和导入令牌。当您的密钥材料导入时，这些物品可以保护您的密钥材料 Amazon KMS。

在此步骤中，您将选择 RSA 包装密钥的类型（“密钥规范”）以及用于加密向 Amazon KMS传输的传输中数据的包装算法。每次导入或重新导入相同的密钥材料时，您可以选择不同的包装密钥规范和包装密钥算法。

3. 加密密钥材料 – 使用在步骤 2 中下载的包装公有密钥加密您在自己的系统上创建的密钥材料。
4. 导入密钥材料 – 上传您在步骤 3 中创建的已加密的密钥材料以及您在步骤 2 中下载的导入令牌。

   在此阶段，您可以设置可选的过期时间。导入的密钥材料过期后，将其 Amazon KMS 删除，KMS 密钥将无法使用。要继续使用该 KMS 密钥，您必须重新导入相同的密钥材料。

   导入操作成功完成后，KMS 密钥的密钥状态将从 `PendingImport` 变为 `Enabled`。现在，您可以在加密操作中使用 KMS 密钥。

Amazon KMS 在创建 KMS 密钥、下载封装公钥和导入令牌以及导入密钥材料时，会在 Amazon CloudTrail 日志中记录一个条目。 Amazon KMS 还会在您删除导入的密钥材料或 Amazon KMS 删除过期的密钥材料时记录一个条目。

导入密钥材料的权限

要使用导入的密钥材料创建和管理 KMS 密钥，用户需要在此过程中执行操作的权限。在您创建 KMS 密钥时，您可以在密钥策略中提供 `kms:GetParametersForImport`、`kms:ImportKeyMaterial` 和 `kms:DeleteImportedKeyMaterial` 权限。在 Amazon KMS 控制台中，当您使用外部密钥材料来源创建密钥时，会自动为密钥管理员添加这些权限。

若要使用导入的密钥材料创建 KMS 密钥，委托人需要以下权限。

- kms: CreateKey（IAM 策略）
  - 要将此权限限制为使用已导入密钥材料的 KMS 密钥，请使用值为 kms: KeyOrigin 策略条件EXTERNAL。

```
{
  "Sid": "CreateKMSKeysWithoutKeyMaterial",
  "Effect": "Allow",
  "Resource": "*",
  "Action": "kms:CreateKey",
  "Condition": {
    "StringEquals": {
      "kms:KeyOrigin": "EXTERNAL"
    }
  }
```

```
}
```

- kms: GetParametersForImport（密钥策略或 IAM 策略）
  - 要将此权限限制为使用特定包装算法和封装密钥规范的请求，请使用 kms: WrappingAlgorithm 和 kms: WrappingKeySpec 策略条件。
- kms: ImportKeyMaterial（密钥策略或 IAM 策略）
  - 要允许或禁止过期的密钥材料并控制过期日期，请使用 kms: ExpirationModel 和 kms: ValidTo 策略条件。

要重新导入导入的密钥材料，委托人需要 kms: GetParametersForImport 和 kms: ImportKeyMaterial 权限。

要删除导入的密钥材料，委托人需要 kms: DeleteImportedKeyMaterial 权限。

例如，要授予示例使用导入的密钥材料管理 KMS 密钥所有方面的 KMSAdminRole 权限，请在 KMS 密钥的密钥策略中加入如下所示的密钥策略声明。

```
{
  "Sid": "Manage KMS keys with imported key material",
  "Effect": "Allow",
  "Resource": "*",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:role/KMSAdminRole"
  },
  "Action": [
    "kms:GetParametersForImport",
    "kms:ImportKeyMaterial",
    "kms:DeleteImportedKeyMaterial"
  ]
}
```

导入密钥材料的要求

您导入的密钥材料必须与相关 KMS 密钥的密钥规范兼容。对于非对称密钥对，仅导入密钥对的私钥。Amazon KMS 从私钥派生公钥。

Amazon KMS 支持使用导入的密钥材料的 KMS 密钥的以下密钥规范。

- 对称加密密钥
  - 密钥规范：

    - SYMMETRIC_DEFAULT。
  - 要求：
    - 256 位（32 字节）的二进制数据。
    - 在中国区域，必须是 128 位（16 字节）的二进制数据。
- HMAC 密钥
  - 密钥规范：
    - HMAC_224
    - HMAC_256
    - HMAC_384
    - HMAC_512
  - 要求：
    - HMAC 密钥材料必须符合 RFC 2104。
    - 密钥长度必须至少为密钥规范指定的长度。最大密钥长度为 1024 位。
    - 如果密钥材料的长度超过 1024 位，则可以对密钥材料进行哈希处理，然后导入哈希输出。哈希算法必须与您要创建的 HMAC KMS 密钥的密钥规范相匹配。
  - 示例：
    - 要将 2048 位的密钥材料导入 HMAC_256 密钥，请先计算 2048 位密钥材料的 SHA-256 哈希值，然后将生成的 256 位哈希输出导入 KMS 密钥。
  - 有效的密钥长度：
    - HMAC_224：224–1024 位
    - HMAC_256：256–1024 位
    - HMAC_384：384–1024 位
    - HMAC_512：512–1024 位
- RSA 非对称私有密钥
  - 密钥规范：
    - RSA_2048
    - RSA_3072
    - RSA_4096
  - 要求：
    - 您导入的 RSA 非对称私有密钥必须是符合 RFC 3447 的密钥对的一部分。
    - 模数：2048 位、3072 位或 4096 位

    - 素数数量：2（不支持多素数 RSA 密钥）
    - 非对称密钥材料必须为 BER 编码或 DER 编码，采用符合 RFC 5208 的公有密钥密码术标准（PKCS）#8 格式。
- 椭圆曲线非对称私有密钥
    - 密钥规范：
        - ECC_NIST_P256 (secp256r1)
        - ECC_NIST_P384 (secp384r1)
        - ECC_NIST_P521 (secp521r1)
        - ECC_SECG_P256K1 (secp256k1)
        - ECC_N EDWARDS25519 IST_ (ed25519)
    - 要求：
        - 您导入的 ECC 非对称私有密钥必须是符合 RFC 5915 的密钥对的一部分。
        - 曲线：NIST P-256、NIST P-384、NIST P-521、secp256k1、NIST Ed25519。
        - 参数：仅限命名曲线（拒绝带有显式参数的 ECC 密钥）。
        - 公共点坐标：可以是压缩坐标、未压缩坐标或投影坐标。
        - 非对称密钥材料必须为 BER 编码或 DER 编码，采用符合 RFC 5208 的公有密钥密码术标准（PKCS）#8 格式。
- ML-DSA 密钥
    - 密钥规范：
        - ML_DSA_44
        - ML_DSA_65
        - ML_DSA_87

> **Important**
>
> 不支持导入 ML-DSA 密钥。

- SM2 非对称私钥（仅限中国区域）
    - 要求：
        - 您导入的 SM2 非对称私钥必须是符合 0003 的密钥对的一部分。 GM/T
        - 曲线： SM2。
        - 参数：仅限命名曲线（拒绝带有显式参数的SM2 关键帧）。

- 公共点坐标：可以是压缩坐标、未压缩坐标或投影坐标。
- 非对称密钥材料必须为 BER 编码或 DER 编码，采用符合 RFC 5208 的公有密钥密码术标准（PKCS）#8 格式。

步骤 1：创建不带密钥材料的 Amazon KMS key

默认情况下，在 Amazon KMS 创建 KMS 密钥时会为您创建密钥材料。要改为导入自己的密钥材料，请先创建不带密钥材料的 KMS 密钥。然后导入密钥材料。要创建不带密钥材料的 KMS 密钥，请使用 Amazon KMS 控制台或CreateKey操作。

要创建不具有密钥材料的密钥，请指定 EXTERNAL 的源。KMS 密钥的源属性是不可变的。创建密钥后，您就无法将专为导入的密钥材料设计的 KMS 密钥转换为包含来自 Amazon KMS 或任何其他来源的密钥材料的 KMS 密钥。

带 EXTERNAL 且无密钥材料的 KMS 密钥的密钥状态为 PendingImport。KMS 密钥可以无限保留在 PendingImport 状态。但是，您不能在加密操作中使用处于 PendingImport 状态的 KMS 密钥。导入密钥材料时，KMS 密钥的密钥状态会更改为 Enabled，您可以在加密操作中使用该密钥。

Amazon KMS 在创建 KMS 密钥、下载公钥和导入令牌以及导入密钥材料时，会在 Amazon CloudTrail 日志中记录事件。 Amazon KMS 还会在您删除导入的密钥材料或 Amazon KMS 删除过期的密钥材料时记录 CloudTrail 事件。

主题

- 创建不带密钥材料的 KMS 密钥（控制台）
- 创建不含密钥材料的 KMS 密钥 (Amazon KMS API)

创建不带密钥材料的 KMS 密钥（控制台）

您只需为导入的密钥材料创建一次 KMS 密钥。您可以根据需要多次将相同的密钥材料导入和重新导入到现有的 KMS 密钥中，但不能将不同的密钥材料导入一个 KMS 密钥。有关更多信息，请参阅 步骤 2：下载包装公有密钥和导入令牌。

要在您的 客户管理型密钥 表中查找带有导入的密钥材料的现有 KMS 密钥，请使用右上角的齿轮图标显示 KMS 密钥列表中的 Origin（源）列。导入的密钥的源值为外部（导入密钥材料）。

要使用导入的密钥材料创建 KMS 密钥，请首先按照创建首选密钥类型的 KMS 密钥的说明操作，但以下情况除外。

选择密钥用法后，请执行以下操作：

1. 展开 Advanced options (高级选项)。
2. 对于 Key material origin（密钥材料源），请选择 External (Import key material) [外部（导入密钥材料）]。
3. 选择我了解使用导入密钥的安全性和持久性影响旁边的复选框，表示您了解使用导入密钥材料的影响。要了解这些含义，请参阅保护导入的密钥材料。
4. 可选：要使用导入的密钥材料创建多区域 KMS 密钥，请在区域性下选择多区域密钥。
5. 返回基本说明。对于该类型的所有 KMS 密钥，基本过程的其余步骤都相同。

选择完成时，您创建了一个 KMS 密钥，该密钥没有密钥材料，状态（密钥状态）为待导入。

但是，控制台不会返回到客户托管密钥表，而是显示一个页面，您可以在其中下载导入密钥材料所需的公有密钥和导入令牌。现在，您可以立即继续下载步骤，也可以选择取消停止下载。您可以随时返回此下载步骤。

下一步: 步骤 2：下载包装公有密钥和导入令牌。

创建不含密钥材料的 KMS 密钥 (Amazon KMS API)

要使用 Amazon KMS API 创建不含密钥材料的对称加密 KMS 密钥，请发送`Origin`参数设置为`EXTERNAL`的CreateKey请求。以下示例说明如何使用 Amazon Command Line Interface (Amazon CLI) 执行该操作。

```
$ aws kms create-key --origin EXTERNAL
```

该命令成功执行后，您会看到类似以下内容的输出。 Amazon KMS 关键`Origin`是`EXTERNAL`，现在`KeyState`是`PendingImport`。

> **Tip**
>
> 如果命令不成功，则可能会看到 `KMSInvalidStateException` 或 `NotFoundException`。您可以重试请求。

```
{
    "KeyMetadata": {
        "Origin": "EXTERNAL",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
```

```
        "Description": "",
        "Enabled": false,
        "MultiRegion": false,
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "PendingImport",
        "CreationDate": 1568289600.0,
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

从命令输出中复制 `KeyId` 值，以供后面的步骤使用，然后继续 步骤 2：下载包装公有密钥和导入令牌。

> **Note**
>
> 此命令创建对称加密 KMS 密钥，其 `KeySpec` 为 `SYMMETRIC_DEFAULT`，`KeyUsage` 为 `ENCRYPT_DECRYPT`。您可以使用可选参数 `--key-spec` 和 `--key-usage` 来创建非对称或 HMAC KMS 密钥。有关更多信息，请参阅 CreateKey 操作。

步骤 2：下载包装公有密钥和导入令牌

创建不 Amazon KMS key 带密钥材料的后，使用 Amazon KMS 控制台或 GetParametersForImportAPI 下载包装公钥和该 KMS 密钥的导入令牌。包装公有密钥和导入令牌是一个不可分割的集合，必须一起使用。

您将使用包装公有密钥来加密您的密钥材料以供传输。在下载 RSA 封装密钥对之前，请选择 RSA 封装密钥对的长度（密钥规范），以及将用于加密导入的密钥材料以便在步骤 3 中传输的封装算法。Amazon KMS 还支持 SM2 包装密钥规范（仅限中国地区）。

每个包装公有密钥和导入令牌集的有效期为 24 小时。如果您不在 24 小时的下载期限内使用它们导入密钥材料，则必须下载新的公有密钥和令牌集。您可以随时下载新的包装公有密钥和导入令牌集。这使您可以更改 RSA 包装密钥长度（“密钥规范”）或替换丢失的集。

您也可以下载包装公有密钥和导入令牌，以将相同的密钥材料重新导入 KMS 密钥中。您可以执行此操作来设置或更改密钥材料的过期时间，或者恢复过期或删除的密钥材料。每次将密钥材料导入时，都必须下载并重新加密密钥材料。 Amazon KMS

包装公有密钥的使用

下载内容包括您独有的公钥 Amazon Web Services 账户，也称为封装公钥。

在导入密钥材料之前，请使用公共封装密钥对密钥材料进行加密，然后将加密的密钥材料上传到 Amazon KMS。 Amazon KMS 收到您的加密密钥材料后，它会使用相应的私钥对密钥材料进行解密，然后在 AES 对称密钥下重新加密密钥材料，所有这些都在 Amazon KMS 硬件安全模块 (HSM) 中完成。

使用导入令牌

下载包括一个带有元数据的导入令牌，以确保您的密钥材料导入正确。将加密的密钥材料上传到时 Amazon KMS，必须上传在此步骤中下载的相同导入令牌。

选择包装公有密钥规范

为了在导入过程中保护您的密钥材料，您可以使用从中 Amazon KMS下载的封装公钥和支持的封装算法对其进行加密。您在下载包装公有密钥和导入令牌之前选择密钥规范。所有封装密钥对都是在 Amazon KMS 硬件安全模块 (HSMs) 中生成的。私有密钥永远不会让 HSM 处于纯文本状态。

RSA 包装密钥规范

包装公有密钥的密钥规范决定了 RSA 密钥对中密钥的长度，该密钥对在传输到 Amazon KMS的过程中可以保护您的密钥材料。一般来说，我们建议使用实用的最长的包装公有密钥。我们提供了几种封装公钥规范，以支持各种 HSMs 密钥管理器。

Amazon KMS 支持以下用于导入所有类型密钥材料的 RSA 封装密钥的关键规范，除非另有说明。

- RSA_4096（首选）
- RSA_3072
- RSA_2048

> **Note**
>
> 不支持以下组合：ECC_NIST_P521 密钥材料、RSA_2048 公有包装密钥规范和 RSAES_OAEP_SHA_* 包装算法。

> 您不能使用 RSA_2048 公有包装密钥直接包装 ECC_NIST_P521 密钥材料。使用更大的包装密钥或 RSA_AES_KEY_WRAP_SHA_* 包装算法。

SM2 包装密钥规范（仅限中国地区）

Amazon KMS 支持以下用于导入非对称密钥材料的 SM2 包装密钥的密钥规范。

- SM2

选择包装算法

要在导入过程中保护您的密钥材料，请使用下载的包装公有密钥和支持的包装算法为其加密。

Amazon KMS 支持多种标准 RSA 封装算法和两步混合包装算法。通常，我们建议使用与您导入的密钥材料和包装密钥规范兼容的最安全的包装算法。通常，选择硬件安全模块 (HSM) 支持的算法或用于保护密钥材料的密钥管理系统。

下表显示了每种类型的密钥材料和 KMS 密钥支持的包装算法。算法是以首选项顺序列出的。

密钥材料	支持的包装算法和规范
对称加密密钥 256 位 AES 密钥 128 位 SM4 密钥（仅限中国地区）	包装算法： RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 已淘汰的包装算法： RSAES_ _V1 PKCS1 **Note** 截至 2023 年 10 月 10 日，Amazon KMS 不支持 RSAES_ PKCS1 _V1_5 包装算法。 包装密钥规范： RSA_2048

密钥材料	支持的包装算法和规范
	RSA_3072 RSA_4096
非对称 RSA 私有密钥	包装算法： RSA_AES_KEY_WRAP_SHA_256 RSA_AES_KEY_WRAP_SHA_1 SM2PKE（仅限中国区域） 包装密钥规范： RSA_2048 RSA_3072 RSA_4096 SM2 （仅限中国地区）

密钥材料	支持的包装算法和规范
非对称椭圆曲线（ECC）私有密钥 您不能使用 RSAES_OAEP_SHA_* 包装算法和 RSA_2048 包装密钥规范来包装 ECC_NIST_P521 密钥材料。	包装算法： RSA_AES_KEY_WRAP_SHA_256 RSA_AES_KEY_WRAP_SHA_1 RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 SM2PKE（仅限中国区域） 包装密钥规范： RSA_2048 RSA_3072 RSA_4096 SM2 （仅限中国地区）
非对称 SM2 私钥（仅限中国区域）	包装算法： RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 SM2PKE（仅限中国区域） 包装密钥规范： RSA_2048 RSA_3072 RSA_4096 SM2 （仅限中国地区）

密钥材料	支持的包装算法和规范
HMAC 密钥	包装算法： RSAES_OAEP_SHA_256 RSAES_OAEP_SHA_1 包装密钥规范： RSA_2048 RSA_3072 RSA_4096

> **Note**
>
> 中国区域不支持 RSA_AES_KEY_WRAP_SHA_256 和 RSA_AES_KEY_WRAP_SHA_1 包装算法。

- RSA_AES_KEY_WRAP_SHA_256 – 一种两步混合包装算法，该算法将加密密钥材料与您生成的 AES 对称密钥相结合，然后使用下载的 RSA 公有包装密钥和 RSAES_OAEP_SHA_256 包装算法对 AES 对称密钥进行加密。

  除中国区域外，包装 RSA 私有密钥材料需要使用 RSA_AES_KEY_WRAP_SHA_* 包装算法。中国区域必须使用 SM2PKE 包装算法。
- RSA_AES_KEY_WRAP_SHA_1 – 一种两步混合包装算法，该算法将加密密钥材料与您生成的 AES 对称密钥相结合，然后使用下载的 RSA 包装公有密钥和 RSAES_OAEP_SHA_1 包装算法对 AES 对称密钥进行加密。

  除中国区域外，包装 RSA 私有密钥材料需要使用 RSA_AES_KEY_WRAP_SHA_* 包装算法。中国区域必须使用 SM2PKE 包装算法。
- RSAES_OAEP_SHA_256 – RSA 加密算法，使用最优非对称加密填充 (OAEP) 与 SHA-256 哈希函数。
- RSAES_OAEP_SHA_1 – RSA 加密算法，使用最优非对称加密填充 (OAEP) 与 SHA-1 哈希函数。

- RSAES_PKCS1_V1_5（已弃用；自 2023 年 10 月 10 日起，Amazon KMS 不支持 RSAES_PKCS1 _V1_5 包装算法）— 填充格式在 PKCS #1 版本 1.5 中定义的 RSA 加密算法。
- SM2PKE（仅限中国地区）— OSCCA GM/T 在 0003.4-2012 中定义的基于椭圆曲线的加密算法。

主题

下载包装公有密钥和导入令牌（控制台）

您可以使用 Amazon KMS 控制台下载封装公钥和导入令牌。

1. 如果您刚刚完成创建不带密钥材料的 KMS 密钥的步骤并且您位于 Download wrapping key and import token（下载包装密钥和导入令牌）页面上，请跳至 Step 9。
2. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/km s 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
3. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
4. 在导航窗格中，选择客户托管密钥。

   > **Tip**
   >
   > 您只能将密钥材料导入源为外部（导入密钥材料）的 KMS 密钥中。这指示已创建不带密钥材料的 KMS 密钥。要向表中添加 Origin (源) 列，请在页面右上角，选择设置图标（ ）。打开 Origin (源)，然后选择 Confirm (确认)。

5. 选择待导入的 KMS 密钥的别名或密钥 ID。
6. 选择 Cryptographic configuration（加密配置）选项卡并查看其值。这些选项卡在 General configuration（常规配置）部分下。

   您只能将密钥材料导入源为外部（导入密钥材料）的 KMS 密钥。有关创建带已导入密钥材料的 KMS 密钥的信息，请参阅 导入密钥的 Amazon KMS 密钥材料。
7. 选择密钥材料选项卡，然后选择导入密钥材料。

   密钥材料选项卡仅针对源值为外部（导入密钥材料）的 KMS 密钥显示。

8. 对于选择包装密钥规范，选择您的 KMS 密钥的配置。创建此密钥后，您无法更改密钥规范。
9. 对于选择包装算法，请选择您将用于为密钥材料加密的选项。有关这些选项的详细信息，请参阅选择包装算法。
10. 选择下载包装公有密钥和导入令牌，然后保存文件。

    如果有 Next (下一步) 选项，而且要立即继续执行此过程，请选择 Next (下一步)。要稍后再继续，请选择 Cancel (取消)。
11. 解压缩 `.zip` 文件，即您在上一步 (`Import_Parameters_<key_id>_<timestamp>`) 中保存的文件。

    此文件夹包含以下文件：

    - 名为 `WrappingPublicKey.bin` 的文件中的包装公有密钥。
    - 名为 `ImportToken.bin` 的文件中的导入令牌。
    - 名为 README.txt 的文本文件。此文件包含以下相关信息：包装公有密钥、用于为密钥材料加密的包装算法，以及包装公有密钥和导入令牌的过期日期和时间。
12. 要继续执行此过程，请参阅为您的密钥材料加密。

下载封装公钥和导入令牌 (Amazon KMS API)

要下载公钥和导入令牌，请使用 GetParametersForImportAPI。指定将与导入的密钥材料关联的 KMS 密钥。此 KMS 密钥的 Origin 值必须为 `EXTERNAL`。

> **Note**
>
> 不能导入 ML-DSA KMS 密钥的密钥材料。

此示例指定了 `RSA_AES_KEY_WRAP_SHA_256` 包装算法、`RSA_3072` 包装公有密钥规范和示例密钥 ID。将这些示例值替换为有效的下载值。在此操作中，对于密钥 ID，您可以使用密钥 ID 或密钥 ARN，但不能使用别名名称或别名 ARN。

```
$ aws kms get-parameters-for-import \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --wrapping-algorithm RSA_AES_KEY_WRAP_SHA_256 \
    --wrapping-key-spec RSA_3072
```

该命令成功执行后，您会看到类似以下内容的输出：

```
{
    "ParametersValidTo": 1568290320.0,
    "PublicKey": "public key (base64 encoded)",
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "ImportToken": "import token (base64 encoded)"
}
```

为了准备下一步的数据，base64 会对公有密钥和导入令牌进行解码，并将解码后的值保存在文件中。

要对公有密钥进行 base64 解码并导入令牌：

1. 复制 base64 编码的公钥（*public key (base64 encoded)*在示例输出中用表示），将其粘贴到新文件中，然后保存该文件。向文件提供一个描述性名称，例如 PublicKey.b64。
2. 使用 OpenSSL 对文件的内容进行 base64 解码，然后将解码后的数据保存到一个新文件中。以下示例会对您在上一步骤 (PublicKey.b64) 中保存的文件中的数据进行解码，并将输出保存到一个名为 WrappingPublicKey.bin 的新文件中。

   ```
   $ openssl enc -d -base64 -A -in PublicKey.b64 -out WrappingPublicKey.bin
   ```

3. 复制 base64 编码的导入令牌（*import token (base64 encoded)*在示例输出中用表示），将其粘贴到新文件中，然后保存该文件。为文件指定一个描述性名称，例如 importtoken.b64。
4. 使用 OpenSSL 对文件的内容进行 base64 解码，然后将解码后的数据保存到一个新文件中。以下示例会对您在上一步骤 (ImportToken.b64) 中保存的文件中的数据进行解码，并将输出保存到一个名为 ImportToken.bin 的新文件中。

   ```
   $ openssl enc -d -base64 -A -in importtoken.b64 -out ImportToken.bin
   ```

继续执行步骤 3：加密密钥材料。

步骤 3：加密密钥材料

下载公有密钥和导入令牌后，使用您下载的公有密钥和指定的包装算法对密钥材料进行加密。如果您需要替换公有密钥或导入令牌，或者更改包装算法，则必须下载新的公有密钥和导入令牌。有关 Amazon KMS 支持的公钥和封装算法的信息，请参阅选择包装公有密钥规范和选择包装算法。

密钥材料必须采用二进制格式。有关详细信息，请参阅 导入密钥材料的要求。

> **Note**
>
> 对于非对称密钥对，仅加密和导入私钥。 Amazon KMS 从私钥派生公钥。
> 不支持以下组合：ECC_NIST_P521 密钥材料、RSA_2048 公有包装密钥规范和 RSAES_OAEP_SHA_* 包装算法。
> 您不能使用 RSA_2048 公有包装密钥直接包装 ECC_NIST_P521 密钥材料。使用更大的包装密钥或 RSA_AES_KEY_WRAP_SHA_* 包装算法。
> 中国区域不支持 RSA_AES_KEY_WRAP_SHA_256 和 RSA_AES_KEY_WRAP_SHA_1 包装算法。

通常，您可以在将密钥材料从硬件安全模块 (HSM) 或密钥管理系统导出时对其进行加密。有关如何以二进制格式导出密钥材料的信息，请参阅有关 HSM 或密钥管理系统的文档。您还可以参阅以下部分，该部分使用 OpenSSL 提供了概念验证演示。

加密密钥材料时，请使用与您在下载公有密钥和导入令牌时指定的相同的包装算法。要查找您指定的包装算法，请查看关联GetParametersForImport请求的 CloudTrail 日志事件。

生成用于测试的密钥材料

以下 OpenSSL 命令生成每种支持类型的密钥材料以供测试。这些示例仅用于测试和 proof-of-concept 演示。对于生产系统，请使用更安全的方法来生成您的密钥材料，例如硬件安全模块或密钥管理系统。

要将非对称密钥对的私有密钥转换为 DER 编码格式，请将密钥材料生成命令传送到以下 `openssl pkcs8` 命令。`topk8` 参数指示 OpenSSL 将私有密钥作为输入并返回 PKCS#8 格式的密钥。（默认行为恰恰相反。）

```
openssl pkcs8 -topk8 -outform der -nocrypt
```

以下命令为每种支持的密钥类型生成测试密钥材料。

- 对称加密密钥（32 字节）

  此命令生成 256 位的对称密钥（32 字节的随机字符串）并将其保存在 `PlaintextKeyMaterial.bin` 文件中。您无需对这些密钥材料进行编码。

  ```
  openssl rand -out PlaintextKeyMaterial.bin 32
  ```

  仅在中国区域，您必须生成 128 位的对称密钥（16 字节的随机字符串）。

```
openssl rand -out PlaintextKeyMaterial.bin 16
```

- HMAC 密钥

  此命令生成指定大小的随机字节字符串。您无需对这些密钥材料进行编码。

  您的 HMAC 密钥的长度必须与 KMS 密钥的密钥规范定义的长度相匹配。例如，如果 KMS 密钥为 HMAC_384，则必须导入 384 位（48 字节）的密钥。

```
openssl rand -out HMAC_224_PlaintextKey.bin 28

openssl rand -out HMAC_256_PlaintextKey.bin 32

openssl rand -out HMAC_384_PlaintextKey.bin 48

openssl rand -out HMAC_512_PlaintextKey.bin 64
```

- RSA 私有密钥

```
openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:2048 | openssl pkcs8 -topk8 -
outform der -nocrypt > RSA_2048_PrivateKey.der

openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:3072 | openssl pkcs8 -topk8 -
outform der -nocrypt > RSA_3072_PrivateKey.der

openssl genpkey -algorithm rsa -pkeyopt rsa_keygen_bits:4096 | openssl pkcs8 -topk8 -
outform der -nocrypt > RSA_4096_PrivateKey.der
```

- ECC 私有密钥

```
openssl genpkey -algorithm ec -pkeyopt ec_paramgen_curve:P-256 | openssl pkcs8 -topk8
 -outform der -nocrypt > ECC_NIST_P256_PrivateKey.der

openssl genpkey -algorithm ec -pkeyopt ec_paramgen_curve:P-384 | openssl pkcs8 -topk8
 -outform der -nocrypt > ECC_NIST_P384_PrivateKey.der

openssl genpkey -algorithm ec -pkeyopt ec_paramgen_curve:P-521 | openssl pkcs8 -topk8
 -outform der -nocrypt > ECC_NIST_P521_PrivateKey.der

openssl genpkey -algorithm ec -pkeyopt ec_paramgen_curve:secp256k1 | openssl pkcs8 -
topk8 -outform der -nocrypt > ECC_SECG_P256K1_PrivateKey.der
```

- SM2 私钥（仅限中国地区）

```
openssl genpkey -algorithm ec -pkeyopt ec_paramgen_curve:sm2 | openssl pkcs8 -topk8 -
outform der -nocrypt > SM2_PrivateKey.der
```

使用 OpenSSL 加密密钥材料的示例

以下示例说明如何使用 OpenSSL 通过您下载的公有密钥对密钥材料进行加密。要使用 SM2 公钥加密密钥材料（仅限中国地区），请使用SM2OfflineOperationHelper该类。有关每种包装算法支持的密钥材料类型的更多信息，请参阅 the section called “选择包装算法”。

> **Important**
>
> 这些示例仅为概念验证演示。对于生产系统，请使用更安全的方法 (如商业 HSM 或密钥管理系统) 来生成和存储您的密钥材料。
> 不支持以下组合：ECC_NIST_P521 密钥材料、RSA_2048 公有包装密钥规范和 RSAES_OAEP_SHA_* 包装算法。
> 您不能使用 RSA_2048 公有包装密钥直接包装 ECC_NIST_P521 密钥材料。使用更大的包装密钥或 RSA_AES_KEY_WRAP_SHA_* 包装算法。

RSAES_OAEP_SHA_1

Amazon KMS 支持对称加密密钥 (SYMMETRIC_DEFAULT)、椭圆曲线 (ECC) 私钥、私钥和 HMAC 密钥的 RSAES_OAEP_SHA_1。 SM2

RSAES_OAEP_SHA_1 不支持 RSA 私有密钥。此外，您不能使用采用任何 RSAES_OAEP_SHA_* 包装算法的 RSA_2048 公有包装密钥来包装 ECC_NIST_P521（secp521r1）私有密钥。您必须使用更大的公有包装密钥或 RSA_AES_KEY_WRAP 包装算法。

以下示例使用您下载的公有密钥和 RSAES_OAEP_SHA_1 包装算法对密钥材料进行加密，并将其保存在 `EncryptedKeyMaterial.bin` 文件中。

在本示例中：

- *`WrappingPublicKey.bin`* 是包含下载的包装公有密钥的文件。

- *PlaintextKeyMaterial.bin* 是包含您正在加密的密钥材料的文件，例如 PlaintextKeyMaterial.bin、HMAC_384_PlaintextKey.bin 或 ECC_NIST_P521_PrivateKey.der。

```
$ openssl pkeyutl \
    -encrypt \
    -in PlaintextKeyMaterial.bin \
    -out EncryptedKeyMaterial.bin \
    -inkey WrappingPublicKey.bin \
    -keyform DER \
    -pubin \
    -pkeyopt rsa_padding_mode:oaep \
    -pkeyopt rsa_oaep_md:sha1
```

RSAES_OAEP_SHA_256

Amazon KMS 支持对称加密密钥 (SYMMETRIC_DEFAULT)、椭圆曲线 (ECC) 私钥、私钥和 HMAC 密钥的 RSAES_OAEP_SHA_256。 SM2

RSAES_OAEP_SHA_256 不支持 RSA 私有密钥。此外，您不能使用采用任何 RSAES_OAEP_SHA_* 包装算法的 RSA_2048 公有包装密钥来包装 ECC_NIST_P521（secp521r1）私有密钥。您必须使用更大的公有密钥或 RSA_AES_KEY_WRAP 包装算法。

以下示例使用您下载的公有密钥和 RSAES_OAEP_SHA_256 包装算法对密钥材料进行加密，并将其保存在 EncryptedKeyMaterial.bin 文件中。

在本示例中：

- *WrappingPublicKey.bin* 是包含已下载的公有包装密钥的文件。如果您是从控制台下载的公有密钥，则此文件的名称为 wrappingKey_*KMS key_key_ID_timestamp*（例如，wrappingKey_f44c4e20-f83c-48f4-adc6-a1ef38829760_0809092909）。
- *PlaintextKeyMaterial.bin* 是包含您正在加密的密钥材料的文件，例如 PlaintextKeyMaterial.bin、HMAC_384_PlaintextKey.bin 或 ECC_NIST_P521_PrivateKey.der。

```
$ openssl pkeyutl \
    -encrypt \
    -in PlaintextKeyMaterial.bin \
```

```
    -out EncryptedKeyMaterial.bin \
    -inkey WrappingPublicKey.bin \
    -keyform DER \
    -pubin \
    -pkeyopt rsa_padding_mode:oaep \
    -pkeyopt rsa_oaep_md:sha256 \
    -pkeyopt rsa_mgf1_md:sha256
```

RSA_AES_KEY_WRAP_SHA_1

RSA_AES_KEY_WRAP_SHA_1 包装算法涉及两个加密操作。

1. 使用您生成的 AES 对称密钥和 AES 对称加密算法对密钥材料进行加密。
2. 使用您下载的公有密钥和 RSAES_OAEP_SHA_1 包装算法加密您使用的 AES 对称密钥。

RSA_AES_KEY_WRAP_SHA_1 包装算法需要 OpenSSL 版本 3.x 或更高版本。

1. 生成 256 位 AES 对称加密密钥

   此命令生成由 256 个随机位组成的 AES 对称加密密钥，并将其保存在 `aes-key.bin` 文件中

   ```
   # Generate a 32-byte AES symmetric encryption key
   $ openssl rand -out aes-key.bin 32
   ```

2. 使用 AES 对称加密密钥加密您的密钥材料

   此命令使用 AES 对称加密密钥对您的密钥材料进行加密，并将加密的密钥材料保存在 `key-material-wrapped.bin` 文件中。

   在此示例命令中：

   - *PlaintextKeyMaterial.bin* 是包含您要导入的密钥材料的文件，例如 PlaintextKeyMaterial.bin、HMAC_384_PlaintextKey.bin、RSA_3072_PrivateKey.
     或 ECC_NIST_P521_PrivateKey.der。
   - *aes-key.bin* 是包含您在上一个命令中生成的 256 位 AES 对称加密密钥的文件。

   ```
   # Encrypt your key material with the AES symmetric encryption key
   $ openssl enc -id-aes256-wrap-pad \
           -K "$(xxd -p < aes-key.bin | tr -d '\n')" \
           -iv A65959A6 \
   ```

```
        -in PlaintextKeyMaterial.bin\
        -out key-material-wrapped.bin
```

3. 使用公有密钥加密您的 AES 对称加密密钥

   此命令使用您下载的公有密钥和 RSAES_OAEP_SHA_1 包装算法对您的 AES 对称加密密钥进行加密，对其进行 DER 编码，然后将其保存在 aes-key-wrapped.bin 文件中。

   在此示例命令中：

   - *WrappingPublicKey.bin* 是包含已下载的公有包装密钥的文件。如果您是从控制台下载的公有密钥，则此文件的名称为 wrappingKey_*KMS key_key_ID_timestamp*（例如，wrappingKey_f44c4e20-f83c-48f4-adc6-a1ef38829760_0809092909
   - *aes-key.bin* 是包含您在本示例序列的第一个命令中生成的 256 位 AES 对称加密密钥的文件。

```
# Encrypt your AES symmetric encryption key with the downloaded public key
$ openssl pkeyutl \
    -encrypt \
    -in aes-key.bin \
    -out aes-key-wrapped.bin \
    -inkey WrappingPublicKey.bin \
    -keyform DER \
    -pubin \
    -pkeyopt rsa_padding_mode:oaep \
    -pkeyopt rsa_oaep_md:sha1 \
    -pkeyopt rsa_mgf1_md:sha1
```

4. 生成要导入的文件

   将文件与加密的密钥材料连接起来，并将文件与加密的 AES 密钥连接起来。将它们保存在 EncryptedKeyMaterial.bin 文件中，也就是您要在 步骤 4：导入密钥材料 中导入的文件。

   在此示例命令中：

   - *key-material-wrapped.bin* 是包含您的已加密密钥材料的文件。
   - *aes-key-wrapped.bin* 是包含已加密 AES 加密密钥的文件。

```
# Combine the encrypted AES key and encrypted key material in a file
$ cat aes-key-wrapped.bin key-material-wrapped.bin > EncryptedKeyMaterial.bin
```

RSA_AES_KEY_WRAP_SHA_256

RSA_AES_KEY_WRAP_SHA_256 包装算法涉及两个加密操作。

1. 使用您生成的 AES 对称密钥和 AES 对称加密算法对密钥材料进行加密。
2. 使用您下载的公有密钥和 RSAES_OAEP_SHA_256 包装算法加密您使用的 AES 对称密钥。

RSA_AES_KEY_WRAP_SHA_256 包装算法需要 OpenSSL 版本 3.x 或更高版本。

1. 生成 256 位 AES 对称加密密钥

   此命令生成由 256 个随机位组成的 AES 对称加密密钥，并将其保存在 aes-key.bin 文件中

   ```
   # Generate a 32-byte AES symmetric encryption key
   $ openssl rand -out aes-key.bin 32
   ```

2. 使用 AES 对称加密密钥加密您的密钥材料

   此命令使用 AES 对称加密密钥对您的密钥材料进行加密，并将加密的密钥材料保存在 key-material-wrapped.bin 文件中。

   在此示例命令中：

   - *PlaintextKeyMaterial.bin* 是包含您要导入的密钥材料的文件，例如 PlaintextKeyMaterial.bin、HMAC_384_PlaintextKey.bin、RSA_3072_PrivateKey.
     或 ECC_NIST_P521_PrivateKey.der。
   - *aes-key.bin* 是包含您在上一个命令中生成的 256 位 AES 对称加密密钥的文件。

   ```
   # Encrypt your key material with the AES symmetric encryption key
   $ openssl enc -id-aes256-wrap-pad \
           -K "$(xxd -p < aes-key.bin | tr -d '\n')" \
           -iv A65959A6 \
           -in PlaintextKeyMaterial.bin\
   ```

```
        -out key-material-wrapped.bin
```

3. 使用公有密钥加密您的 AES 对称加密密钥

   此命令使用您下载的公有密钥和 RSAES_OAEP_SHA_256 包装算法对您的 AES 对称加密密钥进行加密，对其进行 DER 编码，然后将其保存在 aes-key-wrapped.bin 文件中。

   在此示例命令中：

   - *WrappingPublicKey.bin* 是包含已下载的公有包装密钥的文件。如果您是从控制台下载的公有密钥，则此文件的名称为 wrappingKey_*KMS key_key_ID_timestamp*（例如，wrappingKey_f44c4e20-f83c-48f4-adc6-a1ef38829760_0809092909
   - *aes-key.bin* 是包含您在本示例序列的第一个命令中生成的 256 位 AES 对称加密密钥的文件。

```
# Encrypt your AES symmetric encryption key with the downloaded public key
$ openssl pkeyutl \
    -encrypt \
    -in aes-key.bin \
    -out aes-key-wrapped.bin \
    -inkey WrappingPublicKey.bin \
    -keyform DER \
    -pubin \
    -pkeyopt rsa_padding_mode:oaep \
    -pkeyopt rsa_oaep_md:sha256 \
    -pkeyopt rsa_mgf1_md:sha256
```

4. 生成要导入的文件

   将文件与加密的密钥材料连接起来，并将文件与加密的 AES 密钥连接起来。将它们保存在 EncryptedKeyMaterial.bin 文件中，也就是您要在 步骤 4：导入密钥材料 中导入的文件。

   在此示例命令中：

   - *key-material-wrapped.bin* 是包含您的已加密密钥材料的文件。
   - *aes-key-wrapped.bin* 是包含已加密 AES 加密密钥的文件。

```
# Combine the encrypted AES key and encrypted key material in a file
```

```
$ cat aes-key-wrapped.bin key-material-wrapped.bin > EncryptedKeyMaterial.bin
```

继续执行步骤 4：导入密钥材料。

步骤 4：导入密钥材料

在加密密钥材料之后，您可以导入密钥材料，以将其与 Amazon KMS key配合使用。要导入密钥材料，请上传在步骤 3：加密密钥材料中加密的密钥材料以及在步骤 2：下载包装公有密钥和导入令牌中下载的导入令牌。您必须将密钥材料导入您在下载公有密钥和导入令牌时指定的同一 KMS 密钥中。成功导入密钥材料时，KMS 密钥的密钥状态会更改为 Enabled，您可以在加密操作中使用 KMS 密钥。

当您导入密钥材料时，您可以为密钥材料设置可选的过期日期。当密钥材料过期后， Amazon KMS 将删除密钥材料，并且 KMS 密钥将变为不可用。导入密钥材料后，无法设置、更改或取消当前导入的到期日期。要更改这些值，您必须重新导入相同的密钥材料。

按密钥类型划分的密钥材料导入注意事项

对称加密密钥

对于所有来源为 EXTERNAL 的 KMS 密钥，导入的第一个密钥材料将变为当前密钥材料并与之永久关联。带EXTERNAL来源的对称加密密钥支持按需轮换。您可以将多个密钥材料关联到支持按需轮换的导入的密钥。要将新密钥材料NEW_KEY_MATERIAL与 KMS 密钥关联的ImportKeyMaterial操作，必须将importType参数设置为。在您执行RotateKeyOnDemand操作之前，此密钥材料不会与密钥永久关联。在此之前，此密钥材料将处于 PENDING_ROTATION 状态。可选 ImportType 参数的默认值为 EXISTING_KEY_MATERIAL。省略 ImportType 参数或将其指定为 EXISTING_KEY_MATERIAL 时，必须导入先前与 KMS 密钥关联的密钥材料。

非对称密钥和 HMAC 密钥

对于非对称密钥或带有EXTERNAL来源的 HMAC KMS 密钥，只能将一种密钥材料与密钥相关联。Amazon KMS 将拒绝带ImportType参数 ImportKeyMaterial的 API 请求。

多区域密钥

您可以将新的密钥材料导入到多区域对称密钥中。为此，您必须创建主区域密钥和副本区域密钥。然后，您必须将新的密钥材料导入主区域密钥。您不能直接将新的密钥材料导入副本区域密钥。将新的密钥材料导入主区域密钥后，您可以将相同的密钥材料导入副本区域密钥中。

如果主区域密钥或副本区域密钥中的密钥材料被删除或过期，则只有该特定密钥会受到影响。已导入所有永久关联密钥材料的副本区域密钥可用于加密操作。

您只能使用多区域密钥设置或更改主区域密钥的密钥材料描述。

导入密钥材料时的密钥状态

导入与 KMS 密钥永久关联的所有密钥材料后，KMS 密钥才可用于密码操作。如果删除其中任何一个密钥材料或放任其过期，则 KMS 密钥状态将变为 `PendingImport`，并且该密钥将不再能够用于密码操作。

要导入密钥材料，您可以使用Amazon KMS 控制台或 ImportKeyMaterialAPI。您可以直接使用 API，方法是发出 HTTP 请求，也可以使用Amazon SDKs、Amazon Command Line Interface或Amazon Tools for PowerShell。

导入密钥材料时，会在 Amazon CloudTrail 日志中添加一个记录`ImportKeyMaterial`操作的ImportKeyMaterial条目。无论您使用 Amazon KMS 控制台还是 Amazon KMS API， CloudTrail 条目都是一样的。

设置过期时间（可选）

导入 KMS 密钥的密钥材料时，可以将密钥材料的可选过期日期和时间设置为自导入之日起 365 天中的任意一天。导入的密钥材料过期后，将其 Amazon KMS 删除。此操作会将 KMS 密钥的密钥状态更改为 `PendingImport`，这将阻止在任何加密操作中使用该密钥。要使用 KMS 密钥，您必须重新导入原始密钥材料的副本。

确保导入的密钥材料经常过期，可帮助您满足监管要求，但这样做会给在 KMS 密钥下加密的数据带来额外的风险。在您重新导入原始密钥材料的副本之前，包含过期密钥材料的 KMS 密钥不可用，并且在 KMS 密钥下加密的任何数据都不可访问。如果您出于任何原因未能重新导入密钥材料（例如，丢失原始密钥材料的副本），则 KMS 密钥将永久不可用，在 KMS 密钥下加密的数据将无法恢复。

为了降低这种风险，请确保导入的密钥材料的副本可以访问，并设计一个系统，以便在密钥材料过期并中断工作量之前将其删除并重新导入。 Amazon 我们建议您针对导入的密钥材料的过期设置警报，这样您就有足够的时间在密钥材料过期之前重新导入密钥材料。您还可以使用 CloudTrail 日志来审核导入（和重新导入）密钥材料和删除导入的密钥材料的 Amazon KMS 操作，以及删除过期密钥材料的操作。

Amazon KMS 无法恢复、恢复或重现已删除的密钥材料。您可以通过编程定期删除和重新导入已导入的密钥材料，而无需设置过期时间，但是保留原始密钥材料副本的要求相同。

在导入密钥材料时，确定导入的密钥材料是否以及何时过期。不过，您可以
开启和关闭过期时间，也可以通过重新导入密钥材料来设置新的过期时间。使

用ExpirationModel参数开启ImportKeyMaterial到期时间 (KEY_MATERIAL_EXPIRES) 和关闭 (KEY_MATERIAL_DOES_NOT_EXPIRE)，使用ValidTo参数设置到期时间。自导入数据起的最大天数为 365 天；没有最短天数，但必须是未来时间。

设置密钥材料描述

具有EXTERNAL来源的对称加密密钥可以有多个与之关联的密钥材料。在将密钥材料导入此类密钥时，您可以指定可选的密钥材料描述。该描述可用于跟踪相应的密钥材料在 Amazon KMS之外持久保存的位置。

您只能使用多区域密钥设置或更改主区域密钥的密钥材料描述。

重新导入密钥材料

如果您管理带有导入的密钥材料的 KMS 密钥，则可能需要重新导入密钥材料。您可以通过重新导入密钥材料替换过期或删除的密钥材料，或者更改密钥材料的到期模型或到期日期。

您可以在可满足您的安全要求任何时间点重新导入密钥材料。您不必等到密钥材料达到或接近其过期时间。

重新导入密钥材料的过程与首次导入密钥材料的过程相同，但以下情况例外。

- 使用现有 KMS 密钥，而不是创建新的 KMS 密钥。您可以跳过导入过程的步骤 1。
- 重新导入密钥材料时，可以更改到期模型、到期日期和密钥材料描述。

  您只能使用多区域密钥设置或更改主区域密钥的密钥材料描述。

每次将密钥材料导入 KMS 密钥时，您需要为 KMS 密钥下载并使用新的包装密钥和导入令牌。包装过程不会影响密钥材料的内容，因此，您可以使用不同的包装公有密钥和不同的包装算法来导入相同的密钥材料。

导入新密钥材料

要具有导入的密钥材料的对称加密 KMS 密钥执行按需轮换，您首先需要导入以前未与该密钥关联的新密钥材料。

- 单一区域密钥
  - 使用ImportType参数设置为的ImportKeyMaterial操作NEW_KEY_MATERIAL来完成此任务。以这种方式导入的密钥材料将处于PENDING_ROTATION状态，直到您执行RotateKeyOnDemand操作

或在中旋转密钥为止 Amazon Web Services 管理控制台。一个 KMS 密钥在任何时候最多只能有一个密钥材料处于 PENDING_ROTATION 状态。

- 多区域密钥
  - 要将密钥材料导入多区域密钥，必须先将新的密钥材料导入主区域密钥。您不能直接将新的密钥材料导入副本区域密钥。将新的密钥材料导入主区域密钥后，您可以将相同的密钥材料导入到副本区域密钥中。
  - 使用将主区域键的ImportType参数设置**NEW_KEY_MATERIAL**为的ImportKeyMaterial操作来完成此任务。对于副本区域密钥，请使用ImportKeyMaterial操作ImportType的**EXISTING_KEY_MATERIAL**参数。
  - 在密钥材料状态更改为状态之前，必须将多区域密钥的密钥材料导入到所有副本区域密钥和主区域密钥中。PENDING_ROTATION在此之前，新密钥材料的状态为PENDING_MULTI_REGION_IMPORT_AND_ROTATION。一个 KMS 密钥在任何时候最多可以有一个处于PENDING_ROTATION或PENDING_MULTI_REGION_IMPORT_AND_ROTATION状态的密钥材料。

导入密钥材料（控制台）

您可以使用 Amazon Web Services 管理控制台 来导入密钥材料。

1. 如果您在上传已包装的密钥材料页面上，请跳至 Step 8。
2. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/km s 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
3. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
4. 在导航窗格中，选择客户托管密钥。
5. 选择已为其下载公有密钥和导入令牌的 KMS 密钥的密钥 ID 或别名。
6. 选择 Cryptographic configuration（加密配置）选项卡并查看其值。这些选项卡位于 General configuration（常规配置）部分下 KMS 密钥的详细信息页面上。

   您只能将密钥材料导入源为外部（导入密钥材料）的 KMS 密钥。有关创建带已导入密钥材料的 KMS 密钥的信息，请参阅 导入密钥的 Amazon KMS 密钥材料。
7. 对于非对称密钥和 HMAC 密钥，请选择密钥材料选项卡，然后选择导入密钥材料。对于对称加密密钥，请选择密钥材料和轮换选项卡。然后选择导入初始密钥材料或导入新密钥材料或重新导入密钥材料。重新导入密钥材料选项可在密钥材料表的 Actions 菜单中找到。

   如果您下载了密钥材料、导入令牌并加密了密钥材料，请选择下一步。

> **Note**
>
> 对于多区域密钥，您必须先将新的密钥材料导入主区域密钥中。然后，您可以将相同的密钥材料导入到副本区域密钥中。

8. 在加密的密钥材料和导入令牌部分，执行以下操作。

   a. 在包装的密钥材料下，选择选择文件。然后上传包含您的已包装（已加密）密钥材料的文件。

   b. 在导入令牌下，选择选择文件。上传包含您已下载的导入令牌的文件。

9. 在 Expiration option (过期选项) 部分中，确定密钥材料是否过期。要设置到期日期和时间，请选择 Key material expires (密钥材料过期)，并使用日历选择日期和时间。您可以指定的日期距当前日期和时间最多 365 天。
10. 对于对称加密密钥，您可以选择为要导入的密钥材料指定描述。
11. 选择导入密钥材料。

导入密钥材料 (Amazon KMS API)

要导入密钥材料，请使用ImportKeyMaterial操作。以下示例使用 Amazon CLI，但您可以使用受支持的任何编程语言。

要使用此示例，请执行以下操作：

您只能使用多区域密钥设置或更改主区域密钥的密钥材料描述。

1. 将 *1234abcd-12ab-34cd-56ef-1234567890ab* 替换为您在下载公有密钥和导入令牌时指定的 KMS 密钥的密钥 ID。要标识 KMS 密钥，请使用其密钥 ID 或密钥 ARN。该操作不能使用别名或别名 ARN。
2. 将 *EncryptedKeyMaterial.bin* 替换为包含加密的密钥材料的文件的名称。
3. 将 *ImportToken.bin* 替换为包含导入令牌的文件的名称。
4. 如果希望导入的密钥材料过期，请将 `expiration-model` 参数的值设置为其默认值 `KEY_MATERIAL_EXPIRES`，或省略 `expiration-model` 参数。然后，将 `valid-to` 参数的值替换为您希望密钥材料过期的日期和时间。日期和时间最长为请求时间起 365 天。

```
$ aws kms import-key-material --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --encrypted-key-material fileb://EncryptedKeyMaterial.bin \
    --import-token fileb://ImportToken.bin \
    --expiration-model KEY_MATERIAL_EXPIRES \
```

```
    --valid-to 2023-06-17T12:00:00-08:00
```

如果不希望导入的密钥材料过期，请将 expiration-model 参数的值设置为 KEY_MATERIAL_DOES_NOT_EXPIRE，并从命令中省略 valid-to 参数。

```
$ aws kms import-key-material --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --encrypted-key-material fileb://EncryptedKeyMaterial.bin \
    --import-token fileb://ImportToken.bin \
    --expiration-model KEY_MATERIAL_DOES_NOT_EXPIRE
```

5. 如果要导入以前未与 KMS 密钥关联的新密钥材料，请将 ImportType 参数设置为 NEW_KEY_MATERIAL。此选项只能与对称加密密钥一起使用。对于此类密钥，您还可以在以下命令行示例中使用可选 KeyMaterialDescription 参数为导入的密钥材料设置描述：

```
$ aws kms import-key-material --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --encrypted-key-material fileb://EncryptedKeyMaterial.bin \
    --import-token fileb://ImportToken.bin \
    --expiration-model KEY_MATERIAL_EXPIRES \
    --valid-to 2023-06-17T12:00:00-08:00 \
    --import-type NEW_KEY_MATERIAL \
    --key-material-description "Q2 2025 Rotation"
```

> **Tip**
>
> 如果命令不成功，则可能会看到 KMSInvalidStateException 或 NotFoundException。您可以重试请求。

在密钥库中创建 KMS Amazon CloudHSM 密钥

创建 Amazon CloudHSM 密钥库后，可以在密钥库 Amazon KMS keys 中创建。它们必须是对称加密 KMS 密钥，其中包含 Amazon KMS 生成的密钥材料。您不能在自定义密钥存储中创建非对称 KMS 密钥、HMAC KMS 密钥或具有导入的密钥材料的 KMS 密钥。此外，您不能在自定义密钥存储中使用对称加密 KMS 密钥来生成非对称数据密钥对。KMS 无法 IPv6 与 Amazon CloudHSM 密钥存储库进行通信。

要在密钥存储中创建 KMS 密 Amazon CloudHSM 钥， Amazon CloudHSM 密钥库必须连接到关联的 Amazon CloudHSM 集群，并且集群必须至少包含两个 HSMs 处于不同可用区域的活动密钥。要

查找的连接状态和数量 HSMs，请查看中的Amazon CloudHSM 密钥存储页面 Amazon Web Services 管理控制台。使用 API 操作时，使用DescribeCustomKeyStores操作验证 Amazon CloudHSM 密钥库是否已连接。要验证集群 HSMs 中的活动人数及其可用区，请使用 Amazon CloudHSM DescribeClusters操作。

在密钥存储中创建 KMS Amazon CloudHSM 密钥时， Amazon KMS 会在中创建 KMS 密钥 Amazon KMS。但是，它会为关联 Amazon CloudHSM 集群中的 KMS 密钥创建密钥材料。具体而言，Amazon KMS 以您创建的 kmsuser CU 身份登录集群。然后，它在集群中创建持久的、不可提取的 256 位高级加密标准 (AES) 对称密钥。 Amazon KMS 将密钥标签属性的值（仅在集群中可见）设置为 KMS 密钥的 Amazon Resource Name (ARN)。

当命令成功时，新 KMS 密钥的密钥状态为 Enabled，其源为 AWS_CLOUDHSM。创建任何 KMS 密钥后便无法更改其源。当您在 Amazon KMS 控制台的密 Amazon CloudHSM 钥存储库中或使用DescribeKey操作查看 KMS 密钥时，可以看到典型的属性，例如其密钥 ID、密钥状态和创建日期。但是，您也可以查看自定义密钥存储 ID 和（可选） Amazon CloudHSM 集群 ID。

如果您尝试在密钥库中创建 KMS Amazon CloudHSM 密钥失败，请使用错误消息来帮助您确定原因。这可能表示 Amazon CloudHSM 密钥库未连接 (CustomKeyStoreInvalidStateException)，或者关联的 Amazon CloudHSM 集群没有此操作所需的两个密钥库处于活动状态 HSMs (CloudHsmClusterInvalidConfigurationException)。有关帮助信息，请参阅对自定义密钥存储进行故障排除。

有关在密钥存储中创建 KMS 密钥的操作 Amazon CloudTrail 日志的 Amazon CloudHSM 示例，请参阅CreateKey。

在您的 CloudHSM 密钥存储中创建新 KMS 密钥

您可以通过 Amazon KMS 控制台或使用CreateKey操作在密 Amazon CloudHSM 钥存储库中创建对称加密 KMS 密钥。

使用控制 Amazon KMS 台

使用以下过程在密钥库中创建对称加密 KMS Amazon CloudHSM 密钥。

> **Note**
>
> 不要在别名、描述或标签中包含机密或敏感信息。这些字段可能以纯文本形式出现在 CloudTrail 日志和其他输出中。

1. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/kms 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择客户托管密钥。
4. 选择创建密钥。
5. 选择 Symmetric (对称)。
6. 在 Key usage（密钥用法）中，已为您选择了 Encrypt and decrypt（加密和解密）选项。请勿对其进行更改。
7. 选择 Advanced options (高级选项)。
8. 对于密钥材料源，选择 Amazon CloudHSM 密钥存储。

   您无法在密钥库中创建多区域 Amazon CloudHSM 密钥。
9. 选择下一步。
10. 为您的新 KMS Amazon CloudHSM 密钥选择密钥存储区。要创建新的 Amazon CloudHSM 密钥库，请选择创建自定义密钥库。

    您选择的 Amazon CloudHSM 密钥库的状态必须为 “已连接”。其关联的 Amazon CloudHSM 集群必须处于活动状态，并且在不同的可用区中至少包含两个 HSMs 处于活动状态的集群。

    有关连接密 Amazon CloudHSM 钥库的帮助，请参阅断开 Amazon CloudHSM 密钥存储。有关添加的帮助 HSMs，请参阅Amazon CloudHSM 用户指南中的添加 HSM。
11. 选择下一步。
12. 为 KMS 密钥键入别名和可选的描述。
13. （可选）。在 Add Tags（添加标签）页面上，添加标识或分类 KMS 密钥的标签。

    向 Amazon 资源添加标签时， Amazon 会生成一份成本分配报告，其中包含按标签汇总的使用量和成本。标签还可以用来控制对 KMS 密钥的访问。有关轮换 KMS 密钥的信息，请参阅 标签在 Amazon KMS 和 ABAC for Amazon KMS。
14. 选择下一步。
15. 在 Key Administrators（密钥管理员）部分中，选择可管理 KMS 密钥的 IAM 用户和角色。有关更多信息，请参阅允许密钥管理员管理 KMS 密钥。

    > **注意**
    >
    > IAM policy 可以向其他 IAM 用户和角色授予使用 KMS 密钥的权限。

IAM 最佳实践不鼓励使用具有长期凭证的 IAM 用户。而应尽可能使用提供临时凭证的 IAM 角色。有关更多信息，请参阅《IAM 用户指南》中的 IAM 安全最佳实践。
Amazon KMS 控制台在语句标识符下将密钥管理员添加到密钥策略中"Allow access for Key Administrators"。如果修改此语句标识符，则可能会影响控制台显示您对该语句所做修改的方式。

16. （可选）要阻止这些密钥管理员删除此 KMS 密钥，请清除页面底部与 Allow key administrators to delete this key（允许密钥管理员删除此密钥）对应的框。
17. 选择下一步。
18. 在此账户部分中，选择可以在加密操作中使用 KMS 密钥的 IAM 用户和角色。 Amazon Web Services 账户 有关更多信息，请参阅允许密钥用户使用 KMS 密钥。

> **注意**
>
> IAM 最佳实践不鼓励使用具有长期凭证的 IAM 用户。而应尽可能使用提供临时凭证的 IAM 角色。有关更多信息，请参阅《IAM 用户指南》中的 IAM 安全最佳实践。
> Amazon KMS 控制台在语句标识符"Allow use of the key"和下将密钥用户添加到密钥策略中"Allow attachment of persistent resources"。如果修改这些语句标识符，则可能会影响控制台显示您对该语句所做修改的方式。

19. （可选）您可以允许其他人使用 Amazon Web Services 账户 此 KMS 密钥进行加密操作。为此，请在页面底部的 Amazon Web Services 账户“其他” 部分中，选择 “添加另一个”， Amazon Web Services 账户然后输入外部账户的 Amazon Web Services 账户 ID。要添加多个外部账户，请重复此步骤。

> **Note**
>
> 另一方的管理员还 Amazon Web Services 账户 必须通过为其用户创建 IAM 策略来允许访问 KMS 密钥。有关更多信息，请参阅 允许其他账户中的用户使用 KMS 密钥。

20. 选择下一步。
21. 检查密钥的密钥策略语句。要对密钥策略进行更改，请选择编辑。
22. 选择下一步。
23. 检视您选择的密钥设置。您仍然可以返回并更改所有设置。
24. 完成后，选择 Finish（完成）以创建密钥。

操作成功后，屏幕上会显示您选择的密钥存储区中的新 KMS Amazon CloudHSM 密钥。当您选择新 KMS 密钥的名称或别名时，其详细信息页面上的加密配置选项卡会显示 KMS 密钥的来源 (Amazon CloudHSM)、自定义密钥存储的名称、ID 和类型以及 Amazon CloudHSM 集群的 ID。如果此过程失败，则会出现一条描述失败的错误消息。

> **Tip**
>
> 要更轻松地识别自定义密钥存储中的 KMS 密钥，请在 Customer managed keys（客户托管密钥）页面上，将 Custom key store ID（自定义密钥存储 ID）列添加到显示中。单击右上角的齿轮图标并选择 Custom key store ID (自定义密钥存储 ID)。有关更多信息，请参阅 自定义控制台视图。

使用 Amazon KMS API

要在密钥库中创建新的 Amazon KMS key（KMS Amazon CloudHSM 密钥），请使用CreateKey操作。使用 `CustomKeyStoreId` 参数识别自定义密钥存储并指定 `Origin` 值 `AWS_CLOUDHSM`。

您可能还需要使用 `Policy` 参数指定密钥策略。您可以随时更改密钥策略 (PutKeyPolicy) 并添加可选元素，例如描述和标签。

本部分中的示例使用 Amazon Command Line Interface (Amazon CLI)，但您可以使用任何受支持的编程语言。

以下示例首先调用该DescribeCustomKeyStores操作以验证 Amazon CloudHSM 密钥库是否已连接到其关联的 Amazon CloudHSM 集群。默认情况下，此操作将返回您的账户和区域中的所有自定义密钥存储。要仅描述特定的 Amazon CloudHSM 密钥库，请使用其`CustomKeyStoreId`或`CustomKeyStoreName`参数（但不能同时使用两者）。

在运行此命令之前，请将示例自定义密钥存储 ID 替换为有效的 ID。

> **Note**
>
> 不要在 `Description` 或 `Tags` 字段中包含机密或敏感信息。这些字段可能以纯文本形式出现在 CloudTrail 日志和其他输出中。

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
{
```

```
   "CustomKeyStores": [
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleKeyStore",
      "CustomKeyStoreType": "Amazon CloudHSM key store",
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "CONNECTED"
   ],
}
```

下一个示例命令使用该DescribeClusters操作来验证与 (cluster-1a23b4cdefgExampleKeyStore) 关联的 Amazon CloudHSM 集群是否至少有两个处于活动状态。 HSMs如果集群少于两个 HSMs，则CreateKey操作将失败。

```
$ aws cloudhsmv2 describe-clusters
{
    "Clusters": [
        {
            "SubnetMapping": {
               ...
            },
            "CreateTimestamp": 1507133412.351,
            "ClusterId": "cluster-1a23b4cdefg",
            "SecurityGroup": "sg-865af2fb",
            "HsmType": "hsm1.medium",
            "VpcId": "vpc-1a2b3c4d",
            "BackupPolicy": "DEFAULT",
            "Certificates": {
                "ClusterCertificate": "-----BEGIN CERTIFICATE-----\...\n-----END
 CERTIFICATE-----\n"
            },
            "Hsms": [
                {
                    "AvailabilityZone": "us-west-2a",
                    "EniIp": "10.0.1.11",
                    "ClusterId": "cluster-1a23b4cdefg",
                    "EniId": "eni-ea8647e1",
                    "StateMessage": "HSM created.",
                    "SubnetId": "subnet-a6b10bd1",
                    "HsmId": "hsm-abcdefghijk",
                    "State": "ACTIVE"
                },
```

```
                {
                    "AvailabilityZone": "us-west-2b",
                    "EniIp": "10.0.0.2",
                    "ClusterId": "cluster-1a23b4cdefg",
                    "EniId": "eni-ea8647e1",
                    "StateMessage": "HSM created.",
                    "SubnetId": "subnet-b6b10bd2",
                    "HsmId": "hsm-zyxwvutsrqp",
                    "State": "ACTIVE"
                },
            ],
            "State": "ACTIVE"
        }
    ]
}
```

此示例命令使用CreateKey操作在密钥库中创建 KMS Amazon CloudHSM 密钥。要在密钥存储中创建 KMS Amazon CloudHSM 密钥，必须提供密钥存储库的自定义 Amazon CloudHSM 密钥存储 ID，并将`Origin`值指定为`AWS_CLOUDHSM`。

响应包括自定义密钥库和 Amazon CloudHSM 集群的。 IDs

在运行此命令之前，请将示例自定义密钥存储 ID 替换为有效的 ID。

```
$ aws kms create-key --origin AWS_CLOUDHSM --custom-key-store-id cks-1234567890abcdef0
{
  "KeyMetadata": {
    "AWSAccountId": "111122223333",
    "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "CreationDate": 1.499288695918E9,
    "Description": "Example key",
    "Enabled": true,
    "MultiRegion": false,
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "Origin": "AWS_CLOUDHSM"
    "CloudHsmClusterId": "cluster-1a23b4cdefg",
    "CustomKeyStoreId": "cks-1234567890abcdef0"
    "KeySpec": "SYMMETRIC_DEFAULT",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
```

```
    "EncryptionAlgorithms": [
        "SYMMETRIC_DEFAULT"
    ]
  }
}
```

在外部密钥存储中创建 KMS 密钥

创建并连接外部密钥存储库后，可以在密钥库 Amazon KMS keys 中创建。它们必须是源值为外部密钥存储（EXTERNAL_KEY_STORE）的对称加密 KMS 密钥。您不能在自定义密钥存储中创建非对称 KMS 密钥、HMAC KMS 密钥或具有导入的密钥材料的 KMS 密钥。此外，您不能在自定义密钥存储中使用对称加密 KMS 密钥来生成非对称数据密钥对。

与标准 KMS 密钥相比，外部密钥存储中 KMS 密钥的延迟、耐久性和可用性可能较差，因为这些密钥依赖位于 Amazon外部的组件。在外部密钥存储中创建或使用 KMS 密钥之前，请验证您是否需要具有外部密钥存储属性的密钥。

> **Note**
>
> 一些外部密钥管理器为在外部密钥存储中创建 KMS 密钥提供了更简单的方法。有关详细信息，请参阅外部密钥管理器的文档。

若要在外部密钥存储中创建 KMS 密钥，请指定以下内容：

- 外部密钥存储的 ID。
- 外部密钥存储（EXTERNAL_KEY_STORE）的密钥材料源。
- 与外部密钥存储关联的外部密钥管理器中现有外部密钥的 ID。此外部密钥用作 KMS 密钥的密钥材料。创建 KMS 密钥后，您无法更改外部密钥 ID。

  Amazon KMS 在请求加密和解密操作时，向您的外部密钥存储代理提供外部密钥 ID。 Amazon KMS 无法直接访问您的外部密钥管理器或其任何加密密钥。

除了外部密钥外，外部密钥存储中的 KMS 密钥还包含 Amazon KMS 密钥材料。在 KMS 密钥下加密的所有数据首先 Amazon KMS 使用密钥的密 Amazon KMS 钥材料进行加密，然后由您的外部密钥管理器使用您的外部密钥进行加密。这种双重加密过程可确保外部密钥存储中受 KMS 密钥保护的加密文字至少与仅受 Amazon KMS保护的加密文字一样强大。有关更多信息，请参阅 外部密钥存储的工作原理。

CreateKey 操作成功后，新 KMS 密钥的密钥状态为 Enabled。查看外部密钥存储中的 KMS 密钥时，您可以看到各种典型属性，例如其密钥 ID、密钥规范、密钥用法、密钥状态以及创建日期。但是您也可以看到外部密钥存储的 ID 和连接状态以及外部密钥的 ID。

如果您在外部密钥存储中创建 KMS 密钥的尝试失败，请查看错误消息以确定原因。错误消息可能表明外部密钥存储未连接（CustomKeyStoreInvalidStateException），您的外部密钥存储代理无法找到具有指定外部密钥 ID（XksKeyNotFoundException）的外部密钥，或者外部密钥已与同一外部密钥存储 XksKeyAlreadyInUseException 中的 KMS 密钥相关联。

有关在外部密钥存储中创建 KMS 密钥的操作 Amazon CloudTrail 日志的示例，请参阅CreateKey。

主题

外部密钥存储中 KMS 密钥的要求

若要在外部密钥存储中创建 KMS 密钥，外部密钥存储、KMS 密钥和用作 KMS 密钥外部加密密钥材料的外部密钥需要具有以下属性。

外部密钥存储要求

- 必须连接到其外部密钥存储代理。

  若要查看外部密钥存储的连接状态，请参阅 查看外部密钥存储。若要连接外部密钥存储，请参阅 连接和断开外部密钥存储。

KMS 密钥要求

创建 KMS 密钥后，您无法更改这些属性。

- 密钥规范：SYMMETRIC_DEFAULT
- 密钥用法：ENCRYPT_DECRYPT
- 密钥材料源：EXTERNAL_KEY_STORE
- 多区域：FALSE

外部密钥要求

- 256 位 AES 加密密钥（256 个随机位）。KeySpec 的外部密钥必须是 AES_256。
- 已启用并可供使用。Status 的外部密钥必须是 ENABLED。
- 已配置以进行密钥和解密。KeyUsage 的外部密钥必须包含 ENCRYPT 和 DECRYPT。
- 仅与此 KMS 密钥结合使用。外部密钥存储中的每个 KMS key 都必须与不同的外部密钥关联。

  Amazon KMS 还建议将外部密钥专门用于外部密钥存储。此限制更易于识别和解决密钥问题。
- 可由外部密钥存储的外部密钥存储代理访问。

  如果外部密钥存储代理无法使用指定的外部密钥 ID 找到密钥，则 CreateKey 操作将失败。
- 可以处理您的使用 Amazon Web Services 服务 产生的预期流量。 Amazon KMS 建议准备好外部密钥以每秒最多处理 1800 个请求。

在外部密钥存储中创建新的 KMS 密钥

您可以在 Amazon KMS 控制台或使用CreateKey操作在外部密钥存储中创建新的 KMS 密钥。

使用控制 Amazon KMS 台

有两种方法可以在外部密钥存储中创建 KMS 密钥。

- 方法 1（推荐）：选择外部密钥存储，然后在该外部密钥存储中创建 KMS 密钥。
- 方法 2：创建 KMS 密钥，然后指明该密钥在外部密钥存储中。

如果您使用方法 1，即在创建密钥之前选择外部密钥存储，请为您 Amazon KMS 选择所有必需的 KMS 密钥属性并填写外部密钥存储的 ID。此方法可让您避免在创建 KMS 密钥时可能犯的错误。

> **Note**
>
> 不要在别名、描述或标签中包含机密或敏感信息。这些字段可能以纯文本形式出现在 CloudTrail 日志和其他输出中。

方法 1（推荐）：在外部密钥存储中开始

若要使用此方法，请选择您的外部密钥存储，然后创建 KMS 密钥。 Amazon KMS 控制台会为您选择所有必需的属性并填写您的外部密钥存储库的 ID。此方法可让您避免在创建 KMS 密钥时可能犯的许多错误。

1. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/kms 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择 Custom key stores（自定义密钥存储）、External key stores（外部密钥存储）。
4. 选择外部密钥存储的名称。
5. 在右上角，选择 Create a KMS key in this key store（在此密钥存储中创建 KMS 密钥）。

   如果未连接外部密钥存储，系统将提示您将其连接。如果连接尝试失败，则需要解决问题并连接外部密钥存储，然后才能在其中创建新的 KMS 密钥。

   如果已连接外部密钥存储，您将被重定向到 Customer managed keys（客户托管密钥）页面以创建密钥。已为您选择了必需的 Key configuration（密钥配置）值。此外，外部密钥存储的自定义密钥存储 ID 已填写，但您可以对其进行更改。
6. 在外部密钥管理器中输入外部密钥的密钥 ID。此外部密钥必须满足与 KMS 密钥一起使用的要求。创建 KMS 密钥后，您无法更改此值。

   如果外部密钥有多个 IDs，请输入外部密钥存储代理用来识别外部密钥的密钥 ID。
7. 请确认您打算在指定的外部密钥存储中创建 KMS 密钥。
8. 选择下一步。

   此过程的其余步骤与创建标准 KMS 密钥的步骤相同。
9. 为 KMS 密钥键入别名（必需）和描述（可选）。
10. （可选）。在 Add Tags（添加标签）页面上，添加标识或分类 KMS 密钥的标签。

    向 Amazon 资源添加标签时， Amazon 会生成一份成本分配报告，其中包含按标签汇总的使用量和成本。标签还可以用来控制对 KMS 密钥的访问。有关轮换 KMS 密钥的信息，请参阅 标签在 Amazon KMS 和 ABAC for Amazon KMS。
11. 选择下一步。
12. 在 Key Administrators（密钥管理员）部分中，选择可管理 KMS 密钥的 IAM 用户和角色。有关更多信息，请参阅允许密钥管理员管理 KMS 密钥。

> **Note**
>
> IAM 策略可以向其他 IAM 用户和角色授予使用 KMS 密钥的权限。

IAM 最佳实践不鼓励使用具有长期凭证的 IAM 用户。而应尽可能使用提供临时凭证的 IAM 角色。有关更多信息，请参阅《IAM 用户指南》中的 IAM 安全最佳实践。

13. （可选）要阻止这些密钥管理员删除此 KMS 密钥，请清除 Allow key administrators to delete this key（允许密钥管理员删除此密钥）复选框。

    删除 KMS 密钥是一种具有破坏性且不可撤销的操作，将导致加密文字不可恢复。即使您拥有外部密钥材料，也无法在外部密钥存储中重新创建对称 KMS 密钥。但是，删除 KMS 密钥会影响关联的外部密钥。有关从外部密钥存储中删除 KMS 密钥的信息，请参阅删除密钥的特殊注意事项。

14. 选择下一步。

15. 在此账户部分中，选择可以在加密操作中使用 KMS 密钥的 IAM 用户和角色。 Amazon Web Services 账户 有关更多信息，请参阅允许密钥用户使用 KMS 密钥。

    > **Note**
    >
    > IAM policy 可以向其他 IAM 用户和角色授予使用 KMS 密钥的权限。
    > IAM 最佳实践不鼓励使用具有长期凭证的 IAM 用户。而应尽可能使用提供临时凭证的 IAM 角色。有关更多信息，请参阅《IAM 用户指南》中的 IAM 安全最佳实践。

16. （可选）您可以允许其他人使用 Amazon Web Services 账户 此 KMS 密钥进行加密操作。为此，请在页面底部的 Amazon Web Services 账户“其他” 部分中，选择 “添加另一个”， Amazon Web Services 账户然后输入外部账户的 Amazon Web Services 账户 ID。要添加多个外部账户，请重复此步骤。

    > **Note**
    >
    > 另一方的管理员还 Amazon Web Services 账户 必须通过为其用户创建 IAM 策略来允许访问 KMS 密钥。有关更多信息，请参阅 允许其他账户中的用户使用 KMS 密钥。

17. 选择下一步。

18. 检视您选择的密钥设置。您仍然可以返回并更改所有设置。

19. 完成后，选择 Finish（完成）以创建密钥。

方法 2：在客户托管密钥中开始

此过程与使用密钥材料创建对称加密密 Amazon KMS 钥的过程相同。但是，在此过程中，您需要指定外部密钥存储的自定义密钥存储 ID 和外部密钥的密钥 ID。您还必须为外部密钥存储中的 KMS 密钥指定必需的属性值，例如密钥规格和密钥用法。

1. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/km s 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择客户托管密钥。
4. 选择创建密钥。
5. 选择 Symmetric (对称)。
6. 在 Key usage（密钥用法）中，已为您选择了 Encrypt and decrypt（加密和解密）选项。请勿对其进行更改。
7. 选择 Advanced options (高级选项)。
8. 对于 Key material origin（密钥材料源），选择 External key store（外部密钥存储）。
9. 请确认您打算在指定的外部密钥存储中创建 KMS 密钥。
10. 选择下一步。
11. 请选择代表新 KMS 密钥外部密钥存储的行。

    您无法选择已断开连接的外部密钥存储。若要连接已断开连接的密钥存储，请选择密钥存储名称，然后从 Key store actions（密钥存储操作）中选择 Connect（连接）。有关更多信息，请参阅 使用 Amazon KMS 控制台。
12. 在外部密钥管理器中输入外部密钥的密钥 ID。此外部密钥必须满足与 KMS 密钥一起使用的要求。创建 KMS 密钥后，您无法更改此值。

    如果外部密钥有多个 IDs，请输入外部密钥存储代理用来识别外部密钥的密钥 ID。
13. 选择下一步。

    此过程的其余步骤与创建标准 KMS 密钥的步骤相同。
14. 为 KMS 密钥键入别名和可选的描述。
15. （可选）。在 Add Tags（添加标签）页面上，添加标识或分类 KMS 密钥的标签。

    向 Amazon 资源添加标签时， Amazon 会生成一份成本分配报告，其中包含按标签汇总的使用量和成本。标签还可以用来控制对 KMS 密钥的访问。有关轮换 KMS 密钥的信息，请参阅 标签在 Amazon KMS 和 ABAC for Amazon KMS。
16. 选择下一步。

17. 在 Key Administrators（密钥管理员）部分中，选择可管理 KMS 密钥的 IAM 用户和角色。有关更多信息，请参阅允许密钥管理员管理 KMS 密钥。

> **Note**
>
> IAM 策略可以向其他 IAM 用户和角色授予使用 KMS 密钥的权限。

18. （可选）要阻止这些密钥管理员删除此 KMS 密钥，请清除 Allow key administrators to delete this key（允许密钥管理员删除此密钥）复选框。

    删除 KMS 密钥是一种具有破坏性且不可撤销的操作，将导致加密文字不可恢复。即使您拥有外部密钥材料，也无法在外部密钥存储中重新创建对称 KMS 密钥。但是，删除 KMS 密钥会影响关联的外部密钥。有关从外部密钥存储中删除 KMS 密钥的信息，请参阅 删除 Amazon KMS key。

19. 选择下一步。

20. 在此账户部分中，选择可以在加密操作中使用 KMS 密钥的 IAM 用户和角色。 Amazon Web Services 账户 有关更多信息，请参阅允许密钥用户使用 KMS 密钥。

> **Note**
>
> IAM policy 可以向其他 IAM 用户和角色授予使用 KMS 密钥的权限。

21. （可选）您可以允许其他人使用 Amazon Web Services 账户 此 KMS 密钥进行加密操作。为此，请在页面底部的 Amazon Web Services 账户“其他” 部分中，选择 “添加另一个”， Amazon Web Services 账户然后输入外部账户的 Amazon Web Services 账户 ID。要添加多个外部账户，请重复此步骤。

> **Note**
>
> 另一方的管理员还 Amazon Web Services 账户 必须通过为其用户创建 IAM 策略来允许访问 KMS 密钥。有关更多信息，请参阅 允许其他账户中的用户使用 KMS 密钥。

22. 选择下一步。

23. 检视您选择的密钥设置。您仍然可以返回并更改所有设置。

24. 完成后，选择 Finish（完成）以创建密钥。

该过程成功后，显示内容将在您选择的外部密钥存储中显示新 KMS 密钥。选择新 KMS 密钥的名称或别名时，其详细信息页面上的 Cryptographic configuration（加密配置）选项卡会显示 KMS 密钥的源

[External key store（外部密钥存储）]，自定义密钥存储的名称、ID 和类型，以及外部密钥的 ID、密钥用法和状态。如果此过程失败，则会出现一条描述失败的错误消息。对于 ，请参阅 排查外部密钥存储的问题。

> **Tip**
>
> 若要更轻松地识别自定义密钥存储中的 KMS 密钥，请在 Customer managed keys（客户托管密钥）页面上，将 Origin（源）和 Custom key store ID（自定义密钥存储 ID）列添加到显示中。若要更改表格字段，请选择页面右上角的齿轮图标。有关更多信息，请参阅 自定义控制台视图。

使用 Amazon KMS API

要在外部密钥存储中创建新的 KMS 密钥，请使用CreateKey操作。以下参数为必需参数：

- `Origin` 值必须为 `EXTERNAL_KEY_STORE`。
- `CustomKeyStoreId` 参数标识您的外部密钥存储。指定外部密钥存储的 `ConnectionState` 必须是 `CONNECTED`。若要找到 `CustomKeyStoreId` 和 `ConnectionState`，请使用 `DescribeCustomKeyStores` 操作。
- `XksKeyId` 参数标识外部密钥。此外部密钥必须满足与 KMS 密钥关联的要求。

您也可以使用 `CreateKey` 操作的任何可选参数，例如使用 `Policy` 或 `Tags`（标签）参数。

> **Note**
>
> 不要在 `Description` 或 `Tags` 字段中包含机密或敏感信息。这些字段可能以纯文本形式出现在 CloudTrail 日志和其他输出中。

本部分中的示例使用 Amazon Command Line Interface (Amazon CLI)，但您可以使用任何受支持的编程语言。

此示例命令使用该CreateKey操作在外部密钥存储中创建 KMS 密钥。响应包括 KMS 密钥的属性、外部密钥存储的 ID 以及外部密钥的 ID、用法和状态。

在运行此命令之前，请将示例自定义密钥存储 ID 替换为有效的 ID。

```
$ aws kms create-key --origin EXTERNAL_KEY_STORE --custom-key-store-
id cks-1234567890abcdef0 --xks-key-id bb8562717f809024
{
  "KeyMetadata": {
    "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "AWSAccountId": "111122223333",
    "CreationDate": "2022-12-02T07:48:55-07:00",
    "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
    "CustomKeyStoreId": "cks-1234567890abcdef0",
    "Description": "",
    "Enabled": true,
    "EncryptionAlgorithms": [
      "SYMMETRIC_DEFAULT"
    ],
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyManager": "CUSTOMER",
    "KeySpec": "SYMMETRIC_DEFAULT",
    "KeyState": "Enabled",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "MultiRegion": false,
    "Origin": "EXTERNAL_KEY_STORE",
    "XksKeyConfiguration": {
      "Id": "bb8562717f809024"
    }
  }
}
```

识别和查看密钥

您可以使用Amazon Web Services 管理控制台或 Amazon Key Management Service (Amazon KMS) API 查看 Amazon KMS keys 每个账户和区域，包括您管理的 KMS 密钥和由管理的 KMS 密钥 Amazon。

主题

查找密钥 ID 和密钥 ARN

要识别 Amazon KMS key，您可以使用密钥 ID 或 Amazon Resource Name（密钥 ARN）。在加密操作中，您也可以使用别名名称或别名 ARN。

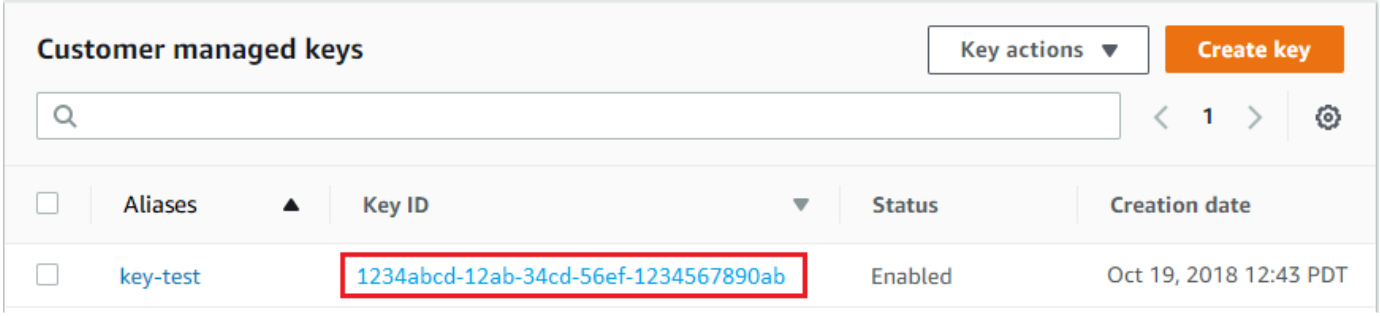
您可以使用 Amazon KMS 控制台或 ListKeys 操作来识别您的账户和区域中每个 KMS 密钥的密钥 ID 和密钥 ARN。

有关 Amazon KMS 支持的 KMS 密钥标识符的详细信息，请参阅 密钥标识符 (KeyId)。要获取查找别名和别名 ARN 的帮助，请参阅 查找 KMS 密钥的别名和别名 ARN。

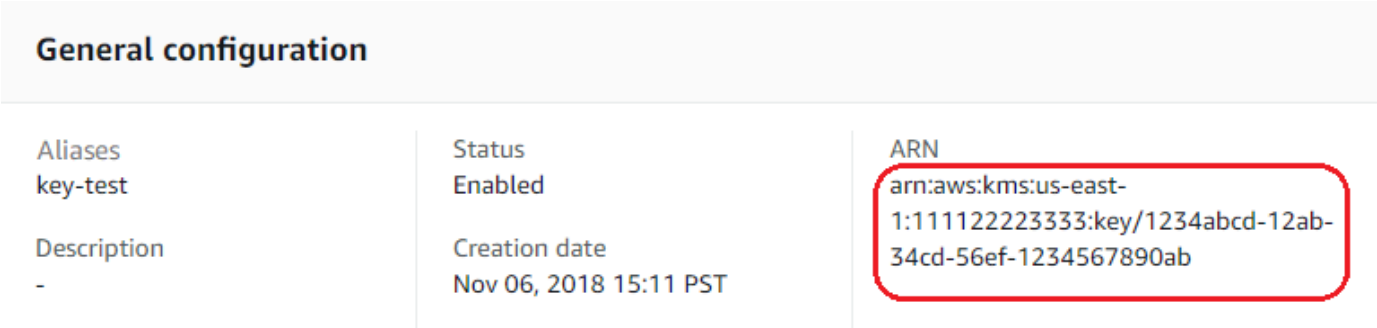
使用 Amazon KMS 控制台

1. 从 https://console.aws.amazon.com/kms 打开 Amazon KMS 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 要查看您账户中自己所创建和管理的密钥，请在导航窗格中选择 Customer managed keys (客户托管密钥)。要查看您账户中 Amazon 为您所创建和管理的密钥，请在导航窗格中选择 Amazon managed keys（Amazon 托管式密钥）。
4. 要查找 KMS 密钥的密钥 ID，请查看以 KMS 密钥别名开头的行。

   默认情况下，Key ID (密钥 ID) 列显示在表中。如果表中未显示“Key ID (密钥 ID)”列，可使用the section called “自定义控制台视图”中介绍的过程恢复该列。您还可以在 KMS 密钥的详细信息页面上查看其密钥 ID。



5. 要查找 KMS 密钥的 Amazon Resource Name (ARN)，请选择密钥 ID 或别名。密钥 ARN 显示在 General Configuration (常规配置) 部分中。



使用 Amazon KMS API

要查找 Amazon KMS key 的密钥 ID 和密钥 ARN，请使用 ListKeys 操作。

`ListKeys` 操作返回调用者账户和区域中所有 KMS 密钥的密钥 ID 和 Amazon 资源名称（ARN）。

例如，这个对 `ListKeys` 操作的调用会返回该虚构账户中每个 KMS 密钥的 ID 和 ARN。有关使用多种编程语言的示例，请参阅ListKeys与 Amazon SDK 或 CLI 配合使用。

```
$ aws kms list-keys
{
    "Keys": [
        {
            "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "KeyArn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        },
        {
            "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "KeyArn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
```

```
        }
    ]
}
```

访问并列出 KMS 密钥详细信息

您可以使用Amazon KMS 控制台或DescribeKey操作来访问和列出有关账户和区域中 KMS 密钥的详细信息。

以下过程演示如何访问 KMS 密钥详细信息，例如密钥 ID、密钥规范、密钥用法等。

使用控制 Amazon KMS 台

每个 KMS 密钥的详细信息页面均显示 KMS 密钥的属性。该页面会因 KMS 密钥类型而略有不同。

要显示有关 KMS 密钥的详细信息，请在Amazon 托管式密钥 或客户托管密钥页面上，选择 KMS 密钥的别名或密钥 ID。

KMS 密钥的详细信息页面中包括 General Configuration（常规配置）部分，其中显示 KMS 密钥的基本属性。此外还包括若干您可以用来查看和编辑 KMS 密钥属性的选项卡，例如密钥策略、加密配置、标签、密钥材料和轮换（适用于支持自动或按需轮换的 KMS 密钥）、区域性（适用于多区域密钥）和公有密钥（适用于非对称 KMS 密钥）。

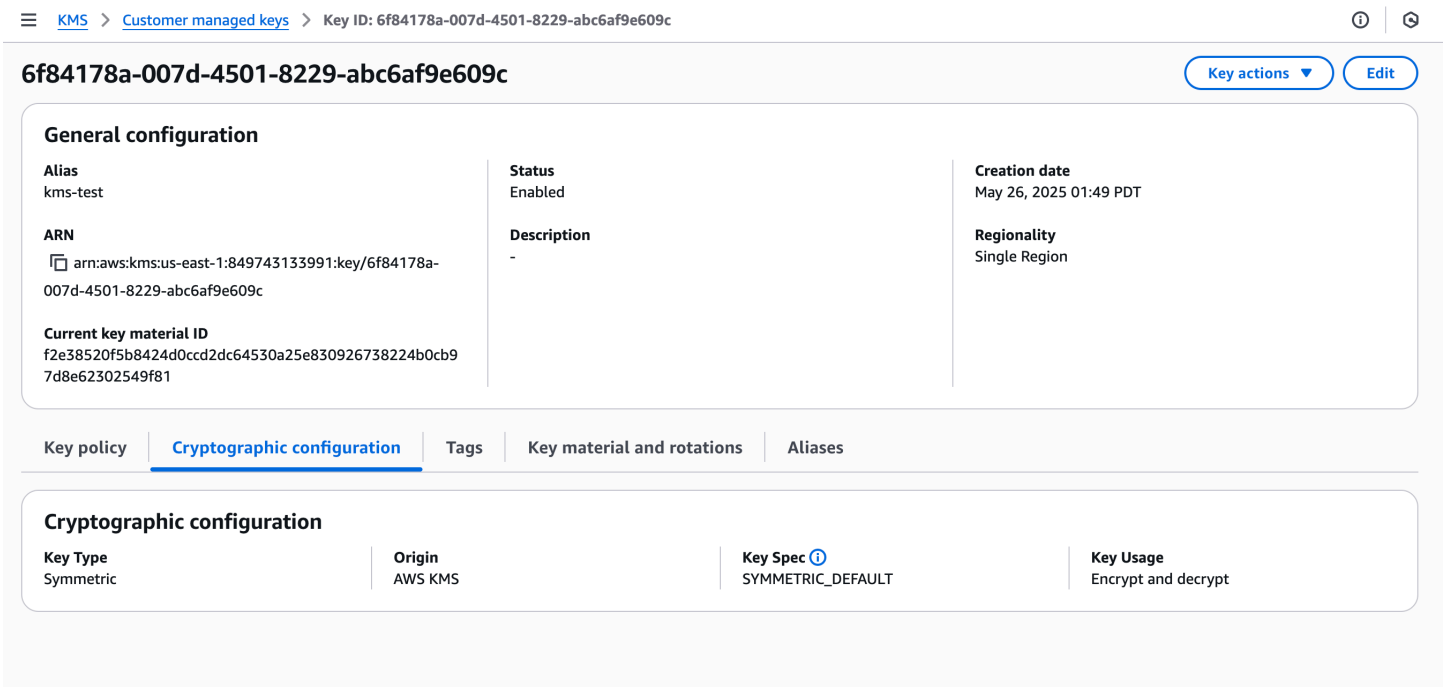
> **Note**
>
> Amazon KMS 控制台显示您在账户和区域中有权查看的 KMS 密钥。其他中的 KMS 密钥 Amazon Web Services 账户 不会出现在控制台中，即使您有权查看、管理和使用它们。要查看其他账户中的 KMS 密钥，请使用DescribeKey操作。

要导航至 KMS 密钥的密钥详细信息页。

1. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/km s 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 要查看您账户中自己所创建和管理的密钥，请在导航窗格中选择 Customer managed keys (客户托管密钥)。要查看您的账户中为您 Amazon 创建和管理的密钥，请在导航窗格中选择Amazon 托管密钥。

4. 要打开密钥详细信息页面，请在密钥表中，选择 KMS 密钥的密钥 ID 或别名。

   如果 KMS 密钥有多个别名，别名摘要（再加 n 个）将在其中一个别名的旁边显示。选择别名摘要将直接进入密钥详细信息页面上的 Aliases（别名）选项卡中。



以下列表描述了详细显示部分中的字段，包括选项卡中的字段。其中某些字段也在表格显示部分中以列的形式出现。

Aliases

位置：Aliases（别名）选项卡

KMS 密钥的友好名称。您可以使用别名在控制台和某些控制台中识别 KMS 密钥 Amazon KMS APIs。有关更多信息，请参阅 中的别名 Amazon KMS。

别名选项卡显示 Amazon Web Services 账户 和区域中与 KMS 密钥关联的所有别名。

进行筛选

位置：General configuration（常规配置）部分

KMS 密钥的 Amazon Resource Name (ARN)。此值唯一标识 KMS 密钥。您可以使用此值识别 Amazon KMS API 操作中的 KMS 密钥。

连接状态

表示自定义密钥存储是否已连接到其备用密钥存储。仅当在自定义密钥存储中创建 KMS 密钥时，此字段才会显示。

有关此字段值的信息，请参阅 Amazon KMS API 参考ConnectionState中的。

Creation date (创建日期)

位置：General configuration（常规配置）部分

KMS 密钥的创建日期和时间。此值显示为设备的本地时间。时区不依赖于区域。

与 Expiration（到期）不同，创建仅指的是 KMS 密钥，而非其密钥材料。

CloudHSM 集群 ID

位置：Cryptographic configuration（加密配置）选项卡

包含 KMS 密钥密钥材料的 Amazon CloudHSM 集群的集群 ID。仅当在自定义密钥存储中创建 KMS 密钥时，此字段才会显示。

如果您选择 CloudHSM 集群 ID，它将在控制台中打开集群页面。 Amazon CloudHSM

当前密钥材料

位置：General configuration（常规配置）部分

来源为 AWS_KMS 的对称加密密钥支持自动轮换和按需轮换。带EXTERNAL来源的对称加密密钥支持按需轮换。此类密钥可以关联多个密钥材料。最近轮换的密钥材料可用于加密和解密。此密钥材料被标识为当前密钥材料。其他密钥材料只能用于解密。KMS 密钥的自动或按需轮换会更改其当前密钥材料。

自定义密钥存储 ID

位置：Cryptographic configuration（加密配置）选项卡

包含 KMS 密钥的自定义密钥存储的 ID。仅当在自定义密钥存储中创建 KMS 密钥时，此字段才会显示。

如果您选择自定义密钥库 ID，它将在 Amazon KMS 控制台中打开自定义密钥存储页面。

自定义密钥存储名称

位置：Cryptographic configuration（加密配置）选项卡

包含 KMS 密钥的自定义密钥存储的名称。仅当在自定义密钥存储中创建 KMS 密钥时，此字段才会显示。

自定义密钥存储类型

位置：Cryptographic configuration（加密配置）选项卡

指示自定义密钥存储是 Amazon CloudHSM 密钥存储，还是外部密钥存储。仅当在自定义密钥存储中创建 KMS 密钥时，此字段才会显示。

描述

位置：General configuration（常规配置）部分

您可以编写和编辑的 KMS 密钥的简要、可选描述。要添加或更新客户托管密钥的描述，请选择 General Configuration（常规配置）上方的 Edit（编辑）。

加密算法

位置：Cryptographic configuration（加密配置）选项卡

列出可在 Amazon KMS中与 KMS 密钥一起使用的加密算法。仅当 Key type (密钥类型) 为 Asymmetric (对称) 且 Key usage (密钥用法) 为 Encrypt and decrypt (加密和解密) 时，此字段才会显示。有关 Amazon KMS 支持的加密算法的信息，请参阅SYMMETRIC_DEFAULT 密钥规范和用于加密和解密的 RSA 密钥规范。

到期日期

位置：Key material（密钥材料）选项卡

KMS 密钥的密钥材料到期的日期和时间。此字段仅针对具有导入的密钥材料的 KMS 密钥显示，也就是说，仅当 Origin（源）为 External（外部）且 KMS 密钥的密钥材料已到期时，此字段才会显示。对称加密密钥可以有多个与之关联的密钥材料。对于此类密钥，此字段指示所关联的密钥材料中最早到期的日期和时间。

外部密钥 ID

位置：Cryptographic configuration（加密配置）选项卡

与外部密钥存储中的 KMS 密钥关联的外部密钥的 ID。此字段仅适用于外部密钥存储中的 KMS 密钥。

外部密钥状态

位置：Cryptographic configuration（加密配置）选项卡

外部密钥存储代理报告的与 KMS 密钥关联的外部密钥的最新状态。此字段仅适用于外部密钥存储中的 KMS 密钥。

外部密钥用途

位置：Cryptographic configuration（加密配置）选项卡

在与 KMS 密钥关联的外部密钥上启用的加密操作。此字段仅适用于外部密钥存储中的 KMS 密钥。

密钥策略

位置：Key policy（密钥策略）选项卡

控制对 KMS 密钥以及 IAM policy 和授权的访问。每个 KMS 密钥都有一个密钥策略。它是唯一的强制性授权元素。要更改客户托管密钥的密钥策略，请在 Key policy（密钥策略）选项卡上选择 Edit（编辑）。有关更多信息，请参阅 the section called “密钥政策”。

密钥材料和轮换

位置：密钥材料和轮换选项卡

此选项卡仅适用于来源为 AWS_KMS 的对称加密密钥（支持自动和按需轮换）以及来源为 EXTERNAL 的单区域对称加密密钥（支持按需轮换）。

该选项卡有三个面板：

自动轮换：为客户自主管理型 KMS 密钥中的密钥材料启用和禁用自动轮换。要更改客户自主管理型密钥的密钥轮换状态，请使用此复选框。您无法启用或禁用 Amazon 托管式密钥 中的密钥材料轮换。 Amazon 托管式密钥 每年自动轮换一次。

按需轮换：为客户自主管理型密钥中的密钥材料启动按需轮换。对于导入的密钥，必须已有已导入的密钥材料处于 PENDING_ROTATION 状态，立即轮换选项才会可用。

密钥材料：列出与 KMS 密钥关联的所有密钥材料。每个密钥材料都有一个唯一的标识符，对应行显示有关该密钥材料的其他信息，例如密钥材料可供在 KMS 中使用的轮换日期。对于导入的密钥，每行还有一个操作菜单，可用于删除特定的密钥材料或将其重新导入到 KMS 密钥。

密钥规范

位置：Cryptographic configuration（加密配置）选项卡

KMS 密钥中密钥材料的类型。 Amazon KMS 支持对称加密 KMS 密钥 (SYMMETRIC_DEFAULT)、不同长度的 HMAC KMS 密钥、不同长度的 RSA 密钥的 KMS 密钥以及具有不同曲线的椭圆曲线密钥。有关更多信息，请参阅 Key spec。

密钥类型

位置：Cryptographic configuration（加密配置）选项卡

指示 KMS 密钥是 Symmetric（对称）还是 Asymmetric（非对称）的。

密钥用法

位置：Cryptographic configuration（加密配置）选项卡

指示 KMS 密钥可用于Encrypt and decrypt（加密和解密）、Sign and verify（签名和验证）或Generate and verify MAC（生成并验证 MAC）。有关更多信息，请参阅 Key usage。

Origin

位置：Cryptographic configuration（加密配置）选项卡

KMS 密钥的密钥材料的来源。有效值为：

- Amazon KMS，针对 Amazon KMS 生成的密钥材料
- Amazon CloudHSM，针对 Amazon CloudHSM 密钥存储中的 KMS 密钥
- External（外部），针对导入的密钥材料（BYOK）
- External key store（外部密钥存储），针对外部密钥存储中的 KMS 密钥

MAC 算法

位置：Cryptographic configuration（加密配置）选项卡

列出可在 Amazon KMS中与 HMAC KMS 密钥一起使用的 MAC 算法。仅当密钥规范是 HMAC 密钥规范（HMAC_*）时，此字段才会显示。有关 Amazon KMS 支持的 MAC 算法的信息，请参阅 HMAC KMS 密钥的密钥规范。

主键

位置：Regionality（区域性）选项卡

表示此 KMS 密钥是多区域主键。授权用户可以使用此部分将主键更改为另一个相关的多区域密钥。仅当 KMS 密钥是多区域主键时，此字段才会显示。

公有密钥

位置：Public key（公有密钥）选项卡

显示非对称 KMS 密钥的公有密钥。经授权的用户可以使用此选项卡复制和下载公有密钥。

区域性

位置：General configuration（常规配置）部分和 Regionality（区域性）选项卡

指示 KMS 密钥是单区域密钥、多区域主键，还是多区域副本密钥。仅当 KMS 密钥是多区域密钥时，此字段才会显示。

相关的多区域密钥

位置：Regionality（区域性）选项卡

显示所有相关多区域主键和副本键，但当前 KMS 密钥除外。仅当 KMS 密钥是多区域密钥时，此字段才会显示。

在主键的相关的多区域密钥部分，授权用户可以创建新的副本密钥。

副本密钥

位置：Regionality（区域性）选项卡

表示此 KMS 密钥是多区域副本密钥。仅当 KMS 密钥是多区域副本密钥时，此字段才会显示。

签名算法

位置：Cryptographic configuration（加密配置）选项卡

列出可在 Amazon KMS中与 KMS 密钥一起使用的签名算法。仅当 Key type (密钥类型) 为 Asymmetric (对称) 且 Key usage (密钥用法) 为 Sign and verify (签名和验证) 时，此字段才会显示。有关 Amazon KMS 支持的签名算法的信息，请参阅用于签名和验证的 RSA 密钥规范和椭圆曲线密钥规范。

Status

位置：General configuration（常规配置）部分

KMS 密钥的密钥状态。您可以在加密操作中使用 KMS 密钥，前提是仅当状态为 Enabled（已启用）时。有关每个 KMS 密钥状态的详细说明及其对可以在 KMS 密钥上运行的操作的影响，请参阅 Amazon KMS 密钥的密钥状态。

标记

位置：Tags（标签）选项卡

描述 KMS 密钥的可选键值对。要添加或更改 KMS 密钥的标签，请在 Tags（标签）选项卡上选择 Edit（编辑）。

向 Amazon 资源添加标签时，Amazon 会生成一份成本分配报告，其中包含按标签汇总的使用量和成本。标签还可以用来控制对 KMS 密钥的访问。有关轮换 KMS 密钥的信息，请参阅 标签在 Amazon KMS 和 ABAC for Amazon KMS。

使用 Amazon KMS API

该DescribeKey操作返回有关指定 KMS 密钥的详细信息。要标识 KMS 密钥，请使用密钥 ID、密钥 ARN、别名名称或别名 ARN。

与仅显示调用者账户和区域中的 KMS 密钥的ListKeys操作不同，授权用户可以使用该DescribeKey操作来获取有关其他账户中 KMS 密钥的详细信息。

> **Note**
>
> DescribeKey 响应包括具有相同值的两个 KeySpec 和 CustomerMasterKeySpec 成员。CustomerMasterKeySpec 成员已弃用。

例如，这个对 DescribeKey 的调用会返回有关对称加密 KMS 密钥的信息。响应中的字段因 Amazon KMS key 规范、密钥状态和密钥材料来源而异。有关使用多种编程语言的示例，请参阅DescribeKey与 Amazon SDK 或 CLI 配合使用。

```
$ aws kms describe-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab

{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "Enabled": true,
        "KeySpec": "SYMMETRIC_DEFAULT",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "CreationDate": 1499988169.234,
        "MultiRegion": false,
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "EncryptionAlgorithms": [
```

```
            "SYMMETRIC_DEFAULT"
        ],
        "CurrentKeyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    }
}
```

此示例对用于签名和验证的非对称 KMS 密钥调用 `DescribeKey` 操作。响应包括 Amazon KMS 支持用于此 KMS 密钥的签名算法。

```
$ aws kms describe-key --key-id 0987dcba-09fe-87dc-65ba-ab0987654321

{
    "KeyMetadata": {
        "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
        "Origin": "AWS_KMS",
        "Arn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
        "KeyState": "Enabled",
        "KeyUsage": "SIGN_VERIFY",
        "CreationDate": 1569973196.214,
        "Description": "",
        "KeySpec": "ECC_NIST_P521",
        "CustomerMasterKeySpec": "ECC_NIST_P521",
        "AWSAccountId": "111122223333",
        "Enabled": true,
        "MultiRegion": false,
        "KeyManager": "CUSTOMER",
        "SigningAlgorithms": [
            "ECDSA_SHA_512"
        ]
    }
}
```

识别不同的密钥类型

以下主题说明了如何在 Amazon KMS 控制台中识别不同的密钥类型和DescribeKey响应。

有关导航到 KMS 密钥详细信息页面上的加密配置选项卡的帮助，请参阅the section called “访问并列出 KMS 密钥详细信息”。

主题

- 识别非对称 KMS 密钥
- 识别 HMAC KMS 密钥
- 识别多区域 KMS 密钥
- 识别带导入的密钥材料的 KMS 密钥
- 识别密钥库中的 KMS Amazon CloudHSM 密钥
- 识别外部密钥存储中的 KMS 密钥

识别非对称 KMS 密钥

在 Amazon KMS 控制台中

客户托管密钥表的密钥类型列显示每个 KMS 密钥是对称还是非对称 KMS 密钥。您可以按密钥类型值筛选表，以仅显示非对称 KMS 密钥。有关更多信息，请参阅 the section called “对您的 KMS 密钥进行排序和筛选”。

KMS 密钥详细信息页面的加密配置选项卡显示密钥类型，用于指示密钥是对称还是非对称。它还会显示密钥用法，指示您的非对称 KMS 密钥是用于加密和解密、签名和验证，还是派生共享密钥。

在 DescribeKey 回应中

当您对非对称 KMS 密钥调用 `DescribeKey` 操作时，响应包括 `KeySpec` 和 `KeyUsage` 值，这些值可用于确定 KMS 密钥是对称还是非对称 KMS 密钥。

如果 `KeySpec` 值为 `SYMMETRIC_DEFAULT`，则密钥是对称加密 KMS 密钥。有关非对称密钥规范的详细信息，请参阅密钥规范引用。

如果 `KeyUsage` 值为 `SIGN_VERIFY` 或 `KEY_AGREEMENT`，则密钥为非对称 KMS 密钥。

对于非对称 KMS 密钥，`DescribeKey` 操作还会返回以下详细信息。

- 对于 `KeyUsage` 值为 `ENCRYPT_DECRYPT` 的非对称 KMS 密钥，该操作将返回 `EncryptionAlgorithms`，其中列出了该密钥的有效加密算法。
- 对于 `KeyUsage` 值为 `SIGN_VERIFY` 的非对称 KMS 密钥，该操作将返回 `SigningAlgorithms`，其中列出了该密钥的有效签名算法。
- 对于 `KeyUsage` 值为 `KEY_AGREEMENT` 的非对称 KMS 密钥，该操作将返回 `KeyAgreementAlgorithms`，其中列出了该密钥的有效密钥协议算法。

有关非对称 KMS 密钥的更多信息，请参阅the section called “非对称密钥”。

识别 HMAC KMS 密钥

在 Amazon KMS 控制台中

HMAC KMS 密钥包含在客户托管密钥表中，但您无法按标识 HMAC 密钥的密钥规格或密钥用法值对该表进行排序或筛选。为了更轻松地查找 HMAC 密钥，为其分配独特的别名或标签。然后，您可以按别名或标签进行排序或筛选。

KMS 密钥详细信息页面的加密配置选项卡显示密钥类型，用于指示密钥是对称还是非对称。HMAC KMS 密钥是对称密钥。加密配置选项卡还显示密钥用法。对于 HMAC KMS 密钥，密钥用法值始终为生成并验证 MAC。

在 DescribeKey 回应中

对 HMAC KMS 密钥调用 `DescribeKey` 操作时，响应将包括 `KeySpec` 和 `KeyUsage` 值。对于 HMAC KMS 密钥，密钥用法值始终为 `GENERATE_VERIFY_MAC`，密钥规范值始终以 `HMAC_` 开头。

有关 HMAC KMS 密钥的更多信息，请参阅 the section called “HMAC 密钥”。

识别多区域 KMS 密钥

在 Amazon KMS 控制台中

客户托管密钥表仅显示所选区域中的 KMS 密钥。您可以查看所选区域中的多区域主密钥和副本密钥。要更改 Amazon 区域，请使用控制台右上角的区域选择器。

为了更轻松地识别客户托管密钥表中的多区域密钥，请在表中添加区域性列。有关帮助信息，请参阅 the section called “自定义您的 KMS 密钥表”。

多区域 KMS 密钥的详细信息页面包含区域性选项卡。主密钥的 Regionality（区域性）选项卡包括 Change primary Region（更改主区域）和 Create new replica keys（创建新的副本密钥）按钮。（副本密钥的 Regionality（区域性）选项卡没有任何按钮。）Related multi-Region keys（相关的多区域密钥）部分列出了与当前密钥相关的所有多区域密钥。如果当前密钥是副本密钥，则此列表将包括主密钥。

如果您从 “相关多区域密钥” 表中选择相关的多区域密钥，则 Amazon KMS 控制台将更改为所选密钥的区域，并打开该密钥的详细信息页面。例如，如果您从下面的相关多区域密钥示例部分中选择**`sa-east-1`**区域中的副本密钥，则 Amazon KMS 控制台将更改为`sa-east-1`区域以显示该副本密钥的详细信息页面。您可以执行此操作来查看副本密钥的别名或密钥策略。要再次更改区域，请使用页面右上角的 Region selector（区域选择器）。

在 DescribeKey 回应中

默认情况下， Amazon KMS API 操作是区域性的，仅返回当前或指定区域中的资源。但是，当您对多区域 KMS 密钥调用DescribeKey操作时，响应会将元素中其他 Amazon 区域的所有相关多区域密钥包括在内。MultiRegionConfiguration

有关多区域 KMS 密钥的更多信息，请参阅 the section called “多区域密钥”。

识别带导入的密钥材料的 KMS 密钥

在 Amazon KMS 控制台中

为了更轻松地识别客户托管密钥表中带导入的密钥材料的 KMS 密钥，请在表中添加源列。“源”列可让您轻松标识具有外部（导入密钥材料）源属性值的 KMS 密钥。有关帮助信息，请参阅 the section called “自定义您的 KMS 密钥表”。

KMS 密钥详细信息页面上的加密配置选项卡显示了源，标识 KMS 密钥的密钥材料的来源。对于带导入密钥材料的 KMS 密钥，源值始终为外部（导入密钥材料）。详细信息页面还包括密钥材料选项卡，该选项卡提供有关导入的密钥材料的详细信息。带有 EXTERNAL Origin 的对称加密密钥支持按需轮换，并且可以有多个与之关联的密钥材料。对于此类密钥，该选项卡标记为密钥材料和轮换。

在 DescribeKey 回应中

对带导入的密钥材料的 KMS 密钥调用 DescribeKey 操作时，响应将包括 Origin、ExpirationModel 和 ValidTo 值。对于带导入密钥材料的 KMS 密钥，源值始终为 EXTERNAL。ExpirationModel 值指示密钥材料是否设置为过期，ValidTo 值指示密钥材料何时过期。当多个密钥材料关联到一个密钥时，ValidTo 值指示所有密钥材料的最早到期时间（待轮换的密钥材料除外），并且仅在这些密钥材料均未设置为过期时才将 ExpirationModel 设置为 DOES_NOT_EXPIRE。有关更多信息，请参阅 设置过期时间（可选）。

有关带导入的密钥材料的 KMS 密钥的更多信息，请参阅 the section called “导入的密钥材料”。

识别密钥库中的 KMS Amazon CloudHSM 密钥

在 Amazon KMS 控制台中

为了便于在客户管理的密钥表中识别密 Amazon CloudHSM 钥存储区中的 KMS 密钥，请在表中添加 Origin 列。来源列可让您使用 Amazon CloudHSM 来源属性值轻松标识 KMS 密钥。有关帮助信息，请参阅 the section called “自定义您的 KMS 密钥表”。

KMS 密钥详细信息页面上的加密配置选项卡显示了源，标识 KMS 密钥的密钥材料的来源。对于密 Amazon CloudHSM 钥存储中的 KMS 密钥，原始值始终为Amazon CloudHSM。

对于密钥存储中的 KMS Amazon CloudHSM 密钥，“加密配置” 选项卡包括一个额外的部分 “自定义密钥存储”，该部分提供与 KMS Amazon CloudHSM 密钥关联的密钥存储和 Amazon CloudHSM 集群的相关信息。

在 DescribeKey 回应中

对 Amazon CloudHSM 密钥存储中的 KMS 密钥调用 `DescribeKey` 操作时，响应将包括 `Origin`，其将标识密钥材料来源。对于密钥存储中的 KMS Amazon CloudHSM 密钥，原始值始终为`AWS_CLOUDHSM`。该操作还会返回密钥存储区中 KMS 密 Amazon CloudHSM 钥的以下特殊字段：

- `CloudHsmClusterId`
- `CustomKeyStoreId`

有关 Amazon CloudHSM 密钥库的更多信息，请参阅the section called “Amazon CloudHSM 密钥存储”。

识别外部密钥存储中的 KMS 密钥

在 Amazon KMS 控制台中

为了更轻松地识别客户托管密钥表中外部密钥存储中的 KMS 密钥，请在表中添加源列。源列可让您轻松识别带外部密钥存储源属性值的 KMS 密钥。有关帮助信息，请参阅 the section called “自定义您的 KMS 密钥表”。

KMS 密钥详细信息页面上的加密配置选项卡显示了源，标识 KMS 密钥的密钥材料的来源。对于外部密钥存储中的 KMS 密钥，源值均为外部密钥存储。

对于外部密钥存储中的 KMS 密钥，加密配置选项卡包括另外两个部分：自定义密钥存储和外部密钥。自定义密钥存储表提供有关与 KMS 密钥关联的外部密钥存储的信息。外部密钥表显示在 Amazon KMS 控制台中，仅适用于外部密钥存储中的 KMS 密钥。此部分提供有关与 KMS 密钥关联的外部密钥的信息。外部密钥是外部的加密密钥 Amazon ，用作外部密钥存储中 KMS 密钥的密钥材料。使用 KMS 密钥加密或解密时，将由外部密钥管理器使用指定的外部密钥执行此操作。

以下值显示在 External key（外部密钥）部分中。

外部密钥 ID

外部密钥管理器中外部密钥的标识符。这是外部密钥存储代理用来识别外部密钥的值。外部密钥 ID 在您创建 KMS 密钥时指定，并且无法更改。如果您用于创建 KMS 密钥的外部密钥 ID 值更改或失效，则必须安排删除 KMS 密钥，并使用正确的外部密钥 ID 值创建新的 KMS 密钥。

在 DescribeKey 回应中

对外部密钥存储中的 KMS 密钥调用 `DescribeKey` 操作时，响应将包括 `Origin`，其将标识密钥材料来源。对于密钥存储中的 KMS Amazon CloudHSM 密钥，原始值始终为`EXTERNAL_KEY_STORE`。该操作还会返回 `CustomKeyStoreId` 元素，该元素标识与 KMS 密钥关联的外部密钥存储。

有关外部密钥存储的更多信息，请参阅 the section called “外部密钥存储”。

自定义控制台视图

您可以自定义 Amazon KMS 控制台视图，以便更轻松地找到您的 KMS 密钥。自定义显示在 Amazon 托管式密钥 和客户托管密钥页面上的表以显示您最需要的信息，或者对表中返回的 KMS 密钥进行排序和筛选。

主题

- 对您的 KMS 密钥进行排序和筛选
- 自定义您的 KMS 密钥表

对您的 KMS 密钥进行排序和筛选

为了更轻松地在控制台中查找 KMS 密钥，可以对密钥表进行排序和筛选。

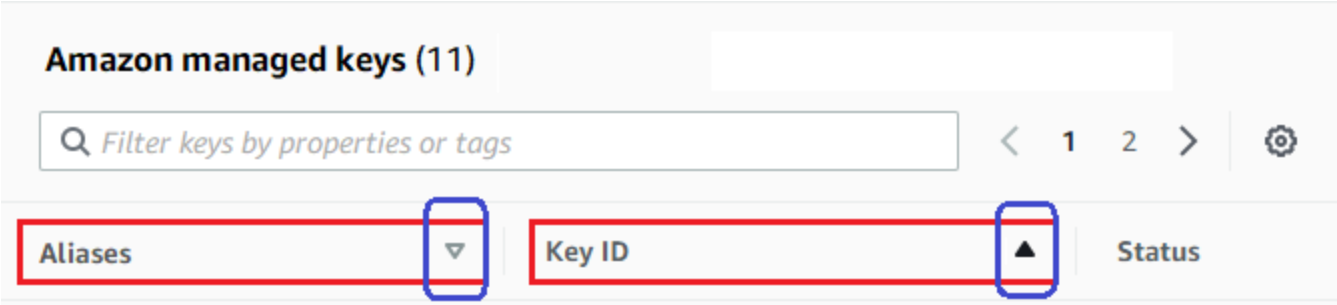
排序

您可以按照升序或降序，对 KMS 密钥的列值进行排序。此功能会对表中的所有 KMS 密钥进行排序，即使 KMS 密钥未显示在当前表页上。

可排序的列由列名称旁边的箭头指示。在 Amazon 托管式密钥 页面上，您可以按 Aliases（别名）或 Key ID（密钥 ID）排序。在 Customer managed keys（客户托管密钥）页面上，可以按 Aliases（别名）、Key ID（密钥 ID）或 Key type（密钥类型）进行排序。

要按升序排列，请选择列标题，直至箭头向上。要按降序排列，请选择列标题，直至箭头向下。一次只能按一列排序。

例如，可以按照密钥 ID 的升序对 KMS 密钥进行排序（而不是默认设置的别名）。



当您在 Customer managed keys（客户托管密钥）页面上按 Key type（密钥类型）以升序对 KMS 密钥进行排序时，所有非对称密钥都显示在所有对称密钥之前。

筛选条件

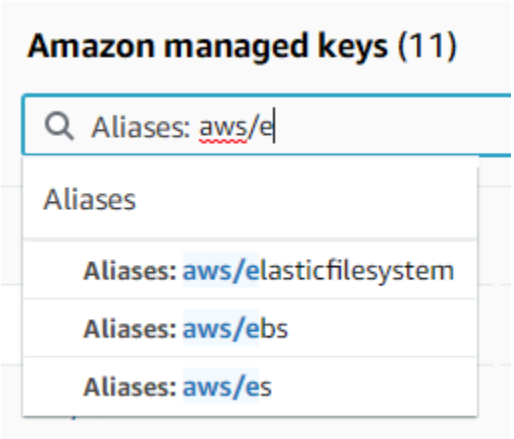
您可以按照 KMS 密钥的属性值或标签筛选 KMS 密钥。筛选条件将应用于表中的所有的 KMS 密钥，即使 KMS 密钥未显示在当前表页上。筛选不区分大小写。

筛选框中列出了可筛选的属性。在 Amazon 托管式密钥 页面上，可以按别名和密钥 ID 进行筛选。在 Customer managed keys（客户托管密钥）页面上，可以按别名、密钥 ID、密钥类型属性和标签进行筛选。

- 在 Amazon 托管式密钥 页面上，可以按别名和密钥 ID 进行筛选。
- 在 Customer managed keys（客户托管密钥）页面上，可以按标签、别名、密钥 ID、密钥类型和区域性属性进行筛选。

要按属性值进行筛选，请选择筛选条件、属性名称，然后从实际属性值的列表中进行选择。要按标签进行筛选，请选择标签键，然后从实际标签值列表中进行选择。选择属性或标签键后，还可以键入全部或部分属性值或标签值。在做出选择之前，将看到结果的预览。

例如，要显示别名名称中包含 `aws/e` 的 KMS 密钥，请选择筛选框，选择 Alias（别名），键入 `aws/e`，然后按 `Enter` 或 `Return` 添加筛选条件。



建议的 KMS 密钥表筛选条件

筛选非对称 KMS 密钥

要在 Customer managed keys（客户托管密钥）页面上仅显示非对称 KMS 密钥，请单击筛选条件框，选择 Key type（密钥类型），然后选择 Key type: Asymmetric（密钥类型：非对称）。仅当您在表中具有非对称 KMS 密钥时，才会显示 Asymmetric（非对称）选项。

筛选多区域密钥

要仅显示多区域密钥，请在 Customer managed keys（客户托管密钥）页面上，选择筛选框，选择 Regionality（区域性），然后选择 Regionality: Multi-Region（区域性：多区域）。Multi-Region（多区域）选项仅当您在表中具有多区域密钥时才会显示。

筛选标签

要仅显示具有特定标签的 KMS 密钥，请选择筛选框，选择标签键，然后从实际标签值中进行选择。还可以键入全部或部分标签值。

生成的表格将显示带有所选标签的所有 KMS 密钥。但是，它不显示标签。要查看标签，请选择 KMS 密钥的密钥 ID 或别名，然后在其详细信息页面上选择 Tags（标签）选项卡。别名显示在 General configuration（常规配置）部分下。

此筛选条件同时需要标签键和标签值。它不会通过仅键入标签键或仅键入标签值来找到 KMS 密钥。要按全部或部分标签键或值筛选标签，请使用ListResourceTags操作获取带标签的 KMS 密钥，然后使用您的编程语言的筛选功能。

按文本筛选

要搜索文本，请在筛选框中键入全部或部分别名、密钥 ID、密钥类型或标签键。（选择标签键后，您可以搜索标签值）。在做出选择之前，将看到结果的预览。

例如，要显示其标签键或可筛选属性中有 `test` 的 KMS 密钥，请在筛选框中键入 `test`。预览会显示筛选条件将选择的 KMS 密钥。在这种情况下，`test` 仅显示在别名属性中。

自定义您的 KMS 密钥表

您可以根据需要自定义中显示在Amazon 托管式密钥和客户管理的密钥页面上的表格。 Amazon Web Services 管理控制台 您可以选择表格的列数、每页的 Amazon KMS keys 数量（页面大小）和文本换行。所选择的配置将在确认后保存，并在打开页面时重新应用。

要自定义您的 KMS 密钥表

1. 在 Amazon 托管式密钥 或 Customer managed keys（客户托管密钥）页面上，选择页面右上角的设置图标

   ()。

2. 在 Preferences (首选项) 页面上，选择您的首选设置，然后选择 Confirm (确认)。

请考虑使用 Page size（页面大小）设置来增加每个页面上显示的 KMS 密钥数量，尤其当您通常使用易于滚动的设备时。

显示的数据列可能因表、作业角色以及账户和区域中的 KMS 密钥类型而异。下表提供了一些建议的配置。有关列的描述，请参阅使用控制 Amazon KMS 台。

建议的 KMS 密钥表配置

您可以自定义 KMS 密钥表中显示的列，以便显示所需的 KMS 密钥相关信息。

Amazon 托管式密钥

默认情况下，Amazon 托管式密钥 表会显示别名、密钥 ID 和状态列。这些列适合于大多数使用案例。

对称加密 KMS 密钥

如果只使用具有 Amazon KMS生成的密钥材料的对称加密 KMS 密钥，则别名、密钥 ID、状态和创建日期列可能是最有用的。

非对称 KMS 密钥

如果使用的是非对称 KMS 密钥，则除别名、密钥 ID 和状态列之外，可考虑添加密钥类型、密钥规范和密钥用法列。这些列将显示 KMS 密钥是对称还是非对称的、密钥材料的类型，以及 KMS 密钥是用于加密还是签名。

HMAC KMS 密钥

如果使用的是 HMAC KMS 密钥，则除别名、密钥 ID 和状态列之外，可考虑添加密钥规范和密钥用法列。这些列将显示 KMS 密钥是否是 HMAC 密钥。由于您无法按密钥规格或密钥使用情况对 KMS 密钥进行排序，因此请使用别名和标签来识别 HMAC 密钥，然后使用 Amazon KMS 控制台的筛选功能按别名或标签进行筛选。

导入的密钥材料

如果您的 KMS 密钥具有导入的密钥材料，请考虑添加 Origin（来源）和 Expiration date（到期日期）列。这些列将向您显示 KMS 密钥中的密钥材料是由导入还是生成的， Amazon KMS 以及密钥材料何时过期（如果有的话）。Creation date（创建日期）字段显示了 KMS 密钥的创建日期（不包含密钥材料）。该字段不体现密钥材料的任何特征。

自定义密钥存储中的密钥

如果您拥有自定义密钥存储中的 KMS 密钥，请考虑添加 Origin（来源）和 Custom key store ID（自定义密钥存储 ID）列。这些列表明 KMS 密钥位于自定义密钥存储中，显示自定义密钥存储类型，并标识自定义密钥存储。

多区域密钥

如果您有多区域密钥，请考虑添加 Regionality（区域性）列。这显示 KMS 密钥是单区域密钥、多区域主键，还是多区域副本密钥。

在 Amazon CloudHSM 密钥存储中查找 KMS 密钥

如果您管理 Amazon CloudHSM 密钥存储，则可能需要在每个 Amazon CloudHSM 密钥存储中标识 KMS 密钥。例如，您可能需要执行以下某些任务。

- 在 Amazon CloudTrail 日志中跟踪 Amazon CloudHSM 密钥存储中的 KMS 密钥。
- 预测断开 Amazon CloudHSM 密钥存储对 KMS 密钥的影响。
- 在删除 Amazon CloudHSM 密钥存储之前计划删除 KMS 密钥。

此外，您可能需要标识 Amazon CloudHSM 集群中用作 KMS 密钥的密钥材料的密钥。尽管 Amazon KMS 管理 KMS 密钥和密钥材料，但您仍可以控制并负责管理 Amazon CloudHSM 集群、HSM 和备份以及 HSM 中的密钥。您可能需要标识密钥以便审核密钥材料，防止意外删除密钥材料或在删除 KMS 密钥后将密钥材料从 HSM 和集群备份中删除。

Amazon CloudHSM 密钥存储中 KMS 密钥的所有密钥材料由 `kmsuser` 加密用户（CU）拥有。Amazon KMS 将密钥标签属性（该属性仅在 Amazon CloudHSM 中可查看）设置为 KMS 密钥的 Amazon 资源名称（ARN）。

要查找 KMS 密钥和密钥材料，请使用下列任一方法。

- 在 Amazon CloudHSM 密钥存储中查找 KMS 密钥 – 如何在一个或所有 Amazon CloudHSM 密钥存储中标识 KMS 密钥。
- 查找 Amazon CloudHSM 密钥存储的所有密钥 – 如何在集群中查找用作 Amazon CloudHSM 密钥存储中 KMS 密钥的密钥材料的所有密钥。
- 查找 KMS 密钥的 Amazon CloudHSM 密钥 – 如何在集群中查找用作 Amazon CloudHSM 密钥存储中特定 KMS 密钥的密钥材料的密钥。
- 查找 Amazon CloudHSM 密钥的 KMS 密钥 — 如何在集群中查找特定密钥的 KMS 密钥。

在 Amazon CloudHSM 密钥存储中查找 KMS 密钥

如果您管理 Amazon CloudHSM 密钥存储，则可能需要在每个 Amazon CloudHSM 密钥存储中标识 KMS 密钥。您可以使用此信息在 Amazon CloudTrail 日志中跟踪 KMS 密钥操作，预测断开自定义密钥存储对 KMS 密钥的影响，或在删除 Amazon CloudHSM 密钥存储前计划删除 KMS 密钥。

在 Amazon CloudHSM 密钥存储中查找 KMS 密钥（控制台）

要在特定的 Amazon CloudHSM 密钥存储中查找 KMS 密钥，请在 Customer managed keys（客户托管密钥）页面上，查看 Custom Key Store Name（自定义密钥存储名称）或 Custom Key Store ID（自定义密钥存储 ID）字段中的值。要确定任何 Amazon CloudHSM 密钥存储中的 KMS 密钥，请查找 Origin（源）值为 Amazon CloudHSM 的 KMS 密钥。要向显示添加可选列，请选择页面右上角的齿轮图标。

在 Amazon CloudHSM 密钥存储中查找 KMS 密钥（API）

要在 Amazon CloudHSM 密钥存储中查找 KMS 密钥，请使用 ListKeys 和 DescribeKey 操作，然后按 `CustomKeyStoreId` 值进行筛选。在运行以下示例之前，请将虚构的自定义密钥存储 ID 值替换为有效值。

Bash

要在特定的 Amazon CloudHSM 密钥存储中查找 KMS 密钥，请获取账户和区域中的所有 KMS 密钥。然后，按自定义密钥存储 ID 进行筛选。

```
for key in $(aws kms list-keys --query 'Keys[*].KeyId' --output text) ;
do aws kms describe-key --key-id $key |
grep '"CustomKeyStoreId": "cks-1234567890abcdef0"' --context 100; done
```

要获取账户和区域中任何 Amazon CloudHSM 密钥存储的 KMS 密钥，请搜索值为 `AWS_CloudHSM` 的 `CustomKeyStoreType`。

```
for key in $(aws kms list-keys --query 'Keys[*].KeyId' --output text) ;
do aws kms describe-key --key-id $key |
grep '"CustomKeyStoreType": "AWS_CloudHSM"' --context 100; done
```

PowerShell

要在特定的 Amazon CloudHSM 密钥存储中查找 KMS 密钥，请使用 Get-KmsKeyList 和 Get-KmsKey cmdlet 获取账户和区域中的所有 KMS 密钥。然后，按自定义密钥存储 ID 进行筛选。

```
PS C:\> Get-KMSKeyList | Get-KMSKey | where CustomKeyStoreId -eq
 'cks-1234567890abcdef0'
```

要获取账户和区域中任何 Amazon CloudHSM 密钥存储的 KMS 密钥，请筛选 `AWS_CLOUDHSM` 的 CustomKeyStoreType 值。

```
PS C:\> Get-KMSKeyList | Get-KMSKey | where CustomKeyStoreType -eq 'AWS_CLOUDHSM'
```

查找 Amazon CloudHSM 密钥存储的所有密钥

您可以标识 Amazon CloudHSM 集群中用作 Amazon CloudHSM 密钥存储的密钥材料的密钥。为此，请在 CloudHSM CLI 中使用 key list 命令。

您也可以使用 key list 命令来查找 Amazon CloudHSM 密钥的 Amazon KMS。当 Amazon KMS 在 Amazon CloudHSM 集群中为 KMS 密钥创建密钥材料时，它会在密钥标签中写入 KMS 密钥的 Amazon Resource Name (ARN)。key list 命令返回 `key-reference` 和 `label`。

> **备注**
>
> 以下过程使用 Amazon CloudHSM 客户端 SDK 5 命令行工具 CloudHSM CLI。CloudHSM CLI 将 `key-handle` 替换为 `key-reference`。
> 2025 年 1 月 1 日，Amazon CloudHSM 将终止对客户端 SDK 3 命令行工具、CloudHSM 管理实用程序（CMU）和密钥管理实用程序（KMU）的支持。有关客户端 SDK 3 命令行工具和客户端 SDK 5 命令行工具之间区别的更多信息，请参阅《Amazon CloudHSM 用户指南》中的从客户端 SDK 3 CMU 和 KMU 迁移到客户端 SDK 5 CloudHSM CLI。

要运行此过程，您需要临时断开 Amazon CloudHSM 密钥存储，以便能以 `kmsuser` CU 身份登录。

1. 断开 Amazon CloudHSM 密钥存储（如果尚未断开），然后以 `kmsuser` 身份登录，如 如何断开和登录 中所述。

   > **Note**
   >
   > 虽然自定义密钥存储已断开连接，但在自定义密钥存储中创建 KMS 密钥或在加密操作中使用现有 KMS 密钥的所有尝试都将失败。此操作可以阻止用户存储和访问敏感数据。

2. 在 CloudHSM CLI 中使用 key list 命令查找 Amazon CloudHSM 集群中当前用户的所有密钥。

   默认情况下，仅显示当前登录用户的 10 个按键，并且输出中仅显示 `key-reference` 和 `label`。有关更多选项，请参阅《Amazon CloudHSM 用户指南》中的密钥列表。

```
aws-cloudhsm > key list
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x0000000000000123",
        "attributes": {
          "label": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
      },
      {
        "key-reference": "0x0000000000000456",
        "attributes": {
```

```
          "label": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
      },.
      ...8 keys later...
    ],
    "total_key_count": 56,
    "returned_key_count": 10,
    "next_token": "10"
  }
}
```

3. 注销并重新连接 Amazon CloudHSM 密钥存储，如 如何注销并重新连接 中所述。

查找 Amazon CloudHSM 密钥的 KMS 密钥

如果您知道 kmsuser 在集群中拥有的密钥的密钥参考或 ID，则可以使用该值来标识 Amazon CloudHSM 密钥存储中的关联 KMS 密钥。

当 Amazon KMS 在 Amazon CloudHSM 集群中为 KMS 密钥创建密钥材料时，它会在密钥标签中写入 KMS 密钥的 Amazon Resource Name (ARN)。除非您已更改标签值，否则可在 CloudHSM CLI 中使用 key list 命令来识别与 Amazon CloudHSM 密钥关联的 KMS 密钥。

> **备注**
>
> 以下过程使用 Amazon CloudHSM 客户端 SDK 5 命令行工具 CloudHSM CLI。CloudHSM CLI 将 key-handle 替换为 key-reference。
> 2025 年 1 月 1 日，Amazon CloudHSM 将终止对客户端 SDK 3 命令行工具、CloudHSM 管理实用程序（CMU）和密钥管理实用程序（KMU）的支持。有关客户端 SDK 3 命令行工具和客户端 SDK 5 命令行工具之间区别的更多信息，请参阅《Amazon CloudHSM 用户指南》中的从客户端 SDK 3 CMU 和 KMU 迁移到客户端 SDK 5 CloudHSM CLI。

要运行这些过程，您需要临时断开 Amazon CloudHSM 密钥存储，以便能以 kmsuser CU 身份登录。

> **Note**
>
> 虽然自定义密钥存储已断开连接，但在自定义密钥存储中创建 KMS 密钥或在加密操作中使用现有 KMS 密钥的所有尝试都将失败。此操作可以阻止用户存储和访问敏感数据。

主题

识别与密钥参考关联的 KMS 密钥

以下过程演示如何在 CloudHSM CLI 中使用 key list 命令和 key-reference 属性筛选器，在集群中查找用作 Amazon CloudHSM 密钥存储中特定 KMS 密钥的密钥材料的密钥。

1. 断开 Amazon CloudHSM 密钥存储（如果尚未断开），然后以 kmsuser 身份登录，如 如何断开和登录 中所述。
2. 使用 CloudHSM CLI 中的 key list 命令，按 key-reference 属性进行筛选。指定 verbose 参数以包含匹配密钥的所有属性和密钥信息。如果不指定 verbose 参数，则密钥列表操作将仅返回匹配密钥的密钥参考和标签属性。

   在运行此命令之前，请将示例 key-reference 替换为您账户中的有效密钥 ID。

```
aws-cloudhsm > key list --filter attr.key-reference="0x0000000000120034" --verbose
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x0000000000120034",
        "key-info": {
          "key-owners": [
            {
              "username": "kmsuser",
              "key-coverage": "full"
            }
          ],
          "shared-users": [],
          "cluster-coverage": "full"
        },
        "attributes": {
          "key-type": "aes",
          "label": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
          "id": "0xbacking-key-id",
          "check-value": "0x29bbd1",
```

```
          "class": "my_test_key",
          "encrypt": true,
          "decrypt": true,
          "token": true,
          "always-sensitive": true,
          "derive": false,
          "destroyable": true,
          "extractable": false,
          "local": true,
          "modifiable": true,
          "never-extractable": false,
          "private": true,
          "sensitive": true,
          "sign": false,
          "trusted": false,
          "unwrap": true,
          "verify": false,
          "wrap": true,
          "wrap-with-trusted": false,
          "key-length-bytes": 32
        }
      }
    ],
    "total_key_count": 1,
    "returned_key_count": 1
  }
}
```

3. 注销并重新连接 Amazon CloudHSM 密钥存储，如 如何注销并重新连接 中所述。

识别与配对密钥 ID 关联的 KMS 密钥

使用 Amazon CloudHSM 密钥存储中的 KMS 密钥进行加密操作的所有 CloudTrail 日志条目都包含带有 `customKeyStoreId` 和 `backingKeyId` 的 `additionalEventData` 字段。`backingKeyId` 字段中返回的值与 CloudHSM 密钥 `id` 属性关联。您可以按 `id` 属性筛选密钥列表操作，识别与特定 `backingKeyId` 关联的 KMS 密钥。

1. 断开 Amazon CloudHSM 密钥存储（如果尚未断开），然后以 `kmsuser` 身份登录，如 如何断开和登录 中所述。
2. 在 CloudHSM CLI 中使用 key list 命令和属性筛选器，在集群中查找用作 Amazon CloudHSM 密钥存储中特定 KMS 密钥的密钥材料的密钥。

以下示例演示如何按 id 属性进行筛选。Amazon CloudHSM 将 id 值识别为十六进制值。要按 id 属性筛选密钥列表操作，必须先将您在 CloudTrail 日志条目中识别的 backingKeyId 值转换为 Amazon CloudHSM 可识别的格式。

a. 使用以下 Linux 命令将 backingKeyId 转换为十六进制表示形式。

```
echo backingKeyId | tr -d '\n' |  xxd -p
```

以下示例演示了如何将 backingKeyId 字节数组转换为十六进制表示形式。

```
echo 5890723622dc15f699aa9ab2387a9f744b2b884c18b2186ee8ada4f556a2eb9d | tr -d
 '\n' |  xxd -p
353839303732333632326463313566363939616139616232333837613966373434623262383834633138623
```

b. 在 backingKeyId 的十六进制表示形式前面加上 0x。

```
0x353839303732333632326463313566363939616139616232333837613966373434623262383834633138623
```

c. 使用转换后的 backingKeyId 值按 id 属性进行筛选。指定 verbose 参数以包含匹配密钥的所有属性和密钥信息。如果不指定 verbose 参数，则密钥列表操作将仅返回匹配密钥的密钥参考和标签属性。

```
aws-cloudhsm > key list --filter
 attr.id="0x353839303732333632326463313566363939616139616232333837613966373434623262383
 --verbose
{
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x0000000000120034",
        "key-info": {
          "key-owners": [
            {
              "username": "kmsuser",
              "key-coverage": "full"
            }
          ],
          "shared-users": [],
          "cluster-coverage": "full"
        },
```

```
        "attributes": {
          "key-type": "aes",
          "label": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
          "id":
 "0x353839303732333632326463313566363939616139616232333837613966373434623262383834633133
          "check-value": "0x29bbd1",
          "class": "my_test_key",
          "encrypt": true,
          "decrypt": true,
          "token": true,
          "always-sensitive": true,
          "derive": false,
          "destroyable": true,
          "extractable": false,
          "local": true,
          "modifiable": true,
          "never-extractable": false,
          "private": true,
          "sensitive": true,
          "sign": false,
          "trusted": false,
          "unwrap": true,
          "verify": false,
          "wrap": true,
          "wrap-with-trusted": false,
          "key-length-bytes": 32
        }
      }
    ],
    "total_key_count": 1,
    "returned_key_count": 1
  }
}
```

3. 注销并重新连接 Amazon CloudHSM 密钥存储，如 如何注销并重新连接 中所述。

查找 KMS 密钥的 Amazon CloudHSM 密钥

您可以在 Amazon CloudHSM 密钥存储中使用 KMS 密钥的 KMS 密钥 ID 来标识 Amazon CloudHSM 集群中用作其密钥材料的密钥。

当 Amazon KMS 在 Amazon CloudHSM 集群中为 KMS 密钥创建密钥材料时，它会在密钥标签中写入 KMS 密钥的 Amazon Resource Name (ARN)。除非您已更改标签值，否则可在 CloudHSM CLI 中使用 key list 命令来查找 KMS 密钥的密钥材料的密钥资源和 ID。

使用 Amazon CloudHSM 密钥存储中的 KMS 密钥进行加密操作的所有 CloudTrail 日志条目都包含带有 `customKeyStoreId` 和 `backingKeyId` 的 `additionalEventData` 字段。`backingKeyId` 字段中返回的值是 `id` Amazon CloudHSM 密钥属性。您可以按 KMS 密钥 ARN 筛选 key list Amazon CloudHSM CLI 操作，识别与特定 KMS 密钥关联的 CloudHSM 密钥 `id` 属性。

要运行此过程，您需要临时断开 Amazon CloudHSM 密钥存储，以便能以 `kmsuser` CU 身份登录。

> **备注**
>
> 以下过程使用 Amazon CloudHSM 客户端 SDK 5 命令行工具 CloudHSM CLI。CloudHSM CLI 将 `key-handle` 替换为 `key-reference`。
> 2025 年 1 月 1 日，Amazon CloudHSM 将终止对客户端 SDK 3 命令行工具、CloudHSM 管理实用程序（CMU）和密钥管理实用程序（KMU）的支持。有关客户端 SDK 3 命令行工具和客户端 SDK 5 命令行工具之间区别的更多信息，请参阅《Amazon CloudHSM 用户指南》中的从客户端 SDK 3 CMU 和 KMU 迁移到客户端 SDK 5 CloudHSM CLI。

1. 断开 Amazon CloudHSM 密钥存储（如果尚未断开），然后以 `kmsuser` 身份登录，如 如何断开和登录 中所述。

   > **Note**
   >
   > 虽然自定义密钥存储已断开连接，但在自定义密钥存储中创建 KMS 密钥或在加密操作中使用现有 KMS 密钥的所有尝试都将失败。此操作可以阻止用户存储和访问敏感数据。

2. 在 CloudHSM CLI 中使用 key list 命令并按 `label` 进行筛选，查找 Amazon CloudHSM 集群中特定密钥的 KMS 密钥。指定 `verbose` 参数以包含匹配密钥的所有属性和密钥信息。如果不指定 `verbose` 参数，则密钥列表操作将仅返回匹配密钥的密钥参考和标签属性。

   以下示例演示如何按存储 KMS 密钥 ARN 的 `label` 属性进行筛选。在运行此命令之前，请将示例 KMS 密钥 ARN 替换为您账户中的有效的 KMS 密钥 ARN。

   ```
   aws-cloudhsm > key list --filter attr.label="arn:aws:kms:us-
   west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab" --verbose
   {
   ```

```
  "error_code": 0,
  "data": {
    "matched_keys": [
      {
        "key-reference": "0x0000000000120034",
        "key-info": {
          "key-owners": [
            {
              "username": "kmsuser",
              "key-coverage": "full"
            }
          ],
          "shared-users": [],
          "cluster-coverage": "full"
        },
        "attributes": {
          "key-type": "aes",
          "label": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
          "id": "0xbacking-key-id",
          "check-value": "0x29bbd1",
          "class": "my_test_key",
          "encrypt": true,
          "decrypt": true,
          "token": true,
          "always-sensitive": true,
          "derive": false,
          "destroyable": true,
          "extractable": false,
          "local": true,
          "modifiable": true,
          "never-extractable": false,
          "private": true,
          "sensitive": true,
          "sign": false,
          "trusted": false,
          "unwrap": true,
          "verify": false,
          "wrap": true,
          "wrap-with-trusted": false,
          "key-length-bytes": 32
        }
      }
    ],
```

```
    "total_key_count": 1,
    "returned_key_count": 1
  }
}
```

3. 注销并重新连接 Amazon CloudHSM 密钥存储，如 如何注销并重新连接 中所述。

启用和禁用密钥

您可以禁用和重新启用客户托管密钥。KMS 密钥在创建完成后默认处于启用状态。如果禁用 KMS 密钥，则在您重新启用它之前，它不能用于任何加密操作。

因为它是临时的且容易撤消，因此，禁用 KMS 密钥是删除 KMS 密钥的安全替代方法，此操作具有破坏性且不可撤销。如果您正在考虑删除 KMS 密钥，请先将其禁用并将 CloudWatch 告警或者类似机制设置为永远不需要使用此密钥来解密加密数据。

当您禁用 KMS 密钥时，它会立即变为不可用（视最终一致性而定）。不过，在再次使用 KMS 密钥（例如解密数据密钥）之前，使用受 KMS 密钥保护的数据密钥加密的资源不会受到影响。此问题会影响 Amazon Web Services 服务，因为许多服务使用数据密钥来保护您的资源。有关更多信息，请参阅不可用的 KMS 密钥如何影响数据密钥。

您无法启用或禁用 Amazon 托管式密钥或 Amazon 拥有的密钥。Amazon 托管式密钥 永久处于已启用状态，以供使用 Amazon KMS 的服务使用。Amazon 拥有的密钥 仅由拥有密钥的服务管理。

> **Note**
>
> Amazon KMS 不会轮换已禁用的客户托管密钥的密钥材料。有关更多信息，请参阅 密钥轮换的工作原理。

使用 Amazon KMS 控制台

您可以使用 Amazon KMS 控制台以启用和禁用客户托管密钥。

1. 登录到 Amazon Web Services 管理控制台，然后通过以下网址打开 Amazon Key Management Service（Amazon KMS）控制台：https://console.aws.amazon.com/kms。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择客户托管密钥。
4. 选中要启用或禁用的 KMS 密钥对应的复选框。
5. 要启用 KMS 密钥，请依次选择 Key actions（密钥操作）、 Enable（启用）。要禁用 KMS 密钥，请依次选择 Key actions（密钥操作）、Disable（禁用）。

使用 Amazon KMS API

EnableKey 操作将启用已禁用的 Amazon KMS key。这些示例使用 Amazon Command Line Interface (Amazon CLI)，但您可以使用任何受支持的编程语言。key-id 参数是必需的。

该操作不返回任何输出。要查看密钥状态，请使用 DescribeKey 操作。

```
$ aws kms enable-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

DisableKey 操作会禁用已启用的 KMS 密钥。key-id 参数是必需的。

```
$ aws kms disable-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

该操作不返回任何输出。要查看密钥状态，请使用 DescribeKey 操作，并查看 Enabled 字段。

```
$ aws kms describe-key --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "KeyMetadata": {
        "Origin": "AWS_KMS",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Description": "",
        "KeyManager": "CUSTOMER",
        "MultiRegion": false,
        "Enabled": false,
        "KeyState": "Disabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "CreationDate": 1502910355.475,
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333"
        "KeySpec": "SYMMETRIC_DEFAULT",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

旋转 Amazon KMS keys

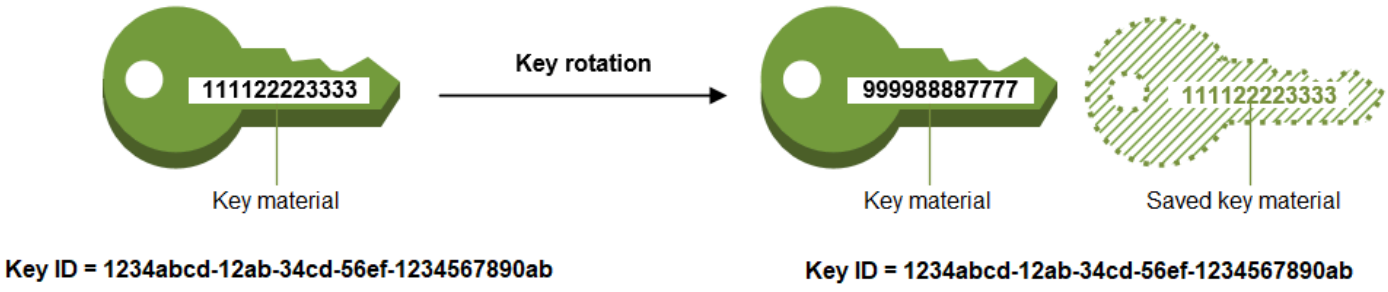
要为客户托管的密钥创建新的加密材料，您可以创建新的 KMS 密钥，然后更改您的应用程序或别名来使用新的 KMS 密钥。或者，您可以通过启用自动密钥轮换或执行按需轮换，来轮换与现有 KMS 密钥关联的密钥材料。

默认情况下，当您为 KMS 密钥启用自动密钥轮换时，每年会为 KMS 密钥 Amazon KMS 生成新的加密材料。您还可以指定自定义，rotation-period以定义启用自动密钥轮换以轮换密钥材料后的天数，以及此后每次自动轮换之间的天数。 Amazon KMS 如果您需要立即启动密钥材料轮换，则无论是否启用自动密钥轮换，都可以执行按需轮换。按需轮换不会更改现有的自动轮换计划。

您可以在 Amazon CloudWatch 和 Amazon Key Management Service 控制台中跟踪 KMS 密钥的密钥材料的轮换情况。 Amazon CloudTrail您还可以使用GetKeyRotationStatus操作来验证 KMS 密钥是否启用了自动轮换，并识别任何正在进行的按需轮换。您可以使用ListKeyRotations操作来查看已完成旋转的详细信息。

密钥轮换只会更改当前密钥材料，即加密操作中使用的密钥。当您使用轮换的 KMS 密钥解密密文时，会 Amazon KMS 使用用于加密密文的密钥材料。无法为解密操作选择特定的密钥材料， Amazon KMS 会自动选择正确的密钥材料。由于使用适当的密钥材料进行 Amazon KMS 透明解密，因此您可以安全地在应用程序中使用轮换的 KMS 密钥，而 Amazon Web Services 服务 无需更改代码。

不管密钥材料有没有变更或变更了多少次，该 KMS 密钥仍是相同的逻辑资源。KMS 密钥的属性不会发生变化，如下图所示。



您可能决定创建新的 KMS 密钥来替代原有的 KMS 密钥。这跟轮换现有 KMS 密钥的密钥材料具有相同效果，因此这通常被视为手动轮换密钥。对于不符合自动或按需密钥轮换要求的 KMS 密钥（包括非对称 KMS 密钥、HMAC KMS 密钥、自定义密钥存储中的 KMS 密钥以及具有导入的密钥材料的多区域 KMS 密钥），建议选择手动轮换。

> **Note**
>
> 密钥轮换不会影响 KMS 密钥所保护的数据。不会轮换 KMS 密钥生成的数据密钥，也不会对任何受 KMS 密钥保护的数据重新加密。密钥轮换不会减轻数据密钥泄露的影响。

密钥轮换和定价

Amazon KMS 对为您的 KMS 密钥维护的密钥材料的第一次和第二次轮换收取月费。此次涨价以第二次轮换为上限，后续任何轮换均不收取费用。有关详细信息，请参阅 Amazon Key Management Service 定价。

> **Note**
>
> 您可以使用《Amazon Cost Explorer Service》https://docs.amazonaws.cn/cost-management/latest/userguide/ce-what-is.html来查看您的密钥存储费用明细。例如，您可以通过将使用类型指定为 `$REGION-KMS-Keys` 并按 API 操作对数据进行分组来筛选视图，以查看按当前和轮换 KMS 密钥计费的密钥的总费用。
> 您可能仍会看到历史日期的旧 `Unknown` API 操作的实例。

密钥轮换和配额

在计算密钥资源配额时，无论轮换密钥材料版本的数量如何，每个 KMS 密钥均算作一个密钥。

有关密钥材料和轮换的详细信息，请参阅《Amazon Key Management Service 加密详细信息》。

主题

为什么要轮换 KMS 密钥？

加密最佳实践不鼓励大量重复使用直接加密数据的密钥，例如生成的数据密钥。 Amazon KMS 当 256 位数据密钥加密数百万条消息时，它们可能会耗尽并开始生成带有细微模式的加密文字，聪明的操作者可以利用这些密文来发现密钥中的位。建议数据密钥仅使用一次或少数几次，以减少这种密钥耗尽问题。

但是，KMS 密钥最常用作“包装密钥”，也称为“密钥加密密钥”。包装密钥不是加密数据，而是对加密数据的数据密钥进行加密。因此，其使用频率远低于数据密钥，并且几乎从未被重复使用到足以出现密钥耗尽的风险。

尽管耗尽风险非常低，但由于业务或合同规则或政府法规，您可能需要轮换 KMS 密钥。被迫轮换 KMS 密钥时，我们建议您在支持自动密钥轮换时使用自动轮换，在不支持自动轮换时使用按需轮换，在既不支持自动密钥轮换也不支持按需密钥轮换时使用手动密钥轮换。

您可以考虑执行按需轮换，演示密钥材料轮换功能或验证自动化脚本。我们建议将按需轮换用于计划外轮换场景，尽可能使用自动密钥轮换并自定义轮换周期。

密钥轮换的工作原理

Amazon KMS 密钥轮换设计为透明且易于使用。 Amazon KMS 仅支持客户托管密钥的可选自动和按需密钥轮换。

自动密钥轮换

Amazon KMS 在轮换周期定义的下一个轮换日期自动轮换 KMS 密钥。您无需记住或计划更新。

只有 Amazon KMS 生成密钥材料的对称加密 KMS 密钥才支持自动密钥轮换（`AWS_KMS`来源）。

对于客户托管的 KMS 密钥，自动轮换是可选的。 Amazon KMS 始终每年轮换 Amazon 托管 KMS 密钥的密钥材料。 Amazon 拥有的 KMS 密钥的轮换由拥有 Amazon Web Services 服务 该密钥的人管理。

按需轮换

无论是否启用了自动密钥轮换，都会立即开始轮换与 KMS 密钥关联的密钥材料。

对称加密 KMS 密钥支持按需密钥轮换，其密钥材料 Amazon KMS 可生成（`AWS_KMS`来源），对称加密 KMS 密钥支持使用导入的密钥材料（`EXTERNAL`来源）。

手动轮换

以下类型的 KMS 密钥不支持自动密钥轮换或按需密钥轮换，但您可以手动轮换这些 KMS 密钥。

- 非对称 KMS 密钥
- HMAC KMS 密钥
- 自定义密钥存储中的 KMS 密钥

管理密钥材料

Amazon KMS 即使禁用了密钥轮换，也会保留带有AWS_KMS来源的 KMS 密钥的所有密钥材料。Amazon KMS 只有在删除 KMS 密钥时才会删除密钥材料。

您负责管理来源为 EXTERNAL 的对称加密密钥的密钥材料。您可以使用DeleteImportedKeyMaterial操作删除任何密钥材料，也可以在导入材料时设置过期时间。一旦 KMS 密钥的任何材料过期或被删除，该 KMS 密钥就会变为无法使用。

使用密钥材料

当您使用轮换的 KMS 密钥加密数据时， Amazon KMS 使用当前的密钥材料。当您使用轮换 KMS 密钥解密密文时， Amazon KMS 会使用与加密时所用密钥材料相同的版本。您无法为解密操作选择特定版本的密钥材料，而是 Amazon KMS 会自动选择正确的版本。

轮换周期

轮换期限定义了在您启用自动密钥轮换后， Amazon KMS 将轮换密钥材料的天数，以及此后每次自动密钥轮换之间的天数。如果您在启用自动密钥轮换时未指定 RotationPeriodInDays 的值，则默认值为 365 天。

您可以使用 k ms: RotationPeriodInDays 条件键进一步限制委托人可以在参数中RotationPeriodInDays指定的值。

轮换日期

轮换日期是指因自动（定期）轮换或按需密钥轮换而更新 KMS 密钥的当前密钥材料的日期。

轮换日期

Amazon KMS 在轮换周期定义的轮换日期自动轮换 KMS 密钥。默认轮换周期为 365 天。

客户托管密钥

由于客户托管密钥的自动密钥轮换是可选的，并且可以随时启用和禁用，因此轮换日期取决于最近启用轮换的日期。如果您修改先前启用了自动密钥轮换的密钥的轮换周期，则日期可能会更改。在密钥的生命周期内，轮换日期可能会更改很多次。

例如，如果您在 2022 年 1 月 1 日创建客户托管密钥，并在 2022 年 3 月 15 日启用自动密钥轮换，且默认轮换周期为 365 天，则 Amazon KMS 会在 2023 年 3 月 15 日、2024 年 3 月 15 日以及之后每 365 天轮换密钥材料。

以下示例假设启用了自动密钥轮换，且默认轮换周期为 365 天。这些示例演示了可能影响密钥轮换周期的特殊情况。

- 禁用密钥轮换 - 如果您在任何时候禁用自动密钥轮换，KMS 密钥将继续使用禁用轮换时使用的密钥材料版本。如果您再次启用自动密钥轮换，则 Amazon KMS 会根据新的启用轮换的日期轮换密钥材料。
- 已禁用 KMS 密钥-当 KMS 密钥处于禁用状态时， Amazon KMS 不会对其进行轮换。但是，密钥轮换状态不会发生改变，并且在 KMS 密钥处于禁用状态时不能对其进行更改。重新启用 KMS 密钥后，如果密钥材料已超过其上次计划轮换日期，则 Amazon KMS 会立即轮换。如果密钥材料没有错过上次预定的轮换日期，则 Amazon KMS 恢复原来的密钥轮换计划。
- 待删除的 KMS 密钥-当 KMS 密钥处于待删除状态时， Amazon KMS 不会对其进行轮换。密钥轮换状态设为 `false`，处于待删除状态时不能更改。如果删除被取消，将恢复之前的密钥轮换状态。如果密钥材料已超过其上次预定的轮换日期，则立即将其 Amazon KMS 轮换。如果密钥材料没有错过上次预定的轮换日期，则 Amazon KMS 恢复原来的密钥轮换计划。

Amazon 托管式密钥

Amazon KMS Amazon 托管式密钥 每年自动轮换（大约 365 天）。您无法启用或禁用 Amazon 托管式密钥 的密钥轮换。

的密钥材料在创建日期一年后首次轮换，此后每年（自上次轮换后大约 365 天）进行轮换。Amazon 托管式密钥

> **Note**
>
> 2022年5月，将轮换时间表 Amazon 托管式密钥 从每三年（约1,095天） Amazon KMS 改为每年（约365天）。

Amazon 拥有的密钥

您无法启用或禁用 Amazon 拥有的密钥的密钥轮换。的密钥轮换策略 Amazon 拥有的密钥 由创建和管理密钥的 Amazon 服务决定。有关详细信息，请参阅服务的用户指南或开发人员指南中的静态加密主题。

轮换多区域密钥

对于来源为 `AWS_KMS` 的对称加密多区域密钥，您可以启用和禁用密钥材料自动轮换并执行按需轮换。密钥轮换是多区域密钥的共享属性。

仅在主密钥上启用和禁用自动密钥轮换。您仅对主密钥启动按需轮换。

- Amazon KMS 同步多区域密钥时，它会将密钥轮换属性设置从主键复制到其所有相关的副本密钥。
- Amazon KMS 轮换密钥材料时，它会为主键创建新的密钥材料，然后将新的密钥材料跨区域边界复制到所有相关的副本密钥中。密钥材料永远不会处于 Amazon KMS 未加密状态。此步骤经过精心控制，以确保在加密操作中使用任何密钥之前密钥材料完全同步。
- Amazon KMS 在主密钥及其每个副本密钥中都提供该密钥材料之前，不会使用新的密钥材料对任何数据进行加密。
- 复制已轮换的主密钥时，新的副本密钥具有相关多区域密钥的当前密钥材料和所有先前版本的密钥材料。

此模式可确保相关的多区域密钥完全可互操作。任何多区域密钥都可以解密由相关多区域密钥加密的任何密文，即使密文在创建密钥之前已加密。

Amazon 服务

您可以对用于 Amazon 服务中服务器端加密的客户托管密钥启用自动密钥轮换。年度轮换是透明的，并且与 Amazon 服务兼容。

监控密钥轮换

Amazon KMS 轮换Amazon 托管式密钥或客户托管密钥的密钥材料时，它会向 Amazon EventBridge 写一个`KMS CMK Rotation`事件，将一个RotateKey 事件写入您的 Amazon CloudTrail 日志。您可以使用这些记录验证 KMS 密钥是否已轮换。

您可以使用 Amazon Key Management Service 控制台查看 KMS 密钥的剩余按需轮换次数以及 KMS 密钥所有已完成的密钥材料轮换的列表。

您可以使用ListKeyRotations操作来查看已完成旋转的详细信息。

最终一致性

密钥轮换会受到与其他 Amazon KMS 管理操作相同的最终一致性影响。新的密钥材料在整个 Amazon KMS中可用之前可能会有一些延迟。但是，轮换密钥材料不会导致加密操作中断或延迟。当前的密钥材料用于加密操作，直到新的密钥材料在整个 Amazon KMS中可用为止。当自动轮换多区域密钥的密钥材料时，将 Amazon KMS 使用当前密钥材料，直到新的密钥材料在所有带有相关多区域密钥的区域中都可用。

启用自动密钥轮换

默认情况下，当您为 KMS 密钥启用自动密钥轮换时，每年会为 KMS 密钥 Amazon KMS 生成新的加密材料。您还可以指定自定义，rotation-period以定义启用自动密钥轮换以轮换密钥材料后的天数，以及此后每次自动轮换之间的天数。 Amazon KMS

自动密钥轮换具有以下优点：

- KMS 密钥的属性，包括其密钥 ID、密钥 ARN、区域、策略和权限，在密钥轮换时不会发生改变。
- 您无需更改引用该 KMS 密钥的密钥 ID 或密钥 ARN 的应用程序或别名。
- 轮换密钥材料不会影响 Amazon Web Services 服务中任何 KMS 密钥的使用。
- 启用密钥轮换后，将在 Amazon KMS 轮换周期定义的下一个轮换日期自动轮换 KMS 密钥。您无需记住或计划更新。

您可以在 Amazon KMS 控制台中或使用EnableKeyRotation操作启用自动密钥轮换。要启用自动密钥轮换，您需要 `kms:EnableKeyRotation` 权限。有关 Amazon KMS 权限的更多信息，请参阅权限参考。

使用控制 Amazon KMS 台

1. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/km s 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择客户托管密钥。（您无法启用或禁用 Amazon 托管式密钥的轮换。它们每年自动轮换一次。）
4. 选择 KMS 密钥的别名和密钥 ID。
5. 选择 Key rotation (密钥轮换) 选项卡。

   密钥轮换选项卡仅出现在对称加密 KMS 密钥的详细信息页面上，其中包含 Amazon KMS 生成的密钥材料（来源是 AWS_KMS），包括多区域对称加密 KMS 密钥。

   您不能自动轮换非对称 KMS 密钥、HMAC KMS 密钥、具有导入的密钥材料的 KMS 密钥或自定义密钥存储中的 KMS 密钥。但是，您可以手动轮换它们。
6. 在自动密钥轮换部分，选择编辑。
7. 对于密钥轮换，请选择启用。

> **Note**
>
> 如果 KMS 密钥已禁用或待删除，则 Amazon KMS 不会轮换密钥材料，也无法更新自动密钥轮换状态或轮换周期。启用 KMS 密钥或取消删除以更新自动密钥轮换配置。有关详细信息，请参阅 密钥轮换的工作原理 和 Amazon KMS 密钥的密钥状态。

8. （可选）键入介于 90 到 2560 天之间的轮换周期。默认值为 365 天。如果您未指定自定义轮换周期，则 Amazon KMS 将每年轮换密钥材料。

   您可以使用 k ms: RotationPeriodInDays 条件键来限制委托人可以为轮换周期指定的值。

9. 选择保存。

使用 Amazon KMS API

您可以使用 Amazon Key Management Service （ Amazon KMS ）API 启用自动密钥轮换，并查看任何客户托管密钥的当前轮换状态。这些示例使用 Amazon Command Line Interface (Amazon CLI)，但您可以使用任何受支持的编程语言。

该EnableKeyRotation操作可为指定的 KMS 密钥启用自动密钥轮换。要标识此操作中的 KMS 密钥，请使用其密钥 ID 或密钥 ARN。在默认情况下，客户托管密钥的密钥轮换处于禁用状态。

您可以使用 kms:RotationPeriodInDays条件键来限制委托人可以为`EnableKeyRotation`请求的`RotationPeriodInDays`参数指定的值。

以下示例在指定的对称加密 KMS 密钥上启用密钥轮换，轮换周期为 180 天，并使用该GetKeyRotationStatus操作来查看结果。

```
$ aws kms enable-key-rotation \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --rotation-period-in-days 180

$ aws kms get-key-rotation-status --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyRotationEnabled": true,
    "RotationPeriodInDays": 180,
    "NextRotationDate": "2024-02-14T18:14:33.587000+00:00"
}
```

禁用自动密钥轮换

对客户托管密钥启用自动密钥轮换后，您可以随时选择禁用自动轮换。

如果您禁用自动密钥轮换，KMS 密钥将继续使用禁用轮换时使用的密钥材料版本。如果您再次启用自动密钥轮换，则 Amazon KMS 会根据新的启用轮换的日期轮换密钥材料。

禁用自动轮换不会影响您执行按需轮换的能力，也不会取消任何正在进行的按需轮换。

您可以在 Amazon KMS 控制台中或使用DisableKeyRotation操作来禁用自动密钥轮换。要禁用自动密钥轮换，您需要 `kms:DisableKeyRotation` 权限。有关 Amazon KMS 权限的更多信息，请参阅权限参考。

使用控制 Amazon KMS 台

1. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/km s 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择客户托管密钥。（您无法启用或禁用 Amazon 托管式密钥的轮换。它们每年自动轮换一次。）
4. 选择 KMS 密钥的别名和密钥 ID。
5. 选择 Key rotation (密钥轮换) 选项卡。

   密钥轮换选项卡仅出现在对称加密 KMS 密钥的详细信息页面上，其中包含 Amazon KMS 生成的密钥材料（来源是 AWS_KMS），包括多区域对称加密 KMS 密钥。

   您不能自动轮换非对称 KMS 密钥、HMAC KMS 密钥、具有导入的密钥材料的 KMS 密钥或自定义密钥存储中的 KMS 密钥。但是，您可以手动轮换它们。
6. 在自动密钥轮换部分，选择编辑。
7. 对于密钥轮换，请选择禁用。

   > **Note**
   >
   > 如果 KMS 密钥已禁用或待删除，则 Amazon KMS 不会轮换密钥材料，也无法更新自动密钥轮换状态或轮换周期。启用 KMS 密钥或取消删除以更新自动密钥轮换配置。有关详细信息，请参阅 密钥轮换的工作原理 和 Amazon KMS 密钥的密钥状态。
8. 选择保存。

使用 Amazon KMS API

您可以使用 Amazon Key Management Service （ Amazon KMS ）API 禁用自动密钥轮换，并查看任何客户托管密钥的当前轮换状态。此示例使用 Amazon Command Line Interface （ Amazon CLI ），但您可以使用任何受支持的编程语言。

该DisableKeyRotation操作将禁用自动密钥轮换。要标识此操作中的 KMS 密钥，请使用其密钥 ID 或密钥 ARN。在默认情况下，客户托管密钥的密钥轮换处于禁用状态。

以下示例在指定的对称加密 KMS 密钥上禁用自动密钥轮换，并使用该GetKeyRotationStatus操作来查看结果。

```
$ aws kms disable-key-rotation --key-id 1234abcd-12ab-34cd-56ef-1234567890ab

$ aws kms get-key-rotation-status --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyRotationEnabled": false
}
```

执行按需密钥轮换

无论是否启用了自动密钥轮换，您都可以对客户托管 KMS 密钥中的密钥材料执行按需轮换。禁用自动轮换 (DisableKeyRotation) 不会影响您执行按需轮换的能力，也不会取消任何正在进行的按需轮换。按需轮换不会更改现有的自动轮换计划。例如，假设一个 KMS 密钥启用了自动密钥轮换，轮换周期为 730 天。如果密钥计划在 2024 年 4 月 14 日自动轮换，而您在 2024 年 4 月 10 日执行按需轮换，则密钥将按计划在 2024 年 4 月 14 日自动轮换，此后每 730 天自动轮换一次。

您最多可以对每个 KMS 密钥执行按需轮换 10 次。您可以使用 Amazon KMS 控制台查看 KMS 密钥的剩余按需轮换次数。

只有对称加密 KMS 密钥支持按需密钥轮换。您无法按需轮换自定义密钥存储库中的非对称 KMS 密钥、HMAC KMS 密钥或 KMS 密钥。要按需轮换一组相关的多区域密钥，请对主密钥调用按需轮换。

拥有`kms:RotateKeyOnDemand`和`kms:GetKeyRotationStatus`权限的授权用户可以使用 Amazon KMS 控制台和 Amazon KMS API 启动按需密钥轮换并查看密钥轮换状态。ListKeyRotations用于查看 KMS 密钥的已完成轮换。

主题

- 启动按需密钥轮换（控制台）
- 启动按需密钥轮换 (Amazon KMS API)

启动按需密钥轮换（控制台）

1. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/kms 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择客户托管密钥。（您无法对 Amazon 托管式密钥执行按需轮换。其将每年自动轮换一次。）
4. 选择 KMS 密钥的别名和密钥 ID。
5. 选择按密钥材料和轮换选项卡。

   密钥材料和轮换选项卡仅在支持自动轮换或按需轮换的对称加密 KMS 密钥的详细信息页面上显示。这包括 Amazon KMS 生成密钥材料的 KMS 密钥（AWS_KMS来源）和带有导入密钥材料的 KMS 密钥（外部来源）

   您无法按需轮换自定义密钥存储库中的非对称 KMS 密钥、HMAC KMS 密钥或 KMS 密钥。但是，您可以手动轮换它们。
6. 选择立即轮换。对于包含已导入密钥材料的对称加密密钥，只有在您之前导入了新的密钥材料且其处于 “待轮换” 状态时，“立即轮换” 选项才可用。

   > **Note**
   >
   > 对于多区域密钥，只能轮换主区域密钥。

7. 阅读并考虑有关密钥剩余按需轮换次数的警告和信息。您还将看到在轮换后将变为当前的密钥材料的 ID、描述和过期时间等信息。如果您决定不继续按需轮换，请选择取消。
8. 选择轮换密钥以确认按需轮换。

   > **Note**
   >
   > 按需轮换会受到与其他 Amazon KMS 管理操作相同的最终一致性影响。新的密钥材料在整个 Amazon KMS中可用之前可能会有一些延迟。控制台顶部的横幅会在按需轮换完成后通知您。

启动按需密钥轮换 (Amazon KMS API)

您可以使用 Amazon Key Management Service （ Amazon KMS ）API 启动按需密钥轮换，并查看任何客户托管密钥的当前轮换状态。此示例使用 Amazon Command Line Interface （ Amazon CLI ），但您可以使用任何受支持的编程语言。

该RotateKeyOnDemand操作会立即启动指定的 KMS 密钥的按需密钥轮换。要在这些操作中标识 KMS 密钥，请使用其密钥 ID 或密钥 ARN。

以下示例在指定的对称加密 KMS 密钥上启动按需密钥轮换，并使用该GetKeyRotationStatus操作验证按需轮换是否正在进行中。`kms:GetKeyRotationStatus` 响应中的 `OnDemandRotationStartDate` 标识了正在进行的按需轮换启动的日期和时间。在此例中，该 KMS 密钥还启用了自动轮换，轮换周期为 365 天。

```
$ aws kms rotate-key-on-demand --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
}

$ aws kms get-key-rotation-status --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyRotationEnabled": true,
    "NextRotationDate": "2024-03-14T18:14:33.587000+00:00",
    "OnDemandRotationStartDate": "2024-02-24T18:44:48.587000+00:00"
    "RotationPeriodInDays": 365
}
```

如果 KMS 密钥不支持自动轮换或未启用自动轮换，则 `kms:GetKeyRotationStatus` 响应的字段数将比下例所示的要少：

```
$ aws kms rotate-key-on-demand --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
}

$ aws kms get-key-rotation-status --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyRotationEnabled": false,
    "OnDemandRotationStartDate": "2024-02-24T18:44:48.587000+00:00"
```

```
}
```

列出轮换和密钥材料

支持自动或按需密钥轮换的 KMS 密钥可以关联多个密钥材料。这些密钥具有一个初始密钥材料，以及每次自动或按需轮换的附加密钥材料。

拥有`kms:ListKeyRotations`权限的授权用户可以使用 Amazon KMS 控制台和 ListKeyRotationsAPI 列出与 KMS 密钥关联的所有密钥材料，包括已完成的自动和按需轮换的密钥材料。

主题

- 列出轮换和密钥材料（控制台）
- 列出轮换和关键材料 (Amazon KMS API)

列出轮换和密钥材料（控制台）

1. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/km s 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择客户托管密钥。
4. 选择 KMS 密钥的别名和密钥 ID。
5. 选择按密钥材料和轮换选项卡。

    - 密钥材料和轮换选项卡仅在支持自动轮换或按需轮换的对称加密 KMS 密钥的详细信息页面上显示。这包括带有已 Amazon KMS 生成密钥材料（`AWS_KMS`来源）的 KMS 密钥和带有导入密钥材料（`EXTERNAL`来源）的 KMS 密钥。
    - 密钥材料和轮换选项卡中的密钥材料表会列出关联到 KMS 密钥的所有密钥材料。每个密钥材料的对应条目会显示 Amazon KMS为其分配的唯一标识符、轮换日期和密钥材料状态。轮换日期用于确定密钥材料在自动或按需密钥轮换后何时变为当前。第一个或 `Pending rotation` 密钥材料没有相关的轮换日期。密钥材料状态决定了密钥材料的 Amazon KMS 使用方式。当前密钥材料既用于加密，也用于解密。非当前密钥材料仅用于解密。密钥材料状态为 `Pending rotation` 表示该密钥材料已进入轮换暂存阶段。在按需密钥轮换使其成为当前密钥材料之前，此密钥材料不会用于任何密码操作。所显示有关密钥材料的其他信息取决于 KMS 密钥的类型。

- 对于来源为 AWS_KMS 的对称加密 KMS 密钥，每行还会显示轮换类型（On-demand 或 Automatic）。
- 对称加密带有导入密钥材料（EXTERNAL来源）的 KMS 密钥仅支持On-demand轮换，因此没有轮换类型列。而是在每行显示导入状态、用户指定的描述、到期信息和一个操作菜单。导入状态要么是“已导入”，表示密钥材料在里面可用，要 Amazon KMS 么是待导入，表示密钥材料在里面不可用 Amazon KMS。操作菜单可用于删除导入的密钥材料或重新导入密钥材料。如果密钥材料的导入状态为待导入，则删除密钥材料操作将处于禁用状态。重新导入密钥材料操作将会始终可用。无需等待密钥材料过期或将其删除，即可重新导入密钥材料。

列出轮换和关键材料 (Amazon KMS API)

您可以使用 Amazon Key Management Service（Amazon KMS）API 启动按需密钥轮换，并查看任何客户自主管理型密钥的当前轮换状态。此示例使用 Amazon Command Line Interface（Amazon CLI），但您可以使用任何受支持的编程语言。

该ListKeyRotations操作列出了指定 KMS 密钥的所有轮换和密钥材料。要在这些操作中标识 KMS 密钥，请使用其密钥 ID 或密钥 ARN。

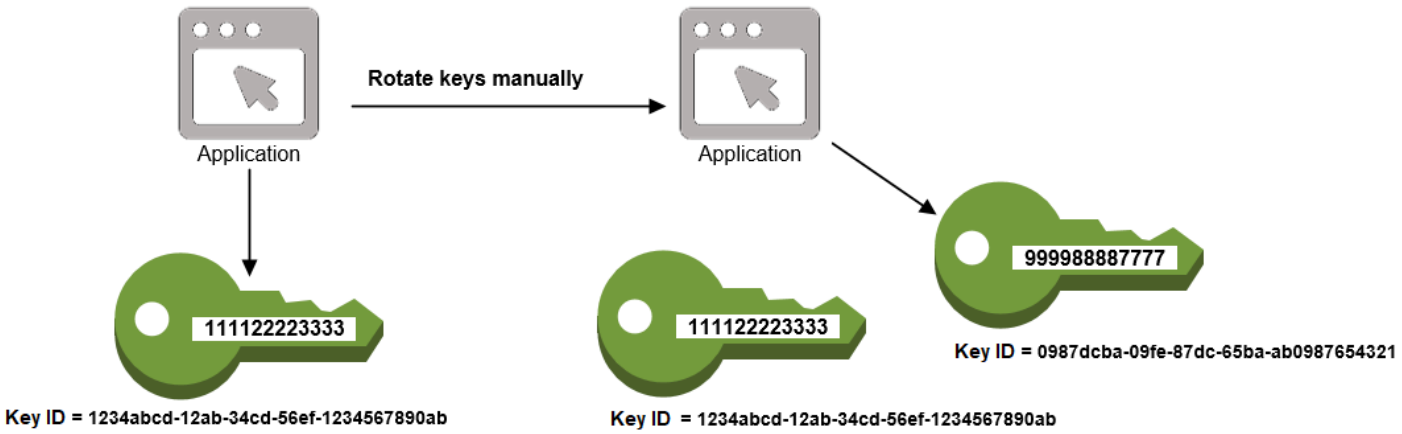
此操作支持可选的 IncludeKeyMaterial 参数。此参数的默认值为 ROTATIONS_ONLY。如果省略此参数，则 Amazon KMS 返回有关通过自动或按需密钥轮换创建的密钥材料的信息。如果指定 ALL_KEY_MATERIAL 值， Amazon KMS 会将第一个密钥材料和任何待轮换的导入密钥材料添加到响应中。此参数只能用于支持自动或按需密钥轮换的 KMS 密钥。

```
$ aws kms list-key-rotations --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --inlcude-key-material ALL_KEY_MATERIAL
{
    "Rotations": [
        {
            "KeyId": 1234abcd-12ab-34cd-56ef-1234567890ab,
            "KeyMaterialId":
 123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0,
            "KeyMaterialDescription": "KeyMaterialA",
            "ImportState": "PENDING_IMPORT",
            "KeyMaterialState": "NON_CURRENT"
        },
        {
            "KeyId": 1234abcd-12ab-34cd-56ef-1234567890ab,
            "KeyMaterialId":
 96083e4fb6dbc41d77578a213a6b6669c044dd4c143e96755396d2bf11fd6068,
            "ImportState": "IMPORTED",
```

```
            "KeyMaterialState": "CURRENT",
            "ExpirationModel": "KEY_MATERIAL_DOES_NOT_EXPIRE",
            "RotationDate": "2025-05-01T15:50:51.045000-07:00",
            "RotationType": "ON_DEMAND"
        }
    ],
    "Truncated": false
}
```

手动轮换密钥

可能需要创建新的 KMS 密钥并用其替换当前 KMS 密钥，而不使用自动或按需密钥轮换功能。当新的 KMS 密钥使用的加密材料与当前 KMS 密钥使用的加密材料不相同时，使用新的 KMS 密钥与更改现有 KMS 密钥的密钥材料具有相同的效果。使用一个 KMS 密钥替换另一个 KMS 密钥的过程被称为手动密钥轮换。



当您想要轮换不符合自动或按需密钥轮换条件的 KMS 密钥（例如非对称 KMS 密钥、HMAC KMS 密钥和自定义密钥存储中的 KMS 密钥）时，手动轮换是一个不错的选择。

> **Note**
>
> 开始使用新的 KMS 密钥时，请务必启用原始 KMS 密钥，以便 Amazon KMS 可以解密原始 KMS 密钥加密的数据。

当您手动轮换 KMS 密钥时，您还需要更新应用程序中对 KMS 密钥 ID 或密钥 ARN 的引用。将友好名称与 KMS 密钥关联的别名，可以使得这个过程变得更容易。使用别名来引用应用程序中的 KMS 密

钥。之后，当您想更改应用程序所使用的 KMS 密钥（而不是编辑应用程序代码）时，更改别名的目标 KMS 密钥即可。有关更多信息，请参阅 了解如何在您的应用程序中使用别名。

> **Note**
>
> 指向最新版本的手动轮换 KMS 密钥的别名是加密操作的好解决方案，例如 E n crypt DescribeKeyGetPublicKey、、、DeriveSharedSecret、、、S ign an GenerateDataKeyd GenerateDataKeyPairVeri VerifyMacfy。GenerateMac管理 KMS 密钥的操作中不允许使用别名，例如DisableKey或ScheduleKeyDeletion。
> 对手动轮换的对称加密 KMS 密钥调用 Decrypt 操作时，请省略命令中的KeyId参数。Amazon KMS 自动使用加密密文的 KMS 密钥。
> 使用非对称 KMS 密钥调用Decrypt或验证，或使用 HMAC KMS 密钥VerifyMac进行调用时，必须使用该KeyId参数。当 KeyId 参数的值是不再指向执行加密操作的 KMS 密钥的别名时，例如当手动轮换密钥时，这些请求将失败。为避免此错误，必须跟踪并为每个操作指定正确的 KMS 密钥。

要更改别名的目标 KMS 密钥，请使用 Amazon KMS API 中的UpdateAlias操作。例如，此命令会更新 alias/TestKey 别名以指向新 KMS 密钥。由于该操作不返回任何输出，因此该示例使用该ListAliases操作来显示别名现在已与其他 KMS 密钥相关联，并且该LastUpdatedDate字段已更新。这些 ListAliases 命令使用中的query参数仅 Amazon CLI 获取alias/TestKey别名。

```
$ aws kms list-aliases --query 'Aliases[?AliasName==`alias/TestKey`]'
{
    "Aliases": [
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/TestKey",
            "AliasName": "alias/TestKey",
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1521097200.123,
            "LastUpdatedDate": 1521097200.123
        },
    ]
}


$ aws kms update-alias --alias-name alias/TestKey --target-key-id
 0987dcba-09fe-87dc-65ba-ab0987654321

$ aws kms list-aliases --query 'Aliases[?AliasName==`alias/TestKey`]'
```

```
{
    "Aliases": [
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/TestKey",
            "AliasName": "alias/TestKey",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1521097200.123,
            "LastUpdatedDate": 1604958290.722
        },
    ]
}
```

更改一组多区域密钥中的主密钥

每组相关的多区域密钥都必须具有一个主密钥。但您可以更改主密钥。此操作称为更新主区域，会将当前主密钥转换为副本密钥，并将其中一个相关的副本密钥转换为主密钥。如果您需要在维护副本密钥的同时删除当前主密钥，或者在与密钥管理员相同的区域中查找主密钥，则可以执行此操作。

您可以选择任何相关的副本密钥作为新的主密钥。操作开始时，主密钥和副本密钥都必须处于 `Enabled` 密钥状态。

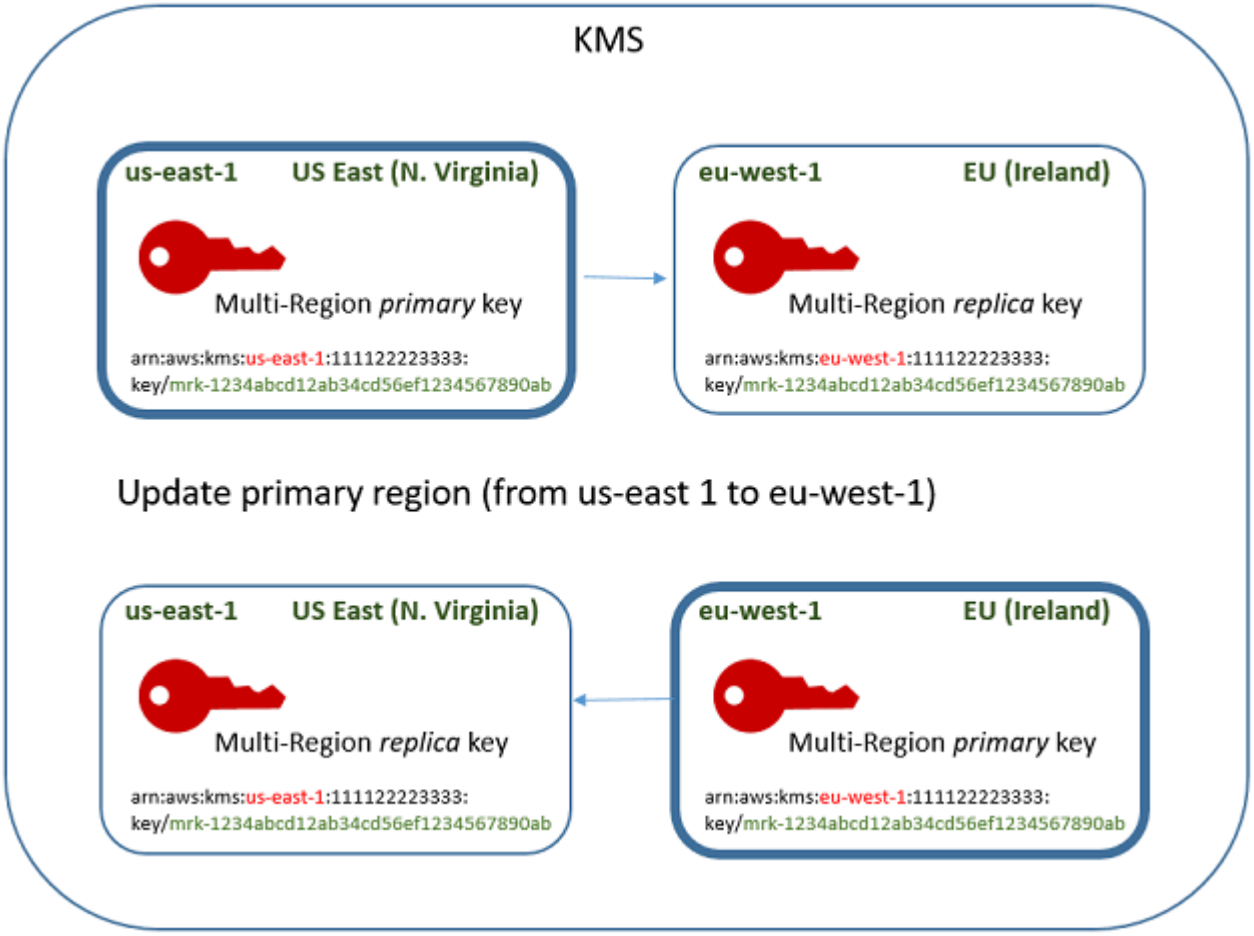
`Updating` 密钥状态

即使在 `UpdatePrimaryRegion` 操作完成后，更新主区域的过程可能仍要进行几秒钟。在此期间，旧的主密钥和新的主密钥具有临时密钥状态 Updating（正在更新）。当密钥状态为 `Updating` 时，您可以在加密操作中使用密钥，但不能复制新的主密钥或执行某些管理操作，例如启用或禁用这些密钥。DescribeKey 之类的操作可能会将旧的主密钥和新的主密钥同时显示为副本。当更新完成时，`Enabled` 密钥状态将恢复。

有关 `Updating` 密钥状态的影响的信息，请参阅 Amazon KMS 密钥的密钥状态。

工作方式

假设您在美国东部（弗吉尼亚北部）(us-east-1) 区域中有一个主密钥，在欧洲（爱尔兰）(eu-west-1) 区域有一个副本密钥。您可以使用更新功能将美国东部（弗吉尼亚北部）(us-east-1) 区域中的主密钥更改为副本密钥，并将欧洲（爱尔兰）(eu-west-1) 区域中的副本密钥更改为主密钥。



更新过程完成后，欧洲（爱尔兰）(eu-west-1) 区域中的多区域密钥为多区域主密钥，美国东部（弗吉尼亚北部）(us-east-1) 区域中的密钥是其副本密钥。如果存在其他相关的副本密钥，则它们将成为新主密钥的副本密钥。Amazon KMS 下一次同步多区域密钥的共享属性时，将从新的主密钥中获得共享属性并将它们复制到其副本密钥（包括以前的主密钥）中。

更新操作不会影响任何多区域密钥的密钥 ARN。它也不会影响共享属性（如密钥材料）或独立属性（如密钥策略）。不过，您可能需要对新的主密钥的密钥策略进行更新。例如，您可能要将信任委托人的 kms:ReplicateKey 权限添加到新的主密钥中，并将其从新的副本密钥中删除。

更新主区域

您可以将副本密钥转换为主密钥，从而将以前的主密钥更改为副本密钥。要更新主区域，您需要在这两个区域中具有 kms:UpdatePrimaryRegion 权限。

您可以在 Amazon KMS 控制台中或使用 UpdatePrimaryRegion 操作更新主区域。

使用 Amazon KMS 控制台

您可以在 Amazon KMS 控制台中更新主密钥。从当前主密钥的密钥详细信息页面开始。

1. 登录到 Amazon Web Services 管理控制台，然后通过以下网址打开 Amazon Key Management Service（Amazon KMS）控制台：https://console.aws.amazon.com/kms。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择客户托管密钥。
4. 选择多区域主密钥的密钥 ID 或别名。这将打开主密钥的密钥详细信息页面。

   要识别多区域主密钥，请使用右上角的工具图标将 Regionality（区域性）列添加到表中。
5. 选择 Regionality（区域性）选项卡。
6. 在 Primary key（主密钥）部分中，选择 Change primary Region（更改主区域）。
7. 选择新的主密钥的区域。您只能从菜单中选择一个区域。

   Change primary Regions（更改主区域）菜单仅包含具有相关多区域密钥的区域。您可能没有权限更新菜单上的所有区域中的主区域。
8. 选择 Change primary Region（更改主区域）。

使用 Amazon KMS API

要更改一组相关的多区域密钥中的主密钥，请使用 UpdatePrimaryRegion 操作。

使用 `KeyId` 参数来标识当前主密钥。使用 `PrimaryRegion` 参数来指示新的主密钥的 Amazon Web Services 区域。如果主密钥在新的主区域中还没有副本，则操作将失败。

以下示例将主密钥从 `us-west-2` 区域中的多区域密钥更改为其在 `eu-west-1` 区域中的副本密钥。`KeyId` 参数标识 `us-west-2` 区域中的当前主密钥。`PrimaryRegion` 参数指定新的主密钥的 Amazon Web Services 区域，即 `eu-west-1`。

```
$ aws kms update-primary-region \
      --key-id arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab \
      --primary-region eu-west-1
```

如果成功，此操作不会返回任何输出，只返回 HTTP 状态代码。要查看影响，请对任意一个多区域密钥调用 DescribeKey 操作。您可能需要等到密钥状态返回 `Enabled`。虽然密钥状态为 Updating（正在更新），但密钥的值可能仍处于变化中。

例如，以下 DescribeKey 调用将获取有关 eu-west-1 区域中多区域密钥的详细信息。输出显示，eu-west-1 区域中的多区域密钥现在为主密钥。us-west-2 区域中相关的多区域密钥（相同的密钥 ID）现在已成为副本密钥。

```
$ aws kms describe-key \
      --key-id arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab \

{
    "KeyMetadata": {
        "AWSAccountId": "111122223333",
        "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
        "Arn": "arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "CreationDate": 1609193147.831,
        "Enabled": true,
        "Description": "multi-region-key",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "Origin": "AWS_KMS",
        "KeyManager": "CUSTOMER",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "MultiRegion": true,
        "MultiRegionConfiguration": {
           "MultiRegionKeyType": "PRIMARY",
           "PrimaryKey": {
              "Arn": "arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
              "Region": "eu-west-1"
           },
           "ReplicaKeys": [
              {
                 "Arn": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                 "Region": "us-west-2"
              }
           ]
        }
    }
```

```
}
```

删除 Amazon KMS key

删除 Amazon KMS key具有破坏性和潜在危险性。这将删除密钥材料以及与 KMS 密钥关联的所有元数据，并且不可撤销。删除 KMS 密钥后，您不能再解密用该密钥加密的数据，这意味着该数据将无法恢复。（唯一的例外是多区域副本密钥以及带有导入密钥材料的非对称密钥和 HMAC KMS 密钥。）对于用于加密的非对称 KMS 密钥来说，这种风险非常大，在这种情况下，用户可以在没有警告或错误的情况下继续生成带有公钥的加密文字，而这些加密文字在从 Amazon KMS 中删除私钥后无法解密。

只有当您确定不再需要使用 KMS 密钥时，才能将其删除。如果您不确定，请考虑禁用 KMS 密钥，而不是将其删除。您可以重新启用被禁用的 KMS 密钥，取消 KMS 密钥的计划删除，但无法恢复已删除的 KMS。

您只能安排删除客户托管的密钥。您无法删除 Amazon 托管式密钥或 Amazon 拥有的密钥。

在删除 KMS 密钥之前，您可能想要了解使用该 KMS 密钥加密了多少密文。Amazon KMS 不会存储此信息，也不会存储任何密文。要获取此信息，您必须确定 KMS 密钥的过去使用情况。如需帮助，请转到 确定 KMS 密钥的过去使用情况。

Amazon KMS 从不会删除您的 KMS 密钥，除非您明确安排删除它们并且强制的等待期到期。

但是，您可能会出于以下一个或多个原因而选择删除 KMS 密钥：

- 完成不再需要的 KMS 密钥的密钥生命周期
- 避免因维护不用的 KMS 密钥而产生的管理开销和成本
- 减少计入您的 KMS 密钥资源配额的 KMS 密钥数量

> **Note**
>
> 如果关闭您的 Amazon Web Services 账户，您的 KMS 密钥将变得无法访问，您也不必再为它们付费。

在您计划删除 KMS 密钥且 KMS 密钥被实际删除时，Amazon KMS 会将一个条目记录在 Amazon CloudTrail 日志中。

关于等待期限

因为删除 KMS 密钥具有破坏性且存在潜在危险，所以 Amazon KMS 要求您将等待期限设置为 7-30 天。默认的等待期限为 30 天。

但是，实际等待期限可能最多比您计划的时间长 24 小时。要获取删除 KMS 密钥的实际日期和时间，请使用 DescribeKey 操作。或者在 Amazon KMS 控制台中的 KMS 密钥详细信息页面的 General configuration（常规配置）部分中，参阅计划删除日期。请务必记下时区。

在等待期限内，KMS 密钥状态和密钥状态为 Pending deletion（等待删除）。

- 待删除的 KMS 密钥不能用于任何加密操作。
- Amazon KMS 不会为待删除的 KMS 密钥轮换备用密钥。

等待期结束后，Amazon KMS 会删除 KMS 密钥、其别名以及所有相关的 Amazon KMS 元数据。

计划删除 KMS 密钥可能不会立即影响由 KMS 密钥加密的数据密钥。有关更多信息，请参阅 不可用的 KMS 密钥如何影响数据密钥。

请利用这段等待期来确保现在或将来都不需要 KMS 密钥。您可以配置 Amazon CloudWatch 告警，使其在有人员或应用程序在等待期间试图使用 KMS 密钥时发出警告。要恢复 KMS 密钥，您可以在等待期限结束前取消密钥删除。等待期限结束后，将无法取消密钥删除，Amazon KMS 将删除 KMS 密钥。

特殊注意事项

在计划删除密钥之前，请查看以下删除特殊用途 KMS 密钥的特殊注意事项。

删除非对称 KMS 密钥

获得授权的用户可以删除对称 KMS 密钥或非对称 KMS 密钥。对于两种密钥类型，计划删除 KMS 密钥的过程是相同的。但是，由于非对称 KMS 密钥的公有密钥可以下载并在 Amazon KMS 外部使用，因此操作会带来重大的额外风险，尤其是对于用于加密的非对称 KMS 密钥（密钥用法为 ENCRYPT_DECRYPT）。

- 您在计划删除 KMS 密钥时，KMS 密钥的密钥状态将更改为 Pending deletion（等待删除），并且 KMS 密钥将无法用在加密操作中。但是，计划删除对 Amazon KMS 外部的公有密钥没有影响。持有公有密钥的用户可以继续使用该密钥加密消息。他们不会收到密钥状态已更改的任何通知。除非删除取消，否则使用公有密钥创建的密文将无法解密。

- 告警、日志以及其他检测试图使用等待删除的 KMS 密钥的策略，无法检测到 Amazon KMS 外部公有密钥的使用。
- KMS 密钥删除后，涉及该 KMS 密钥的所有 Amazon KMS 操作都将失败。但是，持有公有密钥的用户可以继续使用该密钥加密消息。这些密文将无法解密。

如果必须删除密钥用法为 ENCRYPT_DECRYPT 的非对称 KMS 密钥，请使用 CloudTrail 日志条目确定是否已下载并共享公有密钥。如果有，请验证该公有密钥是否在 Amazon KMS 外部使用。然后，考虑禁用 KMS 密钥而不是将其删除。

对于具有导入密钥材料的非对称 KMS 密钥，可缓解删除非对称 KMS 密钥所带来的风险。有关更多信息，请参阅 Deleting KMS keys with imported key material。

删除多区域密钥

要删除主密钥，您必须计划删除其所有副本密钥，然后等待副本密钥被删除。删除主密钥所需的等待时间从删除主密钥的最后一个副本密钥开始。如果您必须从特定区域删除主密钥而不删除其副本密钥，请通过更新主区域将主密钥更改为副本密钥。

您可以随时删除副本密钥。它不依赖于任何其他 KMS 密钥的密钥状态。如果您误删了副本密钥，可以通过在同一区域中复制相同主密钥，以此方式来重新创建副本密钥。您创建的新副本密钥具有与原始副本密钥相同的共享属性。

删除具有导入密钥材料的 KMS 密钥

删除具有导入密钥材料的 KMS 密钥的密钥材料是临时的，并且是可撤销的。要恢复密钥，请重新导入其密钥材料。

相反，删除 KMS 密钥操作是不可逆的。如果您计划删除密钥而且所需等待期到期，Amazon KMS 会永久且不可撤销地删除 KMS 密钥、其密钥材料及与该 KMS 密钥关联的所有元数据。

但是，删除具有导入密钥材料的 KMS 密钥的风险和后果取决于 KMS 密钥的类型（“密钥规范”）。

- 对称加密密钥 - 如果删除对称加密 KMS 密钥，则所有由该密钥加密的其余加密文字都无法恢复。即使您拥有相同的密钥材料，也无法创建新的对称加密 KMS 密钥来解密已删除的对称加密 KMS 密钥的加密文字。每个 KMS 密钥独有的元数据以加密方式绑定到每个对称加密文字。此安全功能保证只有加密对称加密文字的 KMS 密钥才能解密该加密文字，但阻止您重新创建等效的 KMS 密钥。
- 非对称密钥和 HMAC 密钥 - 如果您拥有原始密钥材料，则可以创建与已删除的非对称密钥或 HMAC KMS 密钥具有相同加密属性的新 KMS 密钥。Amazon KMS 生成标准 RSA 加密文字和签名、ECC 签名和 HMAC 标签，其中不包含任何独特的安全功能。此外，您还可以在 Amazon 之外使用 HMAC 密钥或非对称密钥对的私有密钥。

使用相同的非对称或 HMAC 密钥材料创建的新 KMS 密钥将具有不同的密钥标识符。您必须创建新的密钥政策，重新创建所有别名，并更新现有的 IAM policy 和授权，以引用新的密钥。

从 Amazon CloudHSM 密钥存储中删除 KMS 密钥

如果您计划从 Amazon CloudHSM 密钥存储中删除 KMS 密钥，其密钥状态将变为 Pending deletion（等待删除）。KMS 密钥将在整个等待期处于 Pending deletion（等待删除）状态，即使 KMS 密钥因您断开自定义密钥存储而不可用时都是如此。这允许您在等待期内随时取消删除 KMS 密钥。

等待期到期后，Amazon KMS 将从 Amazon KMS 删除 KMS 密钥。然后，Amazon KMS 将尽可能从关联的 Amazon CloudHSM 集群中删除密钥材料。如果 Amazon KMS 无法删除密钥材料（如当密钥存储与 Amazon KMS 断开连接时），您可能需要手动从集群中删除孤立密钥材料。

Amazon KMS 不会从集群备份中删除密钥材料。即使您从 Amazon KMS 删除 KMS 密钥并且从 Amazon CloudHSM 集群删除其密钥材料，通过备份创建的集群也可能包含已删除的密钥材料。要永久删除密钥材料，请使用 DescribeKey 操作确定 KMS 密钥的创建日期。然后，删除所有集群备份（可能包含密钥材料）。

当您计划从 Amazon CloudHSM 密钥存储中删除 KMS 密钥时，KMS 密钥将立即变得不可用（取决于最终一致性）。不过，在再次使用 KMS 密钥（例如解密数据密钥）之前，使用受 KMS 密钥保护的数据密钥加密的资源不会受到影响。此问题会影响 Amazon Web Services 服务，因为许多服务使用数据密钥来保护您的资源。有关更多信息，请参阅 不可用的 KMS 密钥如何影响数据密钥。

从外部密钥存储中删除 KMS 密钥

从外部密钥存储中删除 KMS 密钥对作为其密钥材料的外部密钥没有影响。

如果您计划从外部密钥存储中删除 KMS 密钥，其密钥状态将变为 Pending deletion（待删除）。KMS 密钥将在整个等待期处于 Pending deletion（待删除）状态，即使 KMS 密钥因您断开外部密钥存储而不可用时都是如此。这允许您在等待期内随时取消删除 KMS 密钥。等待期到期后，Amazon KMS 将从 Amazon KMS 删除 KMS 密钥。

当您计划从外部密钥存储中删除 KMS 密钥时，KMS 密钥将立即变得不可用（取决于最终一致性）。不过，在再次使用 KMS 密钥（例如解密数据密钥）之前，使用受 KMS 密钥保护的数据密钥加密的资源不会受到影响。此问题会影响 Amazon Web Services 服务，因为许多服务使用数据密钥来保护您的资源。有关更多信息，请参阅 不可用的 KMS 密钥如何影响数据密钥。

控制密钥删除所需的权限

如果您使用 IAM policy 允许 Amazon KMS 权限，则具有 Amazon 管理员访问权限（`"Action": "*"`）或 Amazon KMS 完全访问权限（`"Action": "kms:*"`）的 IAM 身份，都已被允许计划和取消 KMS 密钥的密钥删除。要允许密钥管理员安排和取消密钥策略中的密钥删除，请使用 Amazon KMS 控制台或 Amazon KMS API。

通常，只有密钥管理员才有权安排或取消密钥删除。但是，您可以通过向密钥策略或 IAM policy 添加 `kms:ScheduleKeyDeletion` 和 `kms:CancelKeyDeletion` 权限，将这些权限授予其他 IAM 身份。您还可以使用 `kms:ScheduleKeyDeletionPendingWindowInDays` 条件键进一步限制主体可以在 ScheduleKeyDeletion 请求的 `PendingWindowInDays` 参数中指定的值。

允许密钥管理员安排和取消密钥删除

使用 Amazon KMS 控制台

授予密钥管理员计划和取消密钥删除的权限。

1. 登录到 Amazon Web Services 管理控制台，然后通过以下网址打开 Amazon Key Management Service（Amazon KMS）控制台：https://console.aws.amazon.com/kms。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择客户托管密钥。
4. 选择要更改其权限的 KMS 密钥的别名或密钥 ID。
5. 选择 key policy（密钥策略）选项卡。
6. 下一步操作具体取决于密钥策略的默认视图和策略视图。只有在使用默认控制台密钥策略时，默认视图才可用。否则，仅策略视图可用。

   当默认视图可用时，Key policy（密钥策略）选项卡上会出现 Switch to policy view（切换到策略视图）或 Switch to default view（切换到默认视图）按钮。

   - 在默认视图中：
     - 在 Key deletion（密钥删除）下，选择 Allow key administrators to delete this key（允许密钥管理员删除此密钥）。
   - 在策略视图中：
     a. 选择编辑。
     b. 在密钥管理员的策略语句中，向 `Action` 元素添加 `kms:ScheduleKeyDeletion` 和 `kms:CancelKeyDeletion` 权限。

```
{
  "Sid": "Allow access for Key Administrators",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:user/KMSKeyAdmin"},
  "Action": [
    "kms:Create*",
    "kms:Describe*",
    "kms:Enable*",
    "kms:List*",
    "kms:Put*",
    "kms:Update*",
    "kms:Revoke*",
    "kms:Disable*",
    "kms:Get*",
    "kms:Delete*",
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": "*"
}
```

c. 选择保存更改。

使用 Amazon KMS API

您可以使用 Amazon Command Line Interface来添加计划和取消密钥删除所需的权限。

添加计划和取消密钥删除所需的权限

1. 使用 aws kms get-key-policy 命令检索现有的密钥策略，然后将策略文档保存到文件中。
2. 在您的首选文本编辑器中打开策略文档。在密钥管理员的策略语句中，添加 kms:ScheduleKeyDeletion 和 kms:CancelKeyDeletion 权限。以下示例显示了一个具有这两项权限的策略语句：

```
{
  "Sid": "Allow access for Key Administrators",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:user/KMSKeyAdmin"},
  "Action": [
    "kms:Create*",
    "kms:Describe*",
```

```
    "kms:Enable*",
    "kms:List*",
    "kms:Put*",
    "kms:Update*",
    "kms:Revoke*",
    "kms:Disable*",
    "kms:Get*",
    "kms:Delete*",
    "kms:ScheduleKeyDeletion",
    "kms:CancelKeyDeletion"
  ],
  "Resource": "*"
}
```

3. 使用 aws kms put-key-policy 命令将密钥策略应用于 KMS 密钥。

计划密钥删除

以下过程介绍如何通过 Amazon Web Services 管理控制台 和 Amazon KMS API 在 Amazon KMS 中计划密钥删除和取消 Amazon KMS keys（KMS 密钥）的密钥删除。

> **Warning**
>
> 删除 KMS 密钥具有破坏性和潜在危险性。只有当您确定不再需要使用 KMS 密钥并且将来也不再需要了，才能继续删除操作。如果您不确定，则应禁用 KMS 密钥，而不是将其删除。

在删除 KMS 密钥之前，您必须具有执行这一操作的权限。有关向密钥管理员授予这些权限的信息，请参阅 控制密钥删除所需的权限。您也可以使用 kms:ScheduleKeyDeletionPendingWindowInDays 条件键进一步限制等待时间，例如强制规定最短等待时间。

在您计划删除 KMS 密钥且 KMS 密钥被实际删除时，Amazon KMS 会将一个条目记录在 Amazon CloudTrail 日志中。

使用 Amazon KMS 控制台

在 Amazon Web Services 管理控制台 中，您可以一次安排和取消删除多个 KMS 密钥。

计划密钥删除

1. 登录到 Amazon Web Services 管理控制台，然后通过以下网址打开 Amazon Key Management Service（Amazon KMS）控制台：https://console.aws.amazon.com/kms。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择客户托管密钥。

   您无法安排删除 Amazon 托管式密钥或 Amazon 拥有的密钥。
4. 选择想要删除的 KMS 密钥旁边的复选框。
5. 依次选择 Key actions (密钥操作)、Schedule key deletion (计划密钥删除)。
6. 阅读并考虑警告，以及有关在等待期限内取消删除的信息。如果决定取消删除，请在页面底部选择 Cancel（取消）。
7. 对于 Waiting period (in days) (等待期限(天))，键入一个介于 7 和 30 之间的天数。
8. 查看正在删除的 KMS 密钥。
9. 选中 Confirm you want to schedule this key for deletion in ***<number of days>*** days（确认您要计划在 n 天后删除此密钥）旁的复选框。
10. 选择计划删除。

KMS 密钥状态将更改为等待删除。

使用 Amazon KMS API

使用 `aws kms schedule-key-deletion` 命令安排删除客户托管的密钥，如以下示例所示。

您无法计划删除 Amazon 托管式密钥或 Amazon 拥有的密钥。

```
$ aws kms schedule-key-deletion --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --
pending-window-in-days 10
```

使用成功后，Amazon CLI 将返回与以下示例中显示的输出类似的输出：

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "DeletionDate": 1598304792.0,
    "KeyState": "PendingDeletion",
    "PendingWindowInDays": 10
}
```

取消密钥删除

计划删除 KMS 密钥后，您可以在密钥仍处于待删除状态时取消删除该密钥。您可以在 Amazon KMS 控制台中取消密钥删除，也可以使用 CancelKeyDeletion 操作来取消密钥删除。对待删除的 KMS 密钥取消删除后，该 KMS 密钥的密钥状态为 `Disabled`。有关启用 KMS 密钥的更多信息，请参阅启用和禁用密钥。

使用 Amazon KMS 控制台

取消密钥删除

1. 从 https://console.aws.amazon.com/kms 打开 Amazon KMS 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择客户托管密钥。
4. 选中想要恢复的 KMS 密钥旁边的复选框。
5. 依次选择 Key actions (密钥操作)、Cancel key deletion (取消密钥删除)。

KMS 密钥状态将从待删除更改为已禁用。要使用 KMS 密钥，您必须将其启用。

使用 Amazon KMS API

从 使用 aws kms cancel-key-deletionAmazon CLI 命令取消密钥删除，如以下示例所示。

```
$ aws kms cancel-key-deletion --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

使用成功后，Amazon CLI 将返回与以下示例中显示的输出类似的输出：

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
}
```

KMS 密钥的状态将从待删除更改为已禁用。要使用 KMS 密钥，您必须将其启用。

创建检测待删除 KMS 密钥的使用的警报

可将 Amazon CloudTrail、Amazon CloudWatch Logs 和 Amazon Simple Notification Service (Amazon SNS) 的功能结合使用，以创建在您账户中的某个人员尝试使用等待删除的 KMS 密钥时向您

发送通知的 Amazon CloudWatch 警报。如果您收到此通知，则可能要取消删除该 KMS 密钥并重新考虑删除它的决定。

以下过程创建在 "*Key ARN* is pending deletion" 错误消息写入 CloudTrail 日志文件时向您发送通知的警报。此错误消息指示有人或应用程序尝试在加密操作中使用该 KMS 密钥。由于通知链接至错误消息，因此，当您使用待删除的 KMS 密钥中允许的 API 操作时（例如 ListKeys、CancelKeyDeletion 和 PutKeyPolicy），不会触发该通知。要查看返回此错误消息的 Amazon KMS API 操作的列表，请参阅Amazon KMS 密钥的密钥状态。

您收到的通知电子邮件不会列出 KMS 密钥或加密操作。可在 CloudTrail 日志中找到此信息。电子邮件会报告警报状态已从 OK (正常) 变为 Alarm (警报)。有关 CloudWatch 警报和状态更改的更多信息，请参阅 Amazon CloudWatch 用户指南 中的使用 Amazon CloudWatch 警报。

> **Warning**
>
> 此 Amazon CloudWatch 告警无法检测到 Amazon KMS 外部非对称 KMS 密钥的公有密钥的使用情况。有关删除用于公有密钥加密的非对称 KMS 密钥的特殊风险的详细信息，包括创建无法解密的密文，请参阅 Deleting asymmetric KMS keys。

在此过程中，您将创建 CloudWatch 日志组指标筛选条件 ，用于查找待删除异常的实例。然后，您根据日志组指标创建 ClouDWatch 警报。有关日志组指标筛选条件更多信息，请参阅《Amazon CloudWatch Logs 用户指南》中的使用筛选条件从日志事件创建指标。

1. 创建一个解析 CloudTrail 日志的 CloudWatch 指标筛选条件。

   使用以下必填值，按照为日志组创建指标筛选条件中的说明进行操作。对于其他字段，请接受默认值并按要求提供名称。

Field	Value
筛选条件模式	`{ $.eventSource = kms* && $.errorMessage = "* is pending deletion."}`
指标值	1

2. 根据您在步骤 1 中创建的指标筛选条件创建 CloudWatch 警报。

   使用以下必填值，按照根据日志组指标筛选条件创建 CloudWatch 警报中的说明进行操作。对于其他字段，请接受默认值并按要求提供名称。

Field	Value
指标筛选条件	您在步骤 1 中创建的指标筛选条件名称。
阈值类型	静态
条件	每当 *metric-name* 大于/等于 1 时
要发出警报的数据点	1 / 1
缺失数据处理	将缺失的数据作为好处理 (未超出阈值)

完成此过程后，每当新的 CloudWatch 警报进入 ALARM 状态时，您都会收到通知。如果您收到此警报的通知，可能意味着仍然需要计划删除的 KMS 密钥来加密或解密数据。在这种情况下，请取消删除 KMS 密钥并重新考虑删除它的决定。

确定 KMS 密钥的过去使用情况

在删除 KMS 密钥之前，您可能想要了解使用该 KMS 密钥加密了多少密文。Amazon KMS 不会存储此信息，也不会存储任何密文。了解 KMS 密钥的过去使用情况可帮助您决定将来是否需要此 KMS 密钥。本主题建议多种有助于您确定 KMS 密钥的过去使用情况的策略。

> **Warning**
>
> 这些用于确定过去使用情况和实际使用情况的策略仅对 Amazon 用户和 Amazon KMS 操作有效。它们无法检测到 Amazon KMS 外部非对称 KMS 密钥的公有密钥的使用情况。有关删除用于公有密钥加密的非对称 KMS 密钥的特殊风险的详细信息，包括创建无法解密的密文，请参阅 Deleting asymmetric KMS keys。

主题

- 检查 KMS 密钥权限以确定潜在使用范围
- 检查 Amazon CloudTrail 日志以确定实际使用情况

检查 KMS 密钥权限以确定潜在使用范围

通过确定当前有权访问 KMS 密钥的对象，可帮助您确定 KMS 密钥的广泛使用程度以及是否仍然需要此 KMS 密钥。要了解如何确定当前有权访问 KMS 密钥的对象，请转到 确定对 Amazon KMS keys 的访问权限。

检查 Amazon CloudTrail 日志以确定实际使用情况

您或许能够使用 KMS 密钥使用情况历史记录来帮助确定是否使用特定 KMS 密钥加密了密文。

所有 Amazon KMS API 活动都记录在 Amazon CloudTrail 日志文件中。如果您在您的 KMS 密钥所在的区域创建了 CloudTrail 跟踪，您可以检查您的 CloudTrail 日志文件，以查看特定 KMS 密钥的所有 Amazon KMS API 活动的历史记录。如果您没有跟踪，则仍可在 CloudTrail 事件历史记录中查看最近的事件。有关 Amazon KMS 如何使用 CloudTrail 的详细信息，请参阅 使用 Amazon CloudTrail 记录 Amazon KMS API 调用。

以下示例显示了使用 KMS 密钥保护存储在 Amazon Simple Storage Service (Amazon S3) 中的对象时生成的 CloudTrail 日志条目。在本示例中，对象通过使用 KMS 密钥 (SSE-KMS) 的服务器端加密保护数据上传到 Amazon S3 中。当您使用 SSE-KMS 将对象上传到 Amazon S3 时，需指定用于保护对象的 KMS 密钥。Amazon S3 使用 Amazon KMS GenerateDataKey 操作来请求用于对象的唯一数据密钥，此请求事件会记录在 CloudTrail 中，记录条目类似于：

```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROACKCEVSQ6C2EXAMPLE:example-user",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admins/example-user",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
        "creationDate": "2015-09-10T23:12:48Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/Admins",
        "accountId": "111122223333",
        "userName": "Admins"
      }
```

```
    },
    "invokedBy": "internal.amazonaws.com"
  },
  "eventTime": "2015-09-10T23:58:18Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "GenerateDataKey",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "internal.amazonaws.com",
  "userAgent": "internal.amazonaws.com",
  "requestParameters": {
    "encryptionContext": {"aws:s3:arn": "arn:aws:s3:::example_bucket/example_object"},
    "keySpec": "AES_256",
    "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "responseElements": null,
  "requestID": "cea04450-5817-11e5-85aa-97ce46071236",
  "eventID": "80721262-21a5-49b9-8b63-28740e7ce9c9",
  "readOnly": true,
  "resources": [{
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "accountId": "111122223333"
  }],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

当您之后从 Amazon S3 下载此对象时，Amazon S3 会向 Amazon KMS 发送 `Decrypt` 请求，以使用指定的 KMS 密钥解密对象的数据密钥。执行该操作时，CloudTrail 日志文件将包含一个与下类似的条目：

```
{
  "eventVersion": "1.02",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AROACKCEVSQ6C2EXAMPLE:example-user",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admins/example-user",
    "accountId": "111122223333",
    "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "attributes": {
        "mfaAuthenticated": "false",
```

```
        "creationDate": "2015-09-10T23:12:48Z"
      },
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AROACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/Admins",
        "accountId": "111122223333",
        "userName": "Admins"
      }
    },
    "invokedBy": "internal.amazonaws.com"
  },
  "eventTime": "2015-09-10T23:58:39Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "Decrypt",
  "awsRegion": "us-west-2",
  "sourceIPAddress": "internal.amazonaws.com",
  "userAgent": "internal.amazonaws.com",
  "requestParameters": {
    "encryptionContext": {"aws:s3:arn": "arn:aws:s3:::example_bucket/example_object"}},
  "responseElements": null,
  "requestID": "db750745-5817-11e5-93a6-5b87e27d91a0",
  "eventID": "ae551b19-8a09-4cfc-a249-205ddba330e3",
  "readOnly": true,
  "resources": [{
    "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "accountId": "111122223333"
  }],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

CloudTrail 将记录所有 Amazon KMS API 活动。通过评估这些日志条目，您可以确定特定 KMS 密钥的过去使用情况，这将有助于您确定是否需要将其删除。

要查看有关 Amazon KMS API 活动如何显示在 CloudTrail 日志文件中的更多示例，请转到 使用 Amazon CloudTrail 记录 Amazon KMS API 调用。有关 CloudTrail 的更多信息，请转到 Amazon CloudTrail 用户指南。

删除导入的密钥材料

您可以随时从 KMS 密钥中删除导入的密钥材料。此外，当导入的带有过期日期的密钥材料到期时，Amazon KMS 会删除该密钥材料。在任意一种情况下，密钥材料删除后，KMS 密钥的密钥状态将变为待导入，并且该 KMS 密钥不能用于任何密码操作。

对称加密密钥可以有多个与之关联的密钥材料，任何处于非状态的密钥材料的删除或过期都会将密钥状态`PENDING_ROTATION`更改为 “待导入”。对于此类密钥，KMS 会为每个密钥材料分配一个唯一的标识符。您可以使用 ListKeyRotationsAPI 来查看这些密钥材料标识符。您可以使用 DeleteImportedKeyMaterialAPI 中的`key-material-id`参数指定其标识符来删除特定的密钥材料。

多区域密钥的注意事项

- 当您删除`PENDING_MULTI_REGION_IMPORT_AND_ROTATION`处于`PENDING_ROTATION`或状态的主区域密钥的密钥材料时，也将删除副本区域密钥的密钥材料。
- 如果您删除副本区域密钥的密钥材料，则主区域密钥和其他副本区域密钥将保持不变。

> **Warning**
>
> `key-material-id` 参数是可选的，如果您不指定该参数，则 Amazon KMS 会删除当前密钥材料。

除了禁用 KMS 密钥和撤回权限外，还可以将删除密钥材料用作一种策略，以快速但暂时地停止使用 KMS 密钥。相比之下，计划删除具有导入密钥材料的 KMS 密钥也会很快停止使用 KMS 密钥。但如果未在等待期内取消删除，则会永久删除 KMS 密钥、关联的密钥材料和所有密钥元数据。有关更多信息，请参阅 Deleting KMS keys with imported key material。

要删除密钥材料，您可以使用 Amazon KMS 控制台或 DeleteImportedKeyMaterialAPI 操作。 Amazon KMS 当您删除导入的密钥材料和Amazon KMS 删除过期的密钥材料时，会在 Amazon CloudTrail 日志中记录一个条目。

删除密钥材料如何影响 Amazon 服务

删除任何密钥材料后，该 KMS 密钥将立即变为不可使用（视最终一致性而定）。不过，在再次使用 KMS 密钥（例如解密数据密钥）之前，使用受 KMS 密钥保护的数据密钥加密的资源不会受到影响。此问题会影响 Amazon Web Services 服务，其中许多使用数据密钥来保护您的资源。有关更多信息，请参阅 不可用的 KMS 密钥如何影响数据密钥。

使用控制 Amazon KMS 台

您可以使用 Amazon KMS 控制台删除密钥材料。

1. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/km s 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择客户托管密钥。
4. 请执行以下操作之一：

   - 选中带导入密钥材料的 KMS 密钥对应的复选框。依次选择 Key actions、Delete key material。对于与多个密钥材料关联的对称加密密钥，这将导致删除当前密钥材料。
   - 对于带有导入密钥材料的对称加密 KMS 密钥，请选择 KMS 密钥的别名或密钥 ID。选择按密钥材料和轮换选项卡。密钥材料表会列出与该密钥关联的所有密钥材料。在要删除的密钥材料对应的行中，选择操作菜单中的删除密钥材料。
5. 确认要删除该密钥材料，然后选择 Delete key material。KMS 密钥的状态（对应于其密钥状态）更改为 Pending import（等待导入）。如果已删除的密钥材料处于 PENDING_ROTATION 状态，则 KMS 密钥的状态不会发生变化。

使用 Amazon KMS API

要使用 Amazon KMS API 删除密钥材料，DeleteImportedKeyMaterial请发送请求。以下示例说明如何使用 Amazon CLI 执行该操作。

将 *1234abcd-12ab-34cd-56ef-1234567890ab* 替换为您要删除其密钥材料的 KMS 密钥的密钥 ID。在该操作中，您可以使用 KMS 密钥的密钥 ID 或 ARN，但不能使用别名。以下命令会删除当前密钥材料，这可能是与该密钥关联的唯一密钥材料。

```
$ aws kms delete-imported-key-material --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

要删除特定的密钥材料，请指定使用 key-material-id 参数标识的密钥材料。请将 *123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0* 替换为要删除的密钥材料的标识符。

```
$ aws kms delete-imported-key-material --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --key-material-id 123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0
```

生成数据密钥

数据密钥是可用于加密数据的对称密钥，包括大量数据和其他数据加密密钥。与无法下载的对称 KMS 密钥不同的是，数据密钥可以返回给您在 Amazon KMS 外部使用。

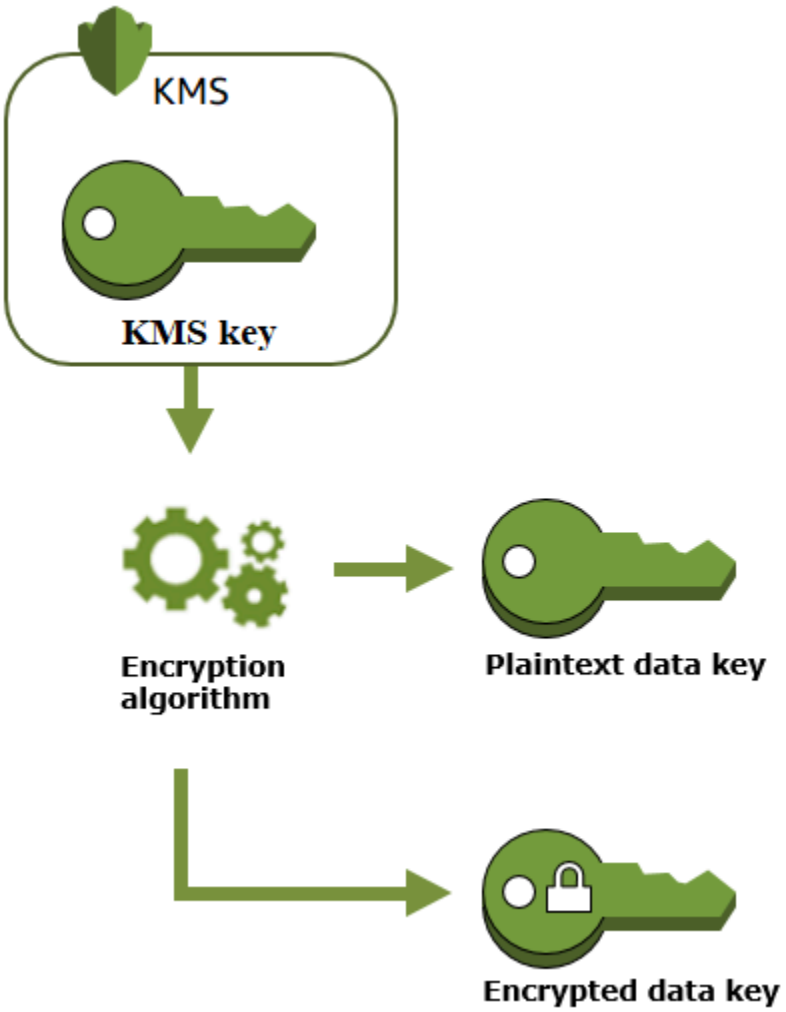
当 Amazon KMS 生成数据密钥时，它会返回供立即使用的明文数据密钥（可选）和可以随数据安全存储的数据密钥的加密副本。准备好解密数据时，首先要求 Amazon KMS 解密已加密的数据密钥。

Amazon KMS 会生成、加密和解密数据密钥。但是，Amazon KMS 不会存储、管理或跟踪您的数据密钥，也不会使用数据密钥执行加密操作。您必须在 Amazon KMS 之外使用和管理数据密钥。有关安全使用数据密钥的帮助，请参阅 Amazon Encryption SDK。

主题

创建数据密钥

要创建数据密钥，请调用 GenerateDataKeyPair 操作。Amazon KMS 生成数据密钥 然后，它会在您指定的对称加密 KMS 密钥下加密数据密钥的副本。此操作会返回数据密钥的明文副本以及由 KMS 密钥加密的数据密钥的副本。下图展示了此操作。



Amazon KMS 还支持 GenerateDataKeyWithoutPlaintext 操作，此操作仅返回加密的数据密钥。当您需要使用数据密钥时，请要求 Amazon KMS 解密它。

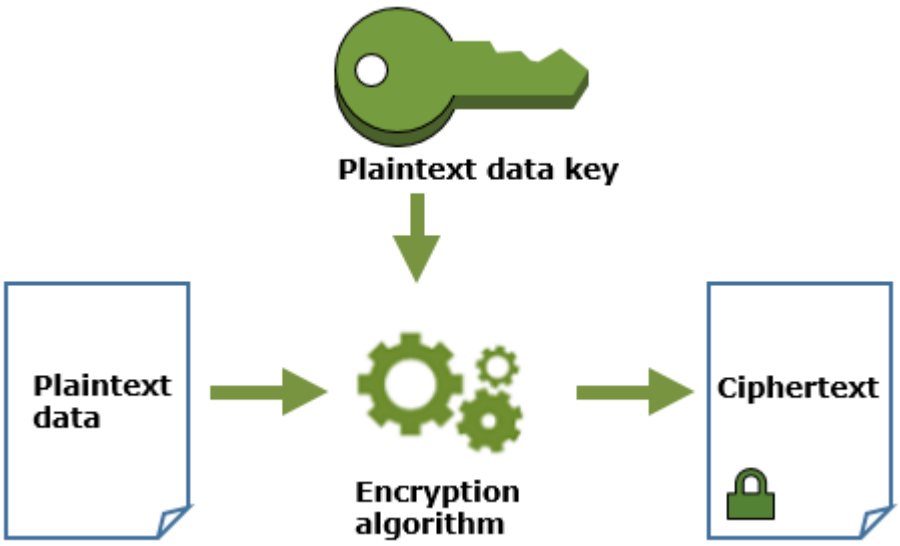
如何使用数据密钥进行加密操作

以下主题说明了由 GenerateDataKey 或 GenerateDataKeyWithoutPlaintext 生成的数据密钥的工作原理。

使用数据密钥加密数据

Amazon KMS 无法使用数据密钥来加密数据。但您可以在 Amazon KMS 之外使用数据密钥，例如使用 OpenSSL 或 Amazon Encryption SDK 等加密库。

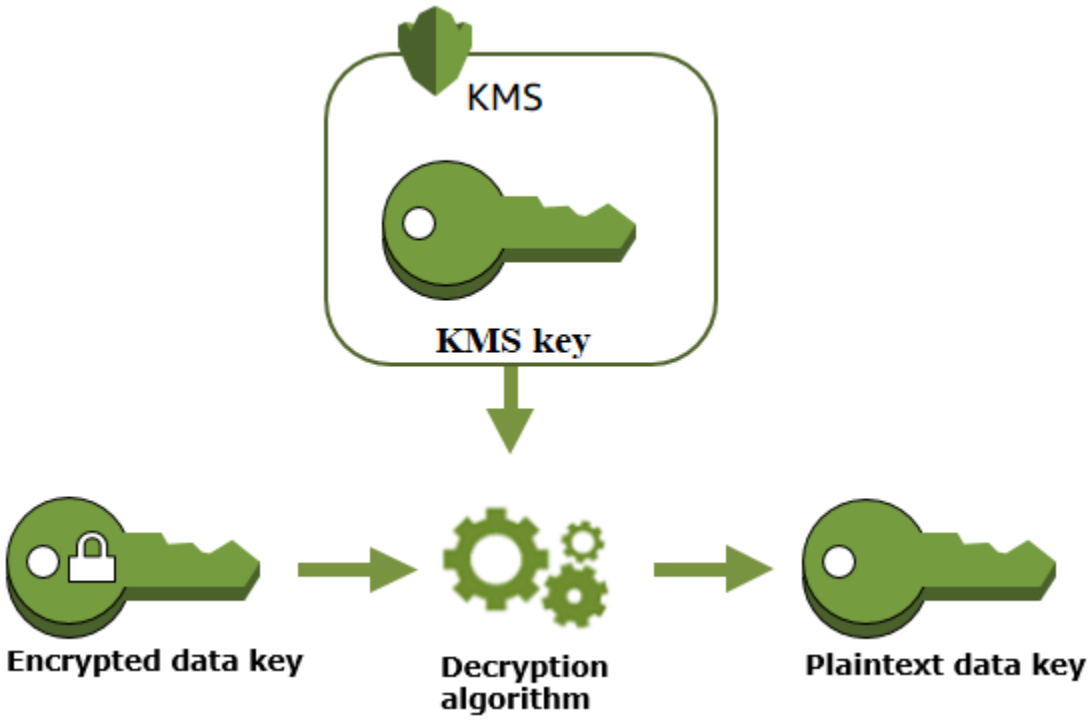
在使用明文数据密钥加密数据后，请尽快从内存中将其删除。您可以安全地存储加密数据密钥及加密数据，以便其可根据需要用于解密数据。



使用数据密钥解密数据

要解密数据，请将加密数据密钥传递至 Decrypt 操作。Amazon KMS 使用您的 KMS 密钥解密数据密钥，然后再返回明文数据密钥。使用明文数据密钥解密数据，并尽快从内存中删除该明文数据密钥。

下图显示了如何使用 Decrypt 操作解密加密的数据密钥。



不可用的 KMS 密钥如何影响数据密钥

当 KMS 密钥不可用时，您可以立即发现（取决于最终一致性）。KMS 密钥的密钥状态会出现变更，以反映其新情况，并且加密操作中使用 KMS 密钥的所有请求都将失败。

但是，对由 KMS 密钥加密的数据密钥，以及由数据密钥加密的数据的影响会延迟，直至再次使用 KMS 密钥（例如用于解密数据密钥）。

KMS 密钥状态变为不可用的原因有许多，包括您可能执行的以下操作。

- 禁用 KMS 密钥
- 安排删除 KMS 密钥
- 从具有已导入密钥材料的 KMS 密钥中删除密钥材料，或允许导入的密钥材料过期。如果来源为 EXTERNAL 的 KMS 密钥关联了多个密钥材料，则任何密钥材料被删除或过期都将导致该密钥无法使用。
- 断开托管 KMS 密钥的 Amazon CloudHSM 密钥存储的连接，或从用作 KMS 密钥的密钥材料的 Amazon CloudHSM 集群中删除密钥。
- 断开托管 KMS 密钥的外部密钥存储的连接，或干扰对外部密钥存储代理的加密和解密请求的任何其他操作，包括从其外部密钥管理器中删除外部密钥。

对于许多使用数据密钥来保护服务所管理的资源的 Amazon Web Services 服务 来说，这种效果尤其重要。以下几个示例使用 Amazon Elastic Block Store（Amazon EBS）和 Amazon Elastic Compute Cloud（Amazon EC2）。不同的 Amazon Web Services 服务 以不同方式使用数据密钥。有关详细信息，请参阅 Amazon Web Services 服务 的“安全性”一章的“数据保护”部分。

例如，考虑以下情景：

1. 您创建加密 EBS 卷并指定 KMS 密钥来保护该卷。Amazon EBS 要求 Amazon KMS 使用您的 KMS 密钥来为该卷生成加密数据密钥。Amazon EBS 使用该卷的元数据存储加密数据密钥。
2. 当您将 EBS 卷附加到 EC2 实例时，Amazon EC2 使用您的 KMS 密钥来解密 EBS 卷的加密数据密钥。Amazon EC2 使用 Nitro 硬件中的数据密钥，该密钥负责加密 EBS 卷的所有磁盘输入/输出。EBS 卷附加到 EC2 实例时，数据密钥会保留在 Nitro 硬件中。
3. 您执行的操作会使 KMS 密钥不可用。这不会立即影响 EC2 实例或 EBS 卷。卷附加到实例时，Amazon EC2 使用数据密钥（而不是 KMS 密钥），来对所有磁盘输入/输出进行加密。
4. 但是，当加密的 EBS 卷从 EC2 实例分离时，Amazon EBS 将从 Nitro 硬件中删除该数据密钥。下次将加密的 EBS 卷附加到 EC2 实例时，附加会失败，因为 Amazon EBS 无法使用 KMS 密钥来解密卷的加密数据密钥。要再次使用 EBS 卷，您必须使该 KMS 密钥可重新使用。

生成数据密钥对

非对称 KMS 密钥表示“数据密钥对”。数据密钥对是由数学上相关的公有密钥和私有密钥组成的非对称数据密钥。它们设计用于 Amazon KMS 外部的客户端加密和解密、签名和验证，或者在两个对等方之间建立共享密钥。

与 OpenSSL 等工具生成的数据密钥对不同，Amazon KMS 会将每个数据密钥对中的私有密钥置于您指定的 Amazon KMS 中的对称加密 KMS 密钥的保护之下。但是，Amazon KMS 不会存储、管理或跟踪数据密钥对，也不会使用数据密钥对执行加密操作。您必须在 Amazon KMS 之外使用和管理数据密钥对。

主题

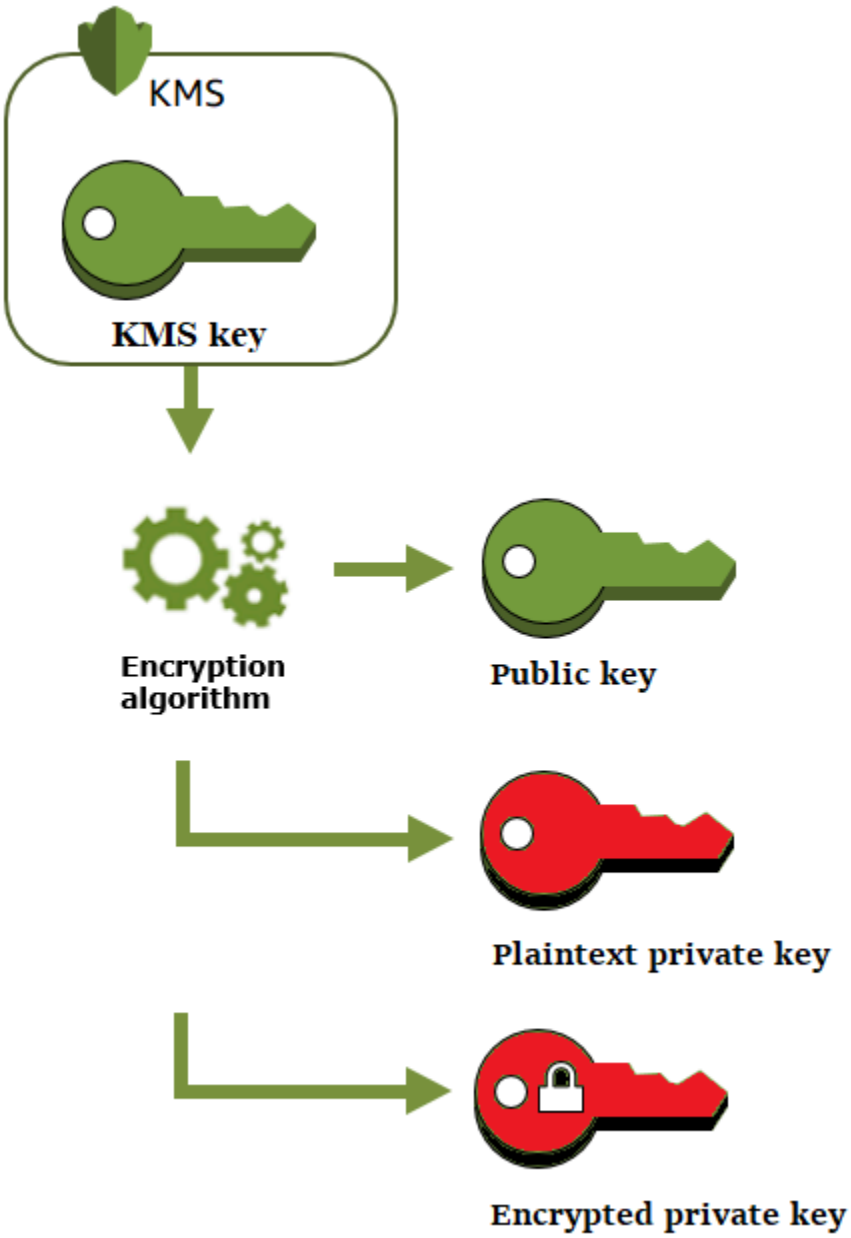
创建数据密钥对

要创建数据密钥对，请调用 GenerateDataKeyPair 或 GenerateDataKeyPairWithoutPlaintext 操作。指定要用于加密私有密钥的对称加密 KMS 密钥。

`GenerateDataKeyPair` 返回一个明文公有密钥、一个明文私有密钥和一个加密的私有密钥。如果您即刻需要明文私有密钥，例如生成数字签名，则可使用此操作。

`GenerateDataKeyPairWithoutPlaintext` 返回一个明文公有密钥和一个加密的私有密钥，但不返回明文私有密钥。如果您并非即刻需要明文私有密钥，例如使用公有密钥进行加密，则可使用此操作。稍后，如果您需要明文私有密钥来解密数据，则可调用 Decrypt 操作。

下图显示了 `GenerateDataKeyPair` 操作。`GenerateDataKeyPairWithoutPlaintext` 操作省略了明文私有密钥。



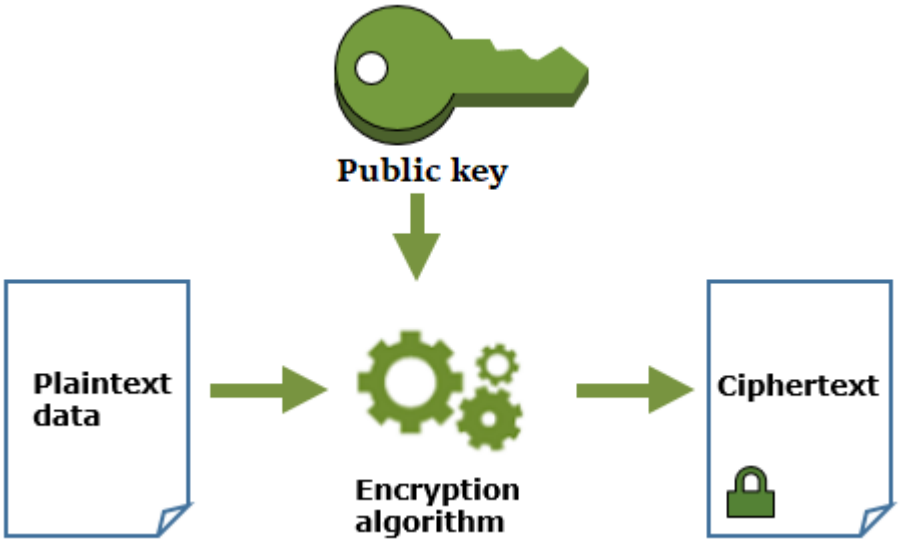
如何使用数据密钥对进行加密操作

以下主题说明了您可以对由 GenerateDataKeyPair 或 GenerateDataKeyPairWithoutPlaintext 生成的数据密钥对执行哪些加密操作，以及它们的工作原理。

使用数据密钥对加密数据

使用数据密钥对加密时，用该密钥对的公有密钥加密数据，然后用同一密钥对的私有密钥解密数据。通常，当多方需要加密数据，而只有持有私有密钥的一方才能解密该数据时，您可以使用数据密钥对。

持有公有密钥的多方使用该密钥加密数据，如下图所示。

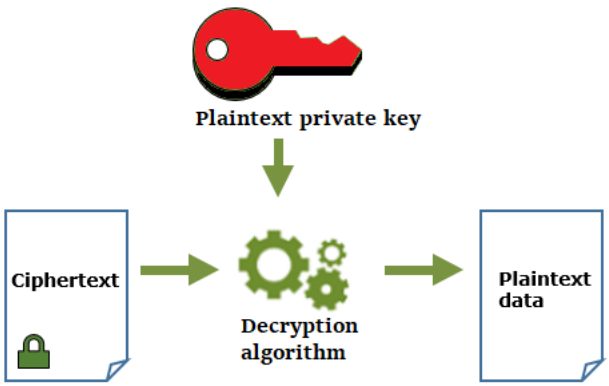


使用数据密钥对解密数据

要解密数据，请使用数据密钥对中的私有密钥。为使操作成功，公有密钥和私有密钥必须来自同一数据密钥对，并且必须使用相同的加密算法。

要对加密的私有密钥进行解密，请将其传递给 Decrypt 操作。使用明文私有密钥解密数据。然后尽快从内存中删除明文私有密钥。

下图显示了如何使用数据密钥对中的私有密钥解密密文。



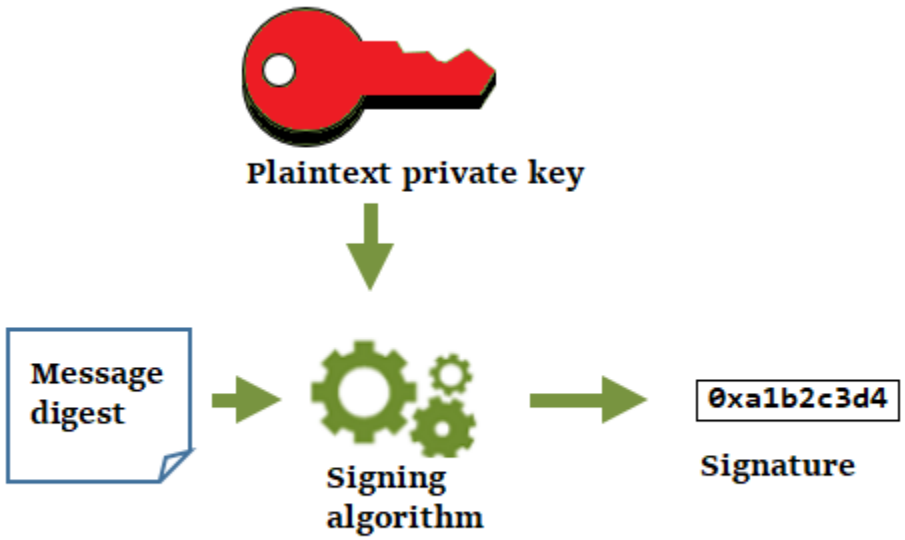
使用数据密钥对签署消息

要为消息生成加密签名，请使用数据密钥对中的私有密钥。持有公有密钥的任何人都可以使用该密钥来验证消息已使用私有密钥签名，并且自签名以后未曾更改。

如果加密私有密钥，请将加密的私有密钥传递给 Decrypt 操作。Amazon KMS 使用 KMS 密钥对数据密钥进行解密，然后返回明文私有密钥。使用明文私有密钥生成签名。然后尽快从内存中删除明文私有密钥。

要签署消息，请使用加密哈希函数（如 OpenSSL 中的 `dgst` 命令）创建消息摘要。然后，将明文私有密钥传递给签名算法。结果是一个表示消息内容的签名。（您可能无需先创建摘要即可签署较短的消息。最大消息大小因您使用的签名工具而异。）

下图显示了如何使用数据密钥对中的私有密钥签署消息。

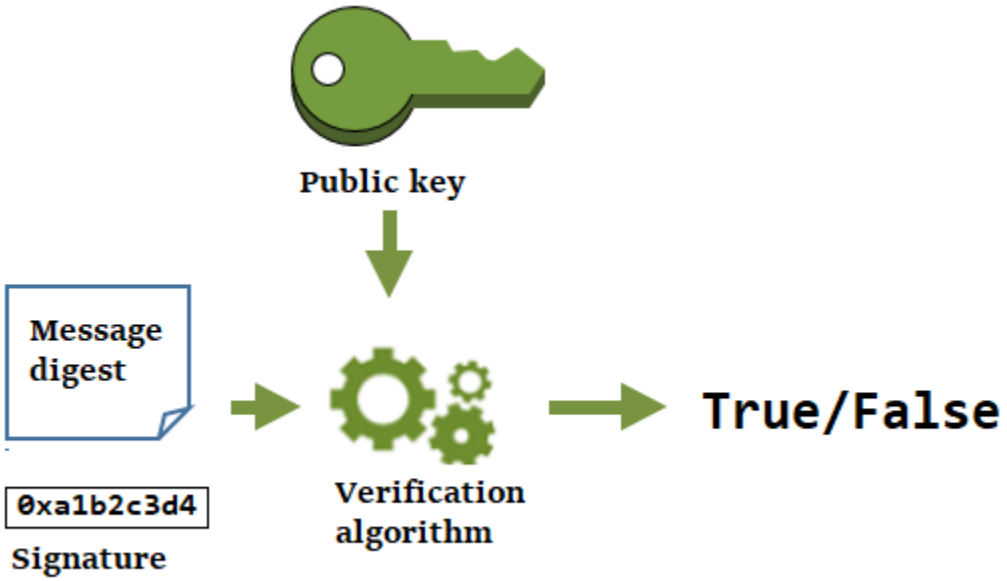


使用数据密钥对验证签名

持有数据密钥对中公有密钥的任何人，都可以使用该密钥来验证使用私有密钥生成的签名。验证确认已授权用户使用指定的私有密钥和签名算法签署了消息，并且消息自签名以后未曾更改。

要使验证成功，验证签名的一方必须生成相同类型的摘要，使用相同的算法，并使用与用于签署消息的私有密钥相对应的公有密钥。

下图显示了如何使用数据密钥对中的公有密钥验证消息签名。



使用数据密钥对派生共享密钥

密钥协议使两个各拥有一个椭圆曲线公有/私有密钥对的对等方能够在不安全的通道上创建共享密钥。要派生共享密钥，两个对等方必须通过不安全的通信渠道（如互联网）交换其公有密钥。然后，各方使用其“私有密钥”和对等方的“公有密钥”，利用密钥协议算法计算相同的共享密钥。您可以使用共享密钥值来派生对称密钥，该密钥可以加密和解密两个对等方之间发送的数据，或者可以生成和验证 HMAC。

> **Note**
>
> Amazon KMS 强烈建议在使用公有密钥派生共享密钥之前，先验证您收到的公有密钥是否来自预期方。

使用公有密钥执行离线操作

在非对称 KMS 密钥中，私有密钥是在 Amazon KMS 中创建的，它永远不会让 Amazon KMS 处于未加密状态。要使用私有密钥，必须调用 Amazon KMS。您可以通过调用 Amazon KMS API 操作在 Amazon KMS 内使用公有密钥。或者，可以下载公有密钥并共享以在 Amazon KMS 外部使用该密钥。

可以共享公有密钥，让其他人在 Amazon KMS 外部加密数据，但您只能用私有密钥解密数据。或者，允许其他人在 Amazon KMS 外部验证您使用私有密钥生成的数字签名。或者，与同事共享您的公有密钥以派生共享密钥。

在 Amazon KMS 内部使用非对称 KMS 密钥中的公有密钥时，您将从每个 Amazon KMS 操作中包含的身份验证、授权和日志记录中获益。您还可以降低对无法解密的数据进行加密的风险。这些功能在 Amazon KMS 外部无效。有关更多信息，请参阅 下载公有密钥的特殊注意事项。

> **Tip**
>
> 正在寻找数据密钥或 SSH 密钥？ 本主题介绍如何管理 Amazon Key Management Service 中的非对称密钥，私有密钥在其中不可导出。有关私有密钥在其中由对称加密 KMS 密钥保护的可导出数据密钥对，请参阅 GenerateDataKeyPair。有关下载与 Amazon EC2 实例关联的公有密钥的帮助，请参阅《Amazon EC2 用户指南》和《Amazon EC2 用户指南》中的检索公有密钥。

主题

下载公有密钥的特殊注意事项

为保护 KMS 密钥，Amazon KMS 提供了访问控制、身份验证加密和每个操作的详细日志。Amazon KMS 还允许您暂时或永久地阻止使用 KMS 密钥。最后，Amazon KMS 操作旨在最大限度降低对无法解密的数据进行加密的风险。在 Amazon KMS 外部使用下载的公有密钥时，这些功能不可用。

授权

在 Amazon KMS 内控制对 KMS 密钥访问的密钥政策和 IAM policy 对在 Amazon 外部执行的操作没有影响。任何可获得公有密钥的用户都可以在 Amazon KMS 外部使用该公有密钥，即使这些用户无权使用 KMS 密钥加密数据或验证签名。

密钥用法限制

密钥用法限制在 Amazon KMS 之外是无效的。如果您使用 KeyUsage 为 SIGN_VERIFY 的 KMS 密钥调用 Encrypt 操作，则 Amazon KMS 操作将失败。但是，如果您在 Amazon KMS 外部使用 SIGN_VERIFY 或 KEY_AGREEMENT 为 KeyUsage 的 KMS 密钥的公有密钥加密数据，数据将无法解密。

算法限制

对 Amazon KMS 支持的加密算法和签名算法的限制，在 Amazon KMS 之外是无效的。如果在 Amazon KMS 外部使用 KMS 密钥的公有密钥加密数据，并使用 Amazon KMS 不支持的加密算法，则数据将无法解密。

禁用和删除 KMS 密钥

为阻止在 Amazon KMS 内的加密操作中使用 KMS 密钥所能采取的措施，不会阻止任何人在 Amazon KMS 外部使用公有密钥。例如，禁用 KMS 密钥、调度删除 KMS 密钥、删除 KMS 密钥或删除 KMS 密钥的密钥材料，对 Amazon KMS 外部的公有密钥没有影响。如果删除非对称 KMS 密钥或者删除或丢失其密钥材料，那么在 Amazon KMS 外部使用公有密钥加密的数据将无法恢复。

日志记录

记录每个 Amazon CloudTrail 操作（包括请求、响应、日期、时间和授权用户）的 Amazon KMS 日志，不会记录 Amazon KMS 外部公有密钥的使用情况。

使用 SM2 密钥对进行离线验证（仅限中国区域）

要使用 SM2 公钥验证 Amazon KMS 外部的签名，您必须指定区分 ID。默认情况下，Amazon KMS 使用 1234567812345678 作为区分 ID。有关更多信息，请参阅使用 SM2 密钥对进行离线验证（仅限中国区域）。

下载公有密钥

您可以通过 Amazon KMS 控制台或使用 GetPublicKey 操作下载非对称 KMS 密钥对的公有密钥。要下载公有密钥，您必须具备对非对称 KMS 密钥的 kms:GetPublicKey 权限。

Amazon KMS 返回的公有密钥是 DER 编码的 X.509 公有密钥，也称为 `SubjectPublicKeyInfo`（SPKI），定义见 RFC 5280。使用 HTTP API 或 Amazon CLI 时，该值将采用 Base64 编码。否则，将不会采用 Base64 编码。

要下载非对称 KMS 密钥对的公有密钥，您必须具备 `kms:GetPublicKey` 权限。有关 Amazon KMS 权限的更多信息，请参阅权限参考。

使用 Amazon KMS 控制台

您可以使用 Amazon Web Services 管理控制台 查看、复制和下载 Amazon Web Services 账户 中的非对称 KMS 密钥的公有密钥。要在不同 Amazon Web Services 账户 中下载非对称 KMS 密钥的公有密钥，请使用 Amazon KMS API。

1. 登录到 Amazon Web Services 管理控制台，然后通过以下网址打开 Amazon Key Management Service（Amazon KMS）控制台：https://console.aws.amazon.com/kms。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择客户托管密钥。
4. 选择非对称 KMS 密钥的别名或密钥 ID。
5. 选择 Cryptographic configuration（加密配置）选项卡。记录密钥规范、密钥用法和加密算法或签名算法字段的值。您需要使用这些值才能在 Amazon KMS 外部使用公有密钥。在共享公有密钥时，请务必共享以上信息。
6. 选择 Public key (公有密钥) 选项卡。
7. 要将公有密钥复制到剪贴板，请选择 Copy (复制)。要将公有密钥下载到文件，请选择 Download (下载)。

使用 Amazon KMS API

GetPublickey 操作会返回非对称 KMS 密钥中的公有密钥。它还会返回在 Amazon KMS 外部正确使用公有密钥所需的关键信息，包括密钥用法和加密算法。请务必保存这些值，并在共享公有密钥时共享它们。

本部分中的示例使用 Amazon Command Line Interface (Amazon CLI)，但您可以使用任何受支持的编程语言。

要指定 KMS 密钥，请使用其密钥 ID、密钥 ARN、别名名称或别名 ARN。使用别名时，应加上 alias/ 前缀。要指定不同 Amazon Web Services 账户 中的 KMS 密钥，必须使用其密钥 ARN 或别名 ARN。

在运行此命令之前，请将示例别名替换为 KMS 密钥的有效标识符。要运行此命令，必须具备对 KMS 密钥的 kms:GetPublicKey 权限。

```
$ aws kms get-public-key --key-id alias/example_RSA_3072

{
    "KeySpec": "RSA_3072",
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyUsage": "ENCRYPT_DECRYPT",
    "EncryptionAlgorithms": [
        "RSAES_OAEP_SHA_1",
        "RSAES_OAEP_SHA_256"
    ],
    "PublicKey": "MIIBojANBgkqhkiG..."
}
```

离线操作示例

下载非对称 KMS 密钥对的公有密钥后，您可以与他人共享该密钥并使用它来执行离线操作。

记录每个 Amazon CloudTrail 操作（包括请求、响应、日期、时间和授权用户）的 Amazon KMS 日志，不会记录 Amazon KMS 外部公有密钥的使用情况。

本主题提供了离线操作示例，以及 Amazon KMS 为简化离线操作提供的工具的详细信息。

主题

- 离线派生共享密钥
- 使用 ML-DSA 密钥对进行离线验证
- 使用 SM2 密钥对进行离线验证（仅限中国区域）

离线派生共享密钥

您可以下载 ECC 密钥对的公有密钥以在离线操作中使用，即 Amazon KMS 外部的操作。

以下 OpenSSL 演练演示了一种使用 ECC KMS 密钥对的公有密钥和使用 OpenSSL 创建的私钥在 Amazon KMS 外派生共享密钥的方法。

1. 在 OpenSSL 中创建 ECC 密钥对，并准备好将其与 Amazon KMS 一起使用。

```
// Create an ECC key pair in OpenSSL and save the private key in
 openssl_ecc_key_priv.pem
export OPENSSL_CURVE_NAME="P-256"
export KMS_CURVE_NAME="ECC_NIST_P256"

export OPENSSL_KEY1_PRIV_PEM="openssl_ecc_key1_priv.pem"
openssl ecparam -name ${OPENSSL_CURVE_NAME} -genkey -out ${OPENSSL_KEY1_PRIV_PEM}


// Derive the public key from the private key
export OPENSSL_KEY1_PUB_PEM="openssl_ecc_key1_pub.pem"
openssl ec -in ${OPENSSL_KEY1_PRIV_PEM} -pubout -outform pem \
    -out ${OPENSSL_KEY1_PUB_PEM}

// View the PEM file containing the public key and extract the public key as a
// Base64 encoded string into OPENSSL_KEY1_PUB_BASE64 for use with Amazon KMS
export OPENSSL_KEY1_PUB_BASE64=`cat ${OPENSSL_KEY1_PUB_PEM} | \
    tee /dev/stderr | grep -v "PUBLIC KEY" | tr -d "\n"`
```

2. 在 Amazon KMS 中创建 ECC 密钥协议密钥对，并准备好将其与 OpenSSL 一起使用。

```
// Create a KMS key on the same curve as the key pair from step 1
// with a key usage of KEY_AGREEMENT
// Save its ARN in KMS_KEY1_ARN.
export KMS_KEY1_ARN=`aws kms create-key --key-spec ${KMS_CURVE_NAME} \
    --key-usage KEY_AGREEMENT | tee /dev/stderr | jq -r .KeyMetadata.Arn`

// Download the public key and save the Base64-encoded version in KMS_KEY1_PUB_BASE64

export KMS_KEY1_PUB_BASE64=`aws kms get-public-key --key-id ${KMS_KEY1_ARN} | \
    tee /dev/stderr | jq -r .PublicKey`

// Create a PEM file for the public KMS key for use with OpenSSL
export KMS_KEY1_PUB_PEM="aws_kms_ecdh_key1_pub.pem"
echo "-----BEGIN PUBLIC KEY-----" > ${KMS_KEY1_PUB_PEM}
echo ${KMS_KEY1_PUB_BASE64} | fold -w 64 >> ${KMS_KEY1_PUB_PEM}
echo "-----END PUBLIC KEY-----" >> ${KMS_KEY1_PUB_PEM}
```

3. 使用 OpenSSL 中的私有密钥和 KMS 公有密钥在 OpenSSL 中派生共享密钥。

```
export OPENSSL_SHARED_SECRET1_BIN="openssl_shared_secret1.bin"
openssl pkeyutl -derive -inkey ${OPENSSL_KEY1_PRIV_PEM} \
```

```
    -peerkey ${KMS_KEY1_PUB_PEM} -out ${OPENSSL_SHARED_SECRET1_BIN}
```

使用 ML-DSA 密钥对进行离线验证

Amazon KMS 支持 ML-DSA 签名的对冲变体，详见联邦信息处理标准（FIPS）204 标准第 3.4 节所述，适用于不超过 4 KB 字节的消息。

要对大于 4 KB 的消息进行签名，请在 Amazon KMS 之外执行消息预处理步骤。此哈希化步骤会创建一条 64 字节的消息表示值 μ，详见 NIST FIPS 204 第 6.2 节所定义。

对于大于 4 KB 的消息，Amazon KMS 提供了一种名为 `EXTERNAL_MU` 的消息类型。使用此消息类型而不是 `RAW` 消息类型时，Amazon KMS 会：

- 假定您已经完成了哈希化步骤
- 跳过其内部哈希化过程
- 支持任何大小的消息

验证消息时所使用的方法取决于外部系统或库的大小限制，以及 64 字节消息表示值 μ 是否受支持：

- 如果消息小于大小限制，请使用 `RAW` 消息类型。
- 如果消息大于大小限制，请使用外部系统中的表示值 μ。

以下各节说明了如何使用 Amazon KMS 对消息进行签名以及如何使用 OpenSSL 验证消息。对于小于和大于 Amazon KMS 所设定 4 KB 消息大小限制的消息，我们都提供了示例。OpenSSL 并未对用于验证的消息实施大小限制。

这两个示例都会首先从 Amazon KMS 中获取公有密钥。使用以下 Amazon CLI 命令：

```
aws kms get-public-key \
    --key-id _<1234abcd-12ab-34cd-56ef-1234567890ab>_ \
    --output text \
    --query PublicKey | base64 --decode > public_key.der
```

消息大小小于 4KB

对于小于 4 KB 的消息，请将 `RAW` 消息类型用于 Amazon KMS。尽管可以使用 `EXTERNAL_MU`，但这对于在大小限制范围内的消息没有必要。

使用以下 Amazon CLI 命令对消息进行签名：

```
aws kms sign \
    --key-id _<1234abcd-12ab-34cd-56ef-1234567890ab>_ \
    --message 'your message' \
    --message-type RAW \
    --signing-algorithm ML_DSA_SHAKE_256 \
    --output text \
    --query Signature | base64 --decode > ExampleSignature.bin
```

要使用 OpenSSL 验证此消息，请使用以下命令：

```
echo -n 'your message' | ./openssl dgst -verify public_key.der -signature
 ExampleSignature.bin
```

消息大小大于 4KB

要对大于 4KB 的消息进行签名，请使用 EXTERNAL_MU 消息类型。使用 EXTERNAL_MU 时，您可以按照 NIST FIPS 204 第 6.2 节的定义，在外部将消息预哈希处理为 64 字节表示值 μ，然后将其传递给签名或验证操作。请注意，这不同于 NIST FIPS 204 第 5.4 节定义的“预哈希 ML-DSA”或 HashML-DSA。

1. 首先构造一个消息前缀。该前缀将包含一个域分隔符、任何上下文的长度以及上下文。域分隔符和上下文长度的默认值均为零。
2. 在消息前添加消息前缀。
3. 使用 SHAKE256 对公有密钥进行哈希处理，然后添加到第 2 步的结果之前。
4. 最后，对第 3 步的结果进行哈希处理以生成 64 字节 EXTERNAL_MU。

以下示例使用 OpenSSL 3.5 来构造 EXTERNAL_MU：

```
{
    openssl asn1parse -inform DER -in public_key.der -strparse 17 -noout -out - 2>/dev/
null |
    openssl dgst -provider default -shake256 -xoflen 64 -binary;
    printf '\x00\x00';
    echo -n "your message"
} | openssl dgst -provider default -shake256 -xoflen 64 -binary > mu.bin
```

创建 mu.bin 文件后，使用以下命令调用 Amazon KMS API 对消息进行签名：

```
aws kms sign \
    --key-id _<1234abcd-12ab-34cd-56ef-1234567890ab>_ \
    --message fileb://mu.bin \
    --message-type EXTERNAL_MU \
    --signing-algorithm ML_DSA_SHAKE_256 \
    --output text \
    --query Signature | base64 --decode > ExampleSignature.bin
```

生成的签名与原始消息上的 RAW 签名相同。可以使用相同的 OpenSSL 3.5 命令来验证消息：

```
echo -n 'your message' | ./openssl dgst -verify public_key.der -signature
 ExampleSignature.bin
```

使用 SM2 密钥对进行离线验证（仅限中国区域）

要使用 SM2 公有密钥验证 Amazon KMS 外部的签名，您必须指定区分 ID。将原始消息 `MessageType:RAW` 传递到签名 API 时，Amazon KMS 将使用 OSCCA 在 GM/T 0009-2012 中定义的默认区分 ID 1234567812345678。您不能在 Amazon KMS 中指定自己的区分 ID。

但是，如果您在 Amazon 外部生成消息摘要，可以指定自己的区分 ID，然后将消息摘要 `MessageType:DIGEST` 传递到 Amazon KMS 以进行签名。要执行此操作，请更改 `SM2OfflineOperationHelper` 类中的 `DEFAULT_DISTINGUISHING_ID` 值。您指定的区分 ID 可以是长度不超过 8192 个字符的任何字符串。在 Amazon KMS 为消息摘要签名后，您需要消息摘要或消息以及用于计算摘要的区分 ID 以进行离线验证。

> **Important**
>
> `SM2OfflineOperationHelper` 参考代码旨在兼容 Bouncy Castle 版本 1.68。如需其他版本的帮助，请联系 bouncycastle.org。

`SM2OfflineOperationHelper` 类

为了帮助您使用 SM2 密钥进行离线操作，Java 的 `SM2OfflineOperationHelper` 类为您执行任务提供了方法。您可以将此帮助程序类用作其他加密提供程序的模型。

在 Amazon KMS 内，原始加密文字转换和 SM2DSA 消息摘要计算会自动进行。并非所有加密提供程序都按相同的方式实现 SM2。有些库（比如 OpenSSL 版本 1.1.1 和更高版本）会自动执行这些操作。Amazon KMS 在使用 OpenSSL 版本 3.0 进行测试时确认了此行为。使用以下带有库的 `SM2OfflineOperationHelper` 类，比如 Bouncy Castle，这需要您手动执行这些转换和计算。

SM2OfflineOperationHelper 类为以下离线操作提供了方法：

- 消息摘要计算

  要离线生成可用于离线验证或传递到 Amazon KMS 以进行签名的消息摘要，请使用 calculateSM2Digest 方法。calculateSM2Digest 方法通过 SM3 哈希算法生成消息摘要。GetPublicKey API 以二进制格式返回您的公有密钥。您必须把二进制密钥解析为 Java PublicKey。提供解析后的公有密钥与消息。该方法会自动将您的消息与默认的区分 ID 1234567812345678 相结合，但您可以通过更改 DEFAULT_DISTINGUISHING_ID 值来设置自己的区分 ID。

- 验证

  要离线验证签名，请使用 offlineSM2DSAVerify 方法。offlineSM2DSAVerify 方法会使用根据指定区分 ID 计算出的消息摘要和您提供的原始消息来验证数字签名。GetPublicKey API 以二进制格式返回您的公有密钥。您必须把二进制密钥解析为 Java PublicKey。向解析后的公有密钥提供原始消息和要验证的签名。有关更多详细信息，请参阅使用 SM2 密钥对进行离线验证。

- Encrypt

  要离线加密明文，请使用 offlineSM2PKEEncrypt 方法。此方法可确保加密文字采用某种 Amazon KMS 可以解密的格式。offlineSM2PKEEncrypt 方法会对明文进行加密，然后将 SM2PKE 生成的原始加密文字转换为 ASN.1 格式。GetPublicKey API 以二进制格式返回您的公有密钥。您必须把二进制密钥解析为 Java PublicKey。为解析后的公有密钥提供您想要加密的明文。

  如果您不确定是否需要进行转换，请使用以下 OpenSSL 操作来测试加密文字的格式。如果操作失败，则需要将加密文字转换为 ASN.1 格式。

  ```
  openssl asn1parse -inform DER -in ciphertext.der
  ```

默认情况下，在为 SM2DSA 操作生成消息摘要时，SM2OfflineOperationHelper 类使用默认的区分 ID 1234567812345678。

```
package com.amazon.kms.utils;

import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
```

```
import javax.crypto.NoSuchPaddingException;
import java.io.IOException;
import java.math.BigInteger;
import java.nio.ByteBuffer;
import java.nio.charset.StandardCharsets;
import java.security.InvalidKeyException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.security.NoSuchProviderException;
import java.security.PrivateKey;
import java.security.PublicKey;

import org.bouncycastle.crypto.CryptoException;
import org.bouncycastle.jce.interfaces.ECPublicKey;

import java.util.Arrays;

import org.bouncycastle.asn1.ASN1EncodableVector;
import org.bouncycastle.asn1.ASN1Integer;
import org.bouncycastle.asn1.DEROctetString;
import org.bouncycastle.asn1.DERSequence;
import org.bouncycastle.asn1.gm.GMNamedCurves;
import org.bouncycastle.asn1.x9.X9ECParameters;
import org.bouncycastle.crypto.CipherParameters;
import org.bouncycastle.crypto.params.ParametersWithID;
import org.bouncycastle.crypto.params.ParametersWithRandom;
import org.bouncycastle.crypto.signers.SM2Signer;
import org.bouncycastle.jcajce.provider.asymmetric.util.ECUtil;

public class SM2OfflineOperationHelper {
    // You can change the DEFAULT_DISTINGUISHING_ID value to set your own
 distinguishing ID,
    // the DEFAULT_DISTINGUISHING_ID can be any string up to 8,192 characters long.
    private static final byte[] DEFAULT_DISTINGUISHING_ID =
 "1234567812345678".getBytes(StandardCharsets.UTF_8);
    private static final X9ECParameters SM2_X9EC_PARAMETERS =
 GMNamedCurves.getByName("sm2p256v1");

    // ***calculateSM2Digest***
    // Calculate message digest
    public static byte[] calculateSM2Digest(final PublicKey publicKey, final byte[]
 message) throws
            NoSuchProviderException, NoSuchAlgorithmException {
        final ECPublicKey ecPublicKey = (ECPublicKey) publicKey;
```

```
        // Generate SM3 hash of default distinguishing ID, 1234567812345678
        final int entlenA = DEFAULT_DISTINGUISHING_ID.length * 8;
        final byte [] entla = new byte[] { (byte) (entlenA & 0xFF00), (byte) (entlenA &
0x00FF) };
        final byte [] a = SM2_X9EC_PARAMETERS.getCurve().getA().getEncoded();
        final byte [] b = SM2_X9EC_PARAMETERS.getCurve().getB().getEncoded();
        final byte [] xg = SM2_X9EC_PARAMETERS.getG().getXCoord().getEncoded();
        final byte [] yg = SM2_X9EC_PARAMETERS.getG().getYCoord().getEncoded();
        final byte[] xa = ecPublicKey.getQ().getXCoord().getEncoded();
        final byte[] ya = ecPublicKey.getQ().getYCoord().getEncoded();
        final byte[] za = MessageDigest.getInstance("SM3", "BC")
                .digest(ByteBuffer.allocate(entla.length +
DEFAULT_DISTINGUISHING_ID.length + a.length + b.length + xg.length + yg.length +
                        xa.length +
ya.length).put(entla).put(DEFAULT_DISTINGUISHING_ID).put(a).put(b).put(xg).put(yg).put(xa).put
                        .array());

        // Combine hashed distinguishing ID with original message to generate final
digest
        return MessageDigest.getInstance("SM3", "BC")
                .digest(ByteBuffer.allocate(za.length +
message.length).put(za).put(message)
                        .array());
    }

    // ***offlineSM2DSAVerify***
    // Verify digital signature with SM2 public key
    public static boolean offlineSM2DSAVerify(final PublicKey publicKey, final byte []
message,
            final byte [] signature) throws InvalidKeyException {
        final SM2Signer signer = new SM2Signer();
        CipherParameters cipherParameters =
ECUtil.generatePublicKeyParameter(publicKey);
        cipherParameters = new ParametersWithID(cipherParameters,
DEFAULT_DISTINGUISHING_ID);
        signer.init(false, cipherParameters);
        signer.update(message, 0, message.length);
        return signer.verifySignature(signature);
    }

    // ***offlineSM2PKEEncrypt***
    // Encrypt data with SM2 public key
```

```
    public static byte[] offlineSM2PKEEncrypt(final PublicKey publicKey, final byte []
 plaintext) throws
            NoSuchPaddingException, NoSuchAlgorithmException, NoSuchProviderException,
 InvalidKeyException,
            BadPaddingException, IllegalBlockSizeException, IOException {
        final Cipher sm2Cipher = Cipher.getInstance("SM2", "BC");
        sm2Cipher.init(Cipher.ENCRYPT_MODE, publicKey);

        // By default, Bouncy Castle returns raw ciphertext in the c1c2c3 format
        final byte [] cipherText = sm2Cipher.doFinal(plaintext);

        // Convert the raw ciphertext to the ASN.1 format before passing it to AWS KMS
        final ASN1EncodableVector asn1EncodableVector = new ASN1EncodableVector();
        final int coordinateLength = (SM2_X9EC_PARAMETERS.getCurve().getFieldSize() +
 7) / 8 * 2 + 1;
        final int sm3HashLength = 32;
        final int xCoordinateInCipherText = 33;
        final int yCoordinateInCipherText = 65;
        byte[] coords = new byte[coordinateLength];
        byte[] sm3Hash = new byte[sm3HashLength];
        byte[] remainingCipherText = new byte[cipherText.length - coordinateLength -
 sm3HashLength];

        // Split components out of the ciphertext
        System.arraycopy(cipherText, 0, coords, 0, coordinateLength);
        System.arraycopy(cipherText, cipherText.length - sm3HashLength, sm3Hash, 0,
 sm3HashLength);
        System.arraycopy(cipherText, coordinateLength, remainingCipherText,
 0,cipherText.length - coordinateLength - sm3HashLength);

        // Build standard SM2PKE ASN.1 ciphertext vector
        asn1EncodableVector.add(new ASN1Integer(new BigInteger(1,
 Arrays.copyOfRange(coords, 1, xCoordinateInCipherText))));
        asn1EncodableVector.add(new ASN1Integer(new BigInteger(1,
 Arrays.copyOfRange(coords, xCoordinateInCipherText, yCoordinateInCipherText))));
        asn1EncodableVector.add(new DEROctetString(sm3Hash));
        asn1EncodableVector.add(new DEROctetString(remainingCipherText));

        return new DERSequence(asn1EncodableVector).getEncoded("DER");
    }
}
```

监控 Amazon KMS keys

对于了解 Amazon KMS 中 Amazon KMS keys 的可用性、状态和使用情况，以及维护 Amazon 解决方案的可靠性、可用性和性能，监控是一个重要部分。从 Amazon 解决方案的各个部分收集监控数据将有助于调试出现的多点故障。不过，在开始监控 KMS 密钥之前，应制定一个监控计划并在计划中回答下列问题：

- 监控目的是什么？
- 您将监控哪些资源？
- 监控这些资源的频率如何？
- 您将使用哪些监控工具？
- 谁负责执行监控任务？
- 出现情况时应通知谁？

下一步是监控 KMS 密钥在一段时间内的情况，以便为环境中正常的 Amazon KMS 使用情况和预期建立基准。监控 KMS 密钥时，存储历史监控数据，以便将此数据与当前数据进行比较，从而确定正常模式和异常情况，并找出解决问题的方法。

例如，您可以监控影响 KMS 密钥的 Amazon KMS API 活动和事件。当数据高于或低于既定标准时，您可能需要进行调查或采取纠正措施。

要建立正常模式的基准，应监控以下各项：

- Amazon KMS数据层面操作的 API 活动。这些是使用 KMS 密钥的加密操作，例如 Decrypt、Encrypt、ReEncrypt 和 GenerateDataKey。
- 对您而言至关重要的Amazon KMS控制层面操作的 API 活动。这些操作用于管理 KMS 密钥，您可能需要监控更改 KMS 密钥可用性的操作（如 ScheduleKeyDeletion、CancelKeyDeletion、DisableKey、EnableKey、ImportKeyMaterial 和 DeleteImportedKeyMaterial）或更改 KMS 密钥访问控制的操作（如 PutKeyPolicy 和 RevokeGrant）。
- 其他 Amazon KMS 指标（如导入的密钥资料过期之前的剩余时间量）和事件（如导入的密钥资料到期，KMS 密钥的删除或密钥轮换）。

监控工具

Amazon 为您提供了可用来监控 KMS 密钥的各种工具。您可以配置其中的一些工具来为您执行监控任务，但有些工具需要手动干预。建议您尽可能实现监控任务自动化。

自动监控工具

您可以使用以下自动化监控工具来监控 KMS 密钥并在发生更改时进行报告。

- Amazon CloudTrail 日志监控 – 在账户间共享日志文件，通过将 CloudTrail 日志文件发送到 CloudWatch Logs 对它们进行实时监控，使用 CloudTrail Processing Library 编写日志处理应用程序，以及验证您的日志文件在被 CloudTrail 交付后未发生更改。有关更多信息，请参见《Amazon CloudTrail 用户指南》的使用 CloudTrail 日志文件。
- Amazon CloudWatch 警报——按您指定的时间段观察单个指标，并根据相对于给定阈值的指标值在若干时间段内执行一项或多项操作。具体操作是将一条通知已发送到某个 Amazon Simple Notification Service（Amazon SNS）主题或 Amazon EC2 Auto Scaling 策略。CloudWatch 告警不调用操作，因为这些操作处于特定状态；状态必须改变并保持指定时间。有关更多信息，请参阅 使用 Amazon CloudWatch 监控 KMS 密钥。
- Amazon EventBridge – 匹配事件并将事件传送到一个或多个目标函数或流来捕获状态信息，并在必要时进行更改或采取纠正措施。有关更多信息，请参阅 使用 Amazon EventBridge 来监控 KMS 密钥 和 Amazon EventBridge 用户指南。
- Amazon CloudWatch Logs – 监控、存储和访问来自 Amazon CloudTrail 或其他来源的日志文件。有关更多信息，请参阅 Amazon CloudWatch Logs 用户指南。

手动监控工具

监控 KMS 密钥的另一个重要环节是手动监控 CloudWatch 告警和事件未涵盖的那些项。Amazon KMS、CloudWatch、Amazon Trusted Advisor 和其他 Amazon 控制面板提供您的 Amazon 环境状态的概览视图。

您可以自定义 Amazon KMS 控制台的 Amazon 托管式密钥和客户自主管理型密钥页面，从而显示有关每个 KMS 密钥的以下信息：

- 密钥 ID
- 状态
- Creation date (创建日期)

- 到期日期（对于具有导入密钥材料的 KMS 密钥）
- Origin
- 自定义密钥存储 ID（对于自定义密钥存储中的 KMS 密钥）

CloudWatch 控制台控制面板显示以下信息：

- 当前告警和状态
- 告警和资源图表
- 服务运行状况

此外，还可以使用 CloudWatch 执行以下操作：

- 创建自定义控制面板以监控您关心的服务
- 绘制指标数据图，以排除问题并弄清楚趋势
- 搜索并浏览您所有的 Amazon 资源指标
- 创建和编辑警报以接收有关问题的通知

Amazon Trusted Advisor 可以帮助您监控 Amazon 资源以提高性能、可靠性、安全性和成本效益。四个 Trusted Advisor 检查可供所有用户使用；超过 50 个检查可供具有“商业”或“企业”支持计划的用户使用。有关更多信息，请参阅 Amazon Trusted Advisor。

使用 Amazon CloudTrail 记录 Amazon KMS API 调用

Amazon KMS 与 Amazon CloudTrail 进行了集成，后者是一项纪录用户、角色和其他 Amazon 服务对 Amazon KMS 进行的所有调用的服务。CloudTrail 将对 Amazon KMS 的所有 API 调用均作为事件捕获，包括来自 Amazon KMS 控制台、Amazon KMS API、Amazon CloudFormation 模板、Amazon Command Line Interface（Amazon CLI）和 Amazon Tools for PowerShell 的调用。

CloudTrail 记录所有 Amazon KMS 操作（包括只读操作），如 ListAliases 和 GetKeyRotationStatus，管理 KMS 密钥的操作，例如 CreateKey 和 PutKeyPolicy，以及加密操作，例如 GenerateDataKey 和 Decrypt。它还记录 Amazon KMS 为您调用的内部操作，例如 DeleteExpiredKeyMaterial、DeleteKey、SynchronizeMultiRegionKey 和 RotateKey。

CloudTrail 会记录所有成功的操作，在某些情况下还会记录失败的调用尝试，例如当调用方被拒绝访问某个资源时。在 KMS 密钥上的跨账户操作将记录在调用方的账户和 KMS 密钥所有者账户中。但是，因访问被拒绝而受到拒绝的跨账户 Amazon KMS 请求仅记录在发起人的账户中。

出于安全原因，有些字段将从 Amazon KMS 日志条目中省略，例如 Encrypt 请求的 `Plaintext` 参数，以及对 GetKeyPolicy 或任何加密操作的响应。为了更轻松地搜索特定 KMS 密钥的 CloudTrail 日志条目，即使 API 操作没有返回密钥 ARN，Amazon KMS 也会将受影响的 KMS 密钥的密钥 ARN 添加到某些 Amazon KMS 密钥管理操作的日志条目中的 `responseElements` 字段中。

尽管默认情况下，所有 Amazon KMS 操作都记录为 CloudTrail 事件，但您可以从 CloudTrail 跟踪中排除 Amazon KMS 操作。有关更多信息，请参阅 从跟踪中排除 Amazon KMS 事件。

了解更多

- 有关认证平台的 Amazon KMS 操作 CloudTrail 日志示例，请参阅监控认证请求。

主题

在 CloudTrail 中查找 Amazon KMS 日志条目

要搜索 CloudTrail 日志条目，请使用 CloudTrail 控制台或 CloudTrail LookupEvents 操作。CloudTrail 支持多种属性值来筛选您的搜索，包括事件名称、用户名和事件源。

为了帮助您在 CloudTrail 中搜索 Amazon KMS 日志条目，Amazon KMS 会填充以下 CloudTrail 日志条目字段。

> **Note**
>
> 从 2022 年 12 月开始，在更改特定 KMS 密钥的所有管理操作中，Amazon KMS 会填充资源类型和资源名称属性。在以下操作的较早的 CloudTrail 条目中，这些属性值可能为空：CreateAlias、CreateGrant、DeleteAlias、DeleteImportedKeyMaterial、ImportKeyMaterial、Replic
> 和 UpdatePrimaryRegion。

属性	值	日志条目
事件源（`EventSource`）	`kms.amazonaws.com`	全部操作。

属性	值	日志条目
资源类型（ResourceType）	AWS::KMS::Key	更改特定 KMS 密钥的管理操作，例如 CreateKey 和 EnableKey，而非 ListKeys。
资源名称（ResourceName）	密钥 ARN（或密钥 ID 和密钥 ARN）	更改特定 KMS 密钥的管理操作，例如 CreateKey 和 EnableKey，而非 ListKeys。

为了帮助您查找对特定 KMS 密钥进行管理操作的日志条目，即使 Amazon KMS API 操作没有返回密钥 ARN，Amazon KMS 也会将受影响的 KMS 密钥的密钥 ARN 记录在日志条目的 responseElements.keyId 元素中。

例如，成功调用 DisableKey 操作不会在响应中返回任何值，但是在 DisableKey 日志条目中，responseElements.keyId 值，而非空值，包含禁用的 KMS 密钥的密钥 ARN。

此功能添加于 2022 年 12 月，会影响以下 CloudTrail 日志条目：CreateAlias、CreateGrant、DeleteAlias、DeleteKey、DisableKey、EnableKey、EnableKeyRotation、I 和 UpdatePrimaryRegion。

从跟踪中排除 Amazon KMS 事件

大多数 Amazon KMS 用户依靠 CloudTrail 跟踪中的事件来提供其 Amazon KMS 资源的使用和管理记录。跟踪可以成为审核关键事件的宝贵数据来源，例如，创建、禁用和删除 Amazon KMS keys，更改密钥策略以及 Amazon 服务代表您使用 KMS 密钥。在某些情况下，CloudTrail 日志条目中的元数据（如加密操作中的加密上下文）可以帮助您避免或解决错误。

但是，由于 Amazon KMS 可以生成大量事件，因此 Amazon CloudTrail 允许您从跟踪中排除 Amazon KMS 事件。此按跟踪设置会排除所有 Amazon KMS 事件。您不能排除特定 Amazon KMS 事件。

> **Warning**
>
> 从 CloudTrail 日志中排除 Amazon KMS 事件可能会掩盖使用 KMS 密钥的操作。请谨慎赋予委托人执行此操作所需的 cloudtrail:PutEventSelectors 权限。

要从跟踪中排除 Amazon KMS 事件，请执行以下操作：

- 在 CloudTrail 控制台中，在您创建跟踪或者更新跟踪时使用记录密钥管理服务事件设置。有关说明，请参阅 Amazon CloudTrail 用户指南中的使用 Amazon Web Services 管理控制台 记录管理事件。
- 在 CloudTrail API 中，使用 PutEventSelectors 操作。将 `ExcludeManagementEventSources` 属性添加到值为 `kms.amazonaws.com` 的事件选择器中。有关示例，请参阅 Amazon CloudTrail 用户指南中的示例：不记录 Amazon Key Management Service 事件的跟踪。

您可以随时更改控制台设置或跟踪的事件选择器，以禁用此排除。随后，跟踪将开始记录 Amazon KMS 事件。但是，它无法恢复在排除生效期间发生的 Amazon KMS 事件。

使用控制台或 API 排除 Amazon KMS 事件时，生成的 CloudTrail `PutEventSelectors` API 操作也会记录在 CloudTrail 日志中。如果 Amazon KMS 事件未显示在 CloudTrail 日志中，请查找 `ExcludeManagementEventSources` 属性设置为 `kms.amazonaws.com` 的 `PutEventSelectors` 事件。

Amazon KMS 日志条目的示例

Amazon KMS 在您调用 Amazon KMS 操作且 Amazon 服务代表您调用操作时将条目写入您的 CloudTrail 日志中。Amazon KMS 也会在为您调用操作时写入一个条目。例如，它会在删除您计划删除的 KMS 密钥时写入条目。

以下主题显示 Amazon KMS 操作的 CloudTrail 日志条目的示例。

有关从认证平台向 Amazon KMS 发出的请求的 CloudTrail 日志条目示例，请参阅监控认证请求。

主题

- CancelKeyDeletion
- ConnectCustomKeyStore
- CreateAlias
- CreateCustomKeyStore
- CreateGrant
- CreateKey
- Decrypt
- DeleteAlias
- DeleteCustomKeyStore

- DeleteExpiredKeyMaterial
- DeleteImportedKeyMaterial
- DeleteKey
- DescribeCustomKeyStores
- DescribeKey
- DisableKey
- DisableKeyRotation
- DisconnectCustomKeyStore
- EnableKey
- EnableKeyRotation
- Encrypt
- GenerateDataKey
- GenerateDataKeyPair
- GenerateDataKeyPairWithoutPlaintext
- GenerateDataKeyWithoutPlaintext
- GenerateMac
- GenerateRandom
- GetKeyPolicy
- GetKeyRotationStatus
- GetParametersForImport
- ImportKeyMaterial
- ListAliases
- ListGrants
- ListKeyRotations
- PutKeyPolicy
- ReEncrypt
- ReplicateKey
- RetireGrant
- RevokeGrant
- RotateKey

- RotateKeyOnDemand
- ScheduleKeyDeletion
- Sign
- SynchronizeMultiRegionKey
- TagResource
- UntagResource
- UpdateAlias
- UpdateCustomKeyStore
- UpdateKeyDescription
- UpdatePrimaryRegion
- VerifyMac
- 验证
- Amazon EC2 示例一
- Amazon EC2 示例二

CancelKeyDeletion

以下示例显示了通过调用 CancelKeyDeletion 操作生成的 Amazon CloudTrail 日志条目。关于删除 Amazon KMS keys 的信息，请查阅 删除 Amazon KMS key。


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-07-27T21:53:17Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CancelKeyDeletion",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
```

```
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "e3452e68-d4b0-4ec7-a768-7ae96c23764f",
    "eventID": "d818bf03-6655-48e9-8b26-f279a07075fd",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

ConnectCustomKeyStore

以下示例显示了通过调用 ConnectCustomKeyStore 操作生成的 Amazon CloudTrail 日志条目。有关连接自定义密钥存储的信息，请参阅 断开 Amazon CloudHSM 密钥存储。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2021-10-21T20:17:32Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ConnectCustomKeyStore",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
```

```
        "customKeyStoreId": "cks-1234567890abcdef0"
    },
    "responseElements": null,
    "additionalEventData": {
        "customKeyStoreName": "ExampleKeyStore",
        "clusterId": "cluster-1a23b4cdefg"
    },
    "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
    "eventID": "114b61b9-0ea6-47f5-a9d2-4f2bdd0017d5",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
}
```

CreateAlias

以下示例显示了 CreateAlias 操作的一个 Amazon CloudTrail 日志条目。`resources` 元素包含别名和 KMS 密钥资源的字段。有关在 Amazon KMS 中创建别名的信息，请参阅 创建别名。

在 2022 年 12 月或之后记录的此操作的 CloudTrail 日志条目将受影响的 KMS 密钥的密钥 ARN 包含在 `responseElements.keyId` 值中，即使此操作未返回密钥 ARN。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-08-14T23:08:31Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateAlias",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "aliasName": "alias/ExampleAlias",
        "targetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
```

```
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "caec1e0c-ce03-419e-bdab-6ab1f7c57c01",
    "eventID": "2dd6e784-8286-46a6-befd-d64e5a02fb28",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        },
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:alias/ExampleAlias"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

CreateCustomKeyStore

以下示例显示了通过调用 Amazon CloudHSM 密钥存储上的 `CreateCustomKeyStore` 操作生成的 Amazon CloudTrail 日志条目。有关创建自定义密钥存储的信息，请参阅 创建 Amazon CloudHSM 密钥存储。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2021-10-21T20:17:32Z",
```

```
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateCustomKeyStore",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "customKeyStoreName": "ExampleKeyStore",
        "clusterId": "cluster-1a23b4cdefg"
    },
    "responseElements": {
        "customKeyStoreId": "cks-1234567890abcdef0"
    },
    "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
    "eventID": "114b61b9-0ea6-47f5-a9d2-4f2bdd0017d5",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
}
```

CreateGrant

以下示例显示了 CreateGrant 操作的一个 Amazon CloudTrail 日志条目。有关在 Amazon KMS 中创建授权的信息，请参阅 Amazon KMS 中的授权。

在 2022 年 12 月或之后记录的此操作的 CloudTrail 日志条目将受影响的 KMS 密钥的密钥 ARN 包含在 `responseElements.keyId` 值中，即使此操作未返回密钥 ARN。

```
{
  "eventVersion": "1.02",
  "userIdentity": {
      "type": "IAMUser",
      "principalId": "EX_PRINCIPAL_ID",
      "arn": "arn:aws:iam::111122223333:user/Alice",
      "accountId": "111122223333",
      "accessKeyId": "EXAMPLE_KEY_ID",
      "userName": "Alice"
  },
  "eventTime": "2014-11-04T00:53:12Z",
  "eventSource": "kms.amazonaws.com",
  "eventName": "CreateGrant",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.2.0",
```

```
  "userAgent": "Amazon Internal",
  "requestParameters": {
      "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "constraints": {
          "encryptionContextSubset": {
              "ContextKey1": "Value1"
          }
      },
      "operations": ["Encrypt",
      "RetireGrant"],
      "granteePrincipal": "EX_PRINCIPAL_ID"
  },
  "responseElements": {
      "grantId": "f020fe75197b93991dc8491d6f19dd3cebb24ee62277a05914386724f3d48758",
      "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  },
  "requestID": "f3c08808-63bc-11e4-bc2b-4198b6150d5c",
  "eventID": "5d529779-2d27-42b5-92da-91aaea1fc4b5",
  "readOnly": false,
  "resources": [{
      "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
      "accountId": "111122223333"
  }],
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

CreateKey

这些示例显示 CreateKey 操作的 Amazon CloudTrail 日志条目。

CreateKey 日志条目可以从 CreateKey 请求或 ReplicateKey 请求的 CreateKey 操作中生成。

以下示例显示了一个创建对称加密 KMS 密钥的 CreateKey 操作的一个 CloudTrail 日志条目。有关创建 KMS 密钥的信息，请参阅 创建 KMS 密钥。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
```

```
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-08-10T22:38:27Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "description": "",
        "origin": "EXTERNAL",
        "bypassPolicyLockoutSafetyCheck": false,
        "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "keySpec": "SYMMETRIC_DEFAULT",
        "keyUsage": "ENCRYPT_DECRYPT"
    },
    "responseElements": {
        "keyMetadata": {
            "AWSAccountId": "111122223333",
            "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "creationDate": "Aug 10, 2022, 10:38:27 PM",
            "enabled": false,
            "description": "",
            "keyUsage": "ENCRYPT_DECRYPT",
            "keyState": "PendingImport",
            "origin": "EXTERNAL",
            "keyManager": "CUSTOMER",
            "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
            "keySpec": "SYMMETRIC_DEFAULT",
            "encryptionAlgorithms": [
                "SYMMETRIC_DEFAULT"
            ],
            "multiRegion": false
        }
    },
    "requestID": "1aef6713-0223-4ff7-9a6d-781360521930",
    "eventID": "36327b37-f4f6-40a9-92ab-48064ec905a2",
    "readOnly": false,
```

```
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

以下示例显示了一个在 Amazon CloudHSM 密钥存储中创建对称加密 KMS 密钥的 `CreateKey` 操作的 CloudTrail 日志。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2021-10-14T17:39:50Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "keyUsage": "ENCRYPT_DECRYPT",
        "bypassPolicyLockoutSafetyCheck": false,
        "origin": "AWS_CLOUDHSM",
        "keySpec": "SYMMETRIC_DEFAULT",
        "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "customKeyStoreId": "cks-1234567890abcdef0",
        "description": ""
    },
    "responseElements": {
```

```
        "keyMetadata": {
            "aWSAccountId": "111122223333",
            "keyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "arn": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
            "creationDate": "Oct 14, 2021, 5:39:50 PM",
            "enabled": true,
            "description": "",
            "keyUsage": "ENCRYPT_DECRYPT",
            "keyState": "Enabled",
            "origin": "AWS_CLOUDHSM",
            "customKeyStoreId": "cks-1234567890abcdef0",
            "cloudHsmClusterId": "cluster-1a23b4cdefg",
            "keyManager": "CUSTOMER",
            "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
            "keySpec": "SYMMETRIC_DEFAULT",
            "encryptionAlgorithms": [
                "SYMMETRIC_DEFAULT"
            ],
            "multiRegion": false
        }
    },
    "additionalEventData": {
        "backingKey": "{\"backingKeyId\":\"backing-key-id\"}"
    },
    "requestID": "4f0b185c-588c-4767-9e90-c618f7e13cad",
    "eventID": "c73964b8-703d-49e4-bd9e-f773d0ee1e65",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

以下示例显示了一个在外部密钥存储中创建对称加密 KMS 密钥的 CreateKey 操作的 CloudTrail 日志。


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2022-09-07T22:37:45Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "tags": [],
        "keyUsage": "ENCRYPT_DECRYPT",
        "description": "",
        "origin": "EXTERNAL_KEY_STORE",
        "multiRegion": false,
        "keySpec": "SYMMETRIC_DEFAULT",
        "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "bypassPolicyLockoutSafetyCheck": false,
        "customKeyStoreId": "cks-1234567890abcdef0",
        "xksKeyId": "bb8562717f809024"
    },
    "responseElements": {
        "keyMetadata": {
            "aWSAccountId": "111122223333",
            "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "creationDate": "Dec 7, 2022, 10:37:45 PM",
            "enabled": true,
            "description": "",
            "keyUsage": "ENCRYPT_DECRYPT",
            "keyState": "Enabled",
            "origin": "EXTERNAL_KEY_STORE",
```

```
            "customKeyStoreId": "cks-1234567890abcdef0",
            "keyManager": "CUSTOMER",
            "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
            "keySpec": "SYMMETRIC_DEFAULT",
            "encryptionAlgorithms": [
                "SYMMETRIC_DEFAULT"
            ],
            "multiRegion": false,
            "xksKeyConfiguration": {
                "id": "bb8562717f809024"
            }
        }
    },
    "requestID": "ba197c82-3ac7-487a-8ff4-7736bbeb1316",
    "eventID": "838ad5f4-5fdd-4044-afd7-4dbd88c6af56",
    "readOnly": false,
    "resources": [
        {
            "accountId": "227179770375",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-east-1:227179770375:key/39c5eb22-
f37c-4956-92ca-89e8f8b57ab2"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


Decrypt

这些示例显示了 Decrypt 操作的 Amazon CloudTrail 日志条目。

`requestParameters`即使请求`encryptionAlgorithm`中未指定加密算法，`Decrypt`操作的 CloudTrail 日志条目也始终包含在中。省略了请求中的密文和响应中的明文。

主题

- 使用标准对称加密密钥解密
- 使用标准对称加密密钥解密失败
- 使用密钥库中的 KMS 密钥进行 Amazon CloudHSM 解密
- 使用外部密钥存储中的 KMS 密钥解密

- 使用外部密钥存储中的 KMS 密钥解密失败
- 通过后量子 TLS 连接使用标准对称加密密钥进行解密

使用标准对称加密密钥解密

以下是使用标准对称加密密钥的Decrypt操作的 CloudTrail 日志条目示例。


```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2025-05-20T20:45:00Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "encryptionContext": {
            "Department": "Engineering",
            "Project": "Alpha"
        }
    },
    "responseElements": null,
    "additionalEventData": {
        "keyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    },
    "requestID": "12345126-30d5-4b28-98b9-9153da559963",
    "eventID": "abcde202-ba1a-467c-b4ba-f729d45ae521",
    "readOnly": true,
    "resources": [
        {
```

```
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES256-GCM-SHA384",
        "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
    }
}
```

使用标准对称加密密钥解密失败

以下示例 CloudTrail 日志条目记录了使用标准对称加密 KMS 密钥的失败Decrypt操作。包含异常（errorCode）和错误消息（errorMessage），可帮助您解决错误。

在这种情况下，Decrypt 请求中指定的对称加密 KMS 密钥不是用于加密数据的对称加密 KMS 密钥。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-11-24T18:57:43Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "errorCode": "IncorrectKeyException"
    "errorMessage": "The key ID in the request does not identify a CMK that can perform
 this operation.",
```

```
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "encryptionContext": {
            "Department": "Engineering",
            "Project": "Alpha"
        }
    },
    "responseElements": null,
    "requestID": "22345126-30d5-4b28-98b9-9153da559963",
    "eventID": "abcde202-ba1a-467c-b4ba-f729d45ae521",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

使用密钥库中的 KMS 密钥进行 Amazon CloudHSM 解密

以下示例 CloudTrail 日志条目记录了在密钥库中使用 KMS Amazon CloudHSM 密钥进行的Decrypt操作。使用自定义密钥存储中的 KMS 密钥进行加密操作的所有日志条目都包含带有 customKeyStoreId 和 backingKeyId 的 additionalEventData 字段。backingKeyId 字段中返回的值是 CloudHSM 密钥 id 属性。请求中未指定 additionalEventData。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2021-10-26T23:41:27Z",
```

```
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "encryptionContext": {
            "Department": "Development",
            "Purpose": "Test"
        }
    },
    "responseElements": null,
    "additionalEventData": {
        "customKeyStoreId": "cks-1234567890abcdef0"
    },
    "requestID": "e1b881f8-2048-41f8-b6cc-382b7857ec61",
    "eventID": "a79603d5-4cde-46fc-819c-a7cf547b9df4",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

使用外部密钥存储中的 KMS 密钥解密

以下示例 CloudTrail 日志条目记录了在外部密钥存储中使用 KMS 密钥进行的`Decrypt`操作。除了 `customKeyStoreId`，`additionalEventData` 字段包括外部密钥 ID（`XksKeyId`）。请求中未指定 `additionalEventData`。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
```

```
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-11-24T00:26:58Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "AWS Internal",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
        "encryptionContext": {
            "Department": "Engineering",
            "Purpose": "Test"
        }
    },
    "responseElements": null,
    "additionalEventData": {
        "customKeyStoreId": "cks-9876543210fedcba9",
        "xksKeyId": "abc01234567890fe"
    },
    "requestID": "f1b881f8-2048-41f8-b6cc-382b7857ec61",
    "eventID": "b79603d5-4cde-46fc-819c-a7cf547b9df4",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

使用外部密钥存储中的 KMS 密钥解密失败

以下示例 CloudTrail 日志条目记录了使用外部密钥存储库中的 KMS 密钥Decrypt执行操作的失败请求。 CloudWatch 除了成功的请求外，还会记录失败的请求。记录失败时， CloudTrail 日志条目包括异常（ErrorCode）和随附的错误消息（ErrorMessage）。

如果失败的请求到达了您的外部密钥存储代理（如本示例所示），则您可以使用 requestId 值将失败的请求与您的外部密钥存储代理日志的相应请求关联起来（如果您的代理提供了这些请求）。

如需帮助解决外部密钥存储中的 Decrypt 请求，请参阅 解密错误。


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-11-24T00:26:58Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "errorCode": "KMSInvalidStateException",
    "errorMessage": "The external key store proxy rejected the request because the
 specified ciphertext or additional authenticated data is corrupted, missing, or
 otherwise invalid.",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
        "encryptionContext": {
            "Department": "Engineering",
            "Purpose": "Test"
        }
    },
    "responseElements": null,
    "additionalEventData": {
        "customKeyStoreId": "cks-9876543210fedcba9",
```

```
        "xksKeyId": "abc01234567890fe"
    },
    "requestID": "f1b881f8-2048-41f8-b6cc-382b7857ec61",
    "eventID": "b79603d5-4cde-46fc-819c-a7cf547b9df4",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

通过后量子 TLS 连接使用标准对称加密密钥进行解密

以下是通过后量子 TLS 连接使用标准对称加密密钥的`Decrypt`操作的 CloudTrail 日志条目示例。该`tlsDetails`部分中的 KeyExchange 字段提到`X25519MLKEM768`。这是一种混合算法，将椭圆曲线 Diffie-Hellman (ECDH) 与曲线 25519（当今在 TL S 中使用的经典密钥交换算法）与基于模块格子的密钥封装机制 (ML-KEM) 相结合，后者是一种公钥加密和密钥建立算法，被美国国家标准与技术研究所 (NIST) 指定为其第一个标准的后量子密钥协议算法。

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2025-11-12T15:16:26Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "aws-sdk-java/2.30.22 md/io#async md/http#AwsCommonRuntime ...",
```

```
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "encryptionContext": {
            "Department": "Engineering",
            "Project": "Alpha"
        }
    },
    "responseElements": null,
    "additionalEventData": {
        "keyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    },
    "requestID": "12345126-30d5-4b28-98b9-9153da559963",
    "eventID": "abcde202-ba1a-467c-b4ba-f729d45ae521",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_256_GCM_SHA384",
        "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com",
        "keyExchange": "X25519MLKEM768"
    }
}
```


DeleteAlias

以下示例显示了 DeleteAlias 操作的一个 Amazon CloudTrail 日志条目。有关删除别名的信息，请参阅删除别名。

在 2022 年 12 月或之后记录的此操作的 CloudTrail 日志条目将受影响的 KMS 密钥的密钥 ARN 包含在 `responseElements.keyId` 值中，即使此操作未返回密钥 ARN。

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2014-11-04T00:52:27Z"
            }
        }
    },
    "eventTime": "2014-11-04T00:52:27Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteAlias",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "aliasName": "alias/my_alias"
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "d9542792-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "12f48554-bb04-4991-9cfc-e7e85f68eda0",
    "readOnly": false,
    "resources": [{
        "ARN": "arn:aws:kms:us-east-1:111122223333:alias/my_alias",
        "accountId": "111122223333"
    },
    {
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
```

```
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

DeleteCustomKeyStore

以下示例显示了通过调用 DeleteCustomKeyStore 操作生成的 Amazon CloudTrail 日志条目。有关创建自定义密钥存储的信息，请参阅 删除 Amazon CloudHSM 密钥存储。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2021-10-21T20:17:32Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteCustomKeyStore",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "customKeyStoreId": "cks-1234567890abcdef0"
    },
    "responseElements": null,
    "additionalEventData": {
        "customKeyStoreName": "ExampleKeyStore",
        "clusterId": "cluster-1a23b4cdefg"
    },
    "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
    "eventID": "114b61b9-0ea6-47f5-a9d2-4f2bdd0017d5",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
}
```

DeleteExpiredKeyMaterial

当您将密钥材料导入到 Amazon KMS key（KMS 密钥）时，您可以为该密钥材料设置到期日期和时间。Amazon KMS 在您导入密钥材料（使用过期设置）和 Amazon KMS 删除过期的密钥材料时在 CloudTrail 日志中记录条目。有关创建带已导入密钥材料的 KMS 密钥的信息，请参阅 导入密钥的 Amazon KMS 密钥材料。

以下示例显示了在 Amazon KMS 删除过期的密钥材料时生成的 Amazon CloudTrail 日志条目。


```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "Amazon Internal"
    },
    "eventTime": "2025-05-22T19:55:11Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteExpiredKeyMaterial",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "Amazon Internal",
    "userAgent": "Amazon Internal",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "cfa932fd-0d3a-4a76-a8b8-616863a2b547",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "keyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    },
    "eventCategory": "Management"
}
```

DeleteImportedKeyMaterial

如果您将密钥材料导入 KMS 密钥中，您随时可以使用 DeleteImportedKeyMaterial 操作删除导入的密钥材料。当您从 KMS 密钥中删除导入的密钥材料时，KMS 密钥的密钥状态会更改为 PendingImport，在任何加密操作中都无法使用 KMS 密钥。有关更多信息，请参阅 删除导入的密钥材料。

以下示例显示了为 DeleteImportedKeyMaterial 操作生成的 Amazon CloudTrail 日志条目。


```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2025-05-20T20:45:08Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteImportedKeyMaterial",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "keyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    },
    "responseElements": {
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "keyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    },
    "requestID": "dcf0e82f-dad0-4622-a378-a5b964ad42c1",
    "eventID": "2afbb991-c668-4641-8a00-67d62e1fecbd",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
```

```
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES256-GCM-SHA384",
        "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
    }
}
```

DeleteKey

这些示例显示了在删除 KMS 密钥时生成的 Amazon CloudTrail 日志条目。要删除 KMS 密钥，您可以使用 ScheduleKeyDeletion 操作。在指定的等待期过期后，Amazon KMS 将删除 KMS 密钥，并在您的 CloudTrail 日志中记录类似以下条目来记录该事件。

在 2022 年 12 月或之后记录的此操作的 CloudTrail 日志条目将受影响的 KMS 密钥的密钥 ARN 包含在 `responseElements.keyId` 值中，即使此操作未返回密钥 ARN。

有关 `ScheduleKeyDeletion` 操作的 CloudTrail 日志条目的示例，请参阅 ScheduleKeyDeletion。有关删除 KMS 密钥的信息，请参阅 删除 Amazon KMS key。

以下 CloudTrail 日志条目示例记录了 Amazon KMS 中使用密钥材料的 KMS 密钥 `DeleteKey` 操作。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "Amazon Internal"
    },
    "eventTime": "2020-07-31T00:07:00Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "Amazon Internal",
    "userAgent": "Amazon Internal",
    "requestParameters": null,
    "responseElements": null,
```

```
    "eventID": "b25f9cda-74e1-4458-847b-4972a0bf9668",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "111122223333",
     "managementEvent": true,
    "eventCategory": "Management"
}
```

以下 CloudTrail 日志条目记录了 Amazon CloudHSM 自定义密钥存储中使用 KMS 密钥的 `DeleteKey` 操作。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "Amazon Internal"
    },
    "eventTime": "2021-10-26T23:41:27Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "Amazon Internal",
    "userAgent": "Amazon Internal",
    "requestParameters": null,
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "additionalEventData": {
        "customKeyStoreId": "cks-1234567890abcdef0",
        "clusterId": "cluster-1a23b4cdefg",
        "backingKeys": "[{\"backingKeyId\":\"backing-key-id\"}]",
        "backingKeysDeletionStatus": "[{\"backingKeyId\":\"backing-key-id\",
\"deletionStatus\":\"SUCCESS\"}]"
    },
```

```
    "eventID": "1234585c-4b0c-4340-ab11-662414b79239",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "111122223333",
    "managementEvent": true,
    "eventCategory": "Management"
}
```

DescribeCustomKeyStores

以下示例显示了通过调用 DescribeCustomKeyStores 操作生成的 Amazon CloudTrail 日志条目。有关查看自定义密钥存储的信息，请参阅 查看 Amazon CloudHSM 密钥存储。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2021-10-21T20:17:32Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DescribeCustomKeyStores",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "customKeyStoreId": "cks-1234567890abcdef0"
    },
    "responseElements": null,
    "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
    "eventID": "2ea1735f-628d-43e3-b2ee-486d02913a78",
```

```
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
}
```

DescribeKey

以下示例显示了 DescribeKey 操作的 Amazon CloudTrail 日志条目。您在 Amazon KMS 控制台调用 DescribeKey 操作或查看 KMS 密钥时，Amazon KMS 将记录类似于以下内容的条目。此调用是在 Amazon KMS 管理控制台中查看密钥的结果。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-09-26T18:01:36Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DescribeKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "requestID": "12345126-30d5-4b28-98b9-9153da559963",
    "eventID": "abcde202-ba1a-467c-b4ba-f729d45ae521",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
```

```
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

DisableKey

以下示例显示了 DisableKey 操作的一个 Amazon CloudTrail 日志条目。有关在 Amazon KMS 中启用和禁用的信息 Amazon KMS keys 的信息，请参阅 启用和禁用密钥。

在 2022 年 12 月或之后记录的此操作的 CloudTrail 日志条目将受影响的 KMS 密钥的密钥 ARN 包含在 `responseElements.keyId` 值中，即使此操作未返回密钥 ARN。

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:43Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DisableKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "12345126-30d5-4b28-98b9-9153da559963",
    "eventID": "abcde202-ba1a-467c-b4ba-f729d45ae521",
    "readOnly": false,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
```

```
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

DisableKeyRotation

以下示例显示了通过调用 DisableKeyRotation 操作生成的 Amazon CloudTrail 日志条目。有关启用自动轮换的信息，请参阅 旋转 Amazon KMS keys。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-09-01T19:31:39Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DisableKeyRotation",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "requestID": "d6a9351a-ed6e-4581-88d1-2a9a8a538497",
    "eventID": "6313164c-83aa-4cc3-9e1a-b7c426f7a5b1",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
```

```
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

DisconnectCustomKeyStore

以下示例显示了通过调用 DisconnectCustomKeyStore 操作生成的 Amazon CloudTrail 日志条目。有关断开自定义密钥存储的信息，请参阅 断开 Amazon CloudHSM 密钥存储。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2021-10-21T20:17:32Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DisconnectCustomKeyStore",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "customKeyStoreId": "cks-1234567890abcdef0"
    },
    "responseElements": null,
    "additionalEventData": {
        "customKeyStoreName": "ExampleKeyStore",
        "clusterId": "cluster-1a23b4cdefg"
    },
    "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
    "eventID": "114b61b9-0ea6-47f5-a9d2-4f2bdd0017d5",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
}
```

EnableKey

以下示例显示了 EnableKey 操作的一个 Amazon CloudTrail 日志条目。有关在 Amazon KMS 中启用和禁用的信息 Amazon KMS keys 的信息，请参阅 启用和禁用密钥。

在 2022 年 12 月或之后记录的此操作的 CloudTrail 日志条目将受影响的 KMS 密钥的密钥 ARN 包含在 `responseElements.keyId` 值中，即使此操作未返回密钥 ARN。


```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:20Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "EnableKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "d528a6fb-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "be393928-3629-4370-9634-567f9274d52e",
    "readOnly": false,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

EnableKeyRotation

以下示例显示对 EnableKeyRotation 操作进行的调用的 Amazon CloudTrail 日志条目。有关轮换密钥时写入的 CloudTrail 日志条目的示例，请参阅 RotateKey。有关轮换 Amazon KMS keys 的更多信息，请参阅 旋转 Amazon KMS keys。

> **Note**
>
> rotation-period 是可选请求参数。如果您在启用自动密钥轮换时未指定轮换周期，则默认值为 365 天。

在 2022 年 12 月或之后记录的此操作的 CloudTrail 日志条目将受影响的 KMS 密钥的密钥 ARN 包含在 `responseElements.keyId` 值中，即使此操作未返回密钥 ARN。


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-07-25T23:41:56Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "EnableKeyRotation",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "rotationPeriodInDays": 180
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "81f5b794-452b-4d6a-932b-68c188165273",
    "eventID": "fefc43a7-8e06-419f-bcab-b3bf18d6a401",
    "readOnly": false,
```

```
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

Encrypt

以下示例显示了 Encrypt 操作的一个 Amazon CloudTrail 日志条目。

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2025-05-20T20:46:16Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Encrypt",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "encryptionContext": {
            "Department": "Engineering"
        },
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
    },
    "responseElements": null,
    "additionalEventData": {
```

```
        "keyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    },
    "requestID": "f3423043-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "91235988-eb87-476a-ac2c-0cdc244e6dca",
    "readOnly": true,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_256_GCM_SHA384",
        "clientProvidedHostHeader": "kms.us-east-1.amazonaws.com"
    }
}
```

GenerateDataKey

以下示例显示了 GenerateDataKey 操作的一个 Amazon CloudTrail 日志条目。

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2025-05-20T20:46:16Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
```

```
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "keySpec": "AES_256",
        "encryptionContext": {
            "Department": "Engineering",
            "Project": "Alpha"
        }
    },
    "responseElements": null,
    "additionalEventData": {
        "keyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    },
    "requestID": "e0eb83e3-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "a9dea4f9-8395-46c0-942c-f509c02c2b71",
    "readOnly": true,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_256_GCM_SHA384",
        "clientProvidedHostHeader": "kms.us-east-1.amazonaws.com"
    }
}
```

GenerateDataKeyPair

以下示例显示了 GenerateDataKeyPair 操作的一个 Amazon CloudTrail 日志条目。此示例记录了一个操作，该操作生成一个使用对称加密 Amazon KMS key 加密的 RSA 密钥对。

```
{
    "eventVersion": "1.11",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
```

```
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2025-05-20T20:46:16Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKeyPair",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "keyPairSpec": "RSA_3072",
        "encryptionContext": {
            "Project": "Alpha"
        },
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "additionalEventData": {
        "keyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    },
    "requestID": "52fb127b-0fe5-42bb-8e5e-f560febde6b0",
    "eventID": "9b6bd6d2-529d-4890-a949-593b13800ad7",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_256_GCM_SHA384",
        "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
    }
}
```

GenerateDataKeyPairWithoutPlaintext

以下示例显示了 GenerateDataKeyPairWithoutPlaintext 操作的一个 Amazon CloudTrail 日志条目。此示例记录了一个操作，该操作生成一个使用对称加密 Amazon KMS key 加密的 RSA 密钥对。


```
{
    "eventVersion": "1.11",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2025-05-20T20:46:16Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKeyPairWithoutPlaintext",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "keyPairSpec": "RSA_4096",
        "encryptionContext": {
            "Index": "5"
        },
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "additionalEventData": {
        "keyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    },
    "requestID": "52fb127b-0fe5-42bb-8e5e-f560febde6b0",
    "eventID": "9b6bd6d2-529d-4890-a949-593b13800ad7",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
```

```
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_256_GCM_SHA384",
        "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
    }
}
```

GenerateDataKeyWithoutPlaintext

以下示例显示了 GenerateDataKeyWithoutPlaintext 操作的一个 Amazon CloudTrail 日志条目。

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2025-05-20T20:46:16Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKeyWithoutPlaintext",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "keySpec": "AES_256",
        "encryptionContext": {
            "Project": "Alpha"
        }
    },
    "responseElements": null,
    "additionalEventData": {
        "keyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
```

```
    },
    "requestID": "d6b8e411-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "f7734272-9ec5-4c80-9f36-528ebbe35e4a",
    "readOnly": true,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_256_GCM_SHA384",
        "clientProvidedHostHeader": "kms.us-east-1.amazonaws.com"
    }
}
```

GenerateMac

以下示例显示了 GenerateMac 操作的一个 Amazon CloudTrail 日志条目。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
     },
    "eventTime": "2022-12-23T19:26:54Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateMac",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "macAlgorithm": "HMAC_SHA_512",
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
```

```
    },
    "responseElements": null,
    "requestID": "e0eb83e3-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "a9dea4f9-8395-46c0-942c-f509c02c2b71",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

GenerateRandom

以下示例显示了 GenerateRandom 操作的一个 Amazon CloudTrail 日志条目。因为此操作不使用 Amazon KMS key，`resources` 字段为空。

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:37Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateRandom",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": null,
    "responseElements": null,
    "requestID": "df1e3de6-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "239cb9f7-ae05-4c94-9221-6ea30eef0442",
    "readOnly": true,
```

```
    "resources": [],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GetKeyPolicy

以下示例显示了 GetKeyPolicy 操作的一个 Amazon CloudTrail 日志条目。有关查看 KMS 密钥的密钥策略的信息，请参阅 查看密钥政策。

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:50:30Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GetKeyPolicy",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "policyName": "default"
    },
    "responseElements": null,
    "requestID": "93746dd6-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "4aa7e4d5-d047-452a-a5a6-2cce282a7e82",
    "readOnly": true,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GetKeyRotationStatus

以下示例显示了 GetKeyRotationStatus 操作的一个 Amazon CloudTrail 日志条目。有关 KMS 密钥的密钥材料的自动轮换和按需轮换的信息，请参阅 旋转 Amazon KMS keys。


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2024-02-20T19:16:45Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GetKeyRotationStatus",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "requestID": "12f9b7e8-49b9-4c1c-a7e3-34ac0cdf0467",
    "eventID": "3d082126-9e7d-4167-8372-a6cfcbed4be6",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES256-GCM-SHA384",
```

```
        "clientProvidedHostHeader": "kms.us-east-1.amazonaws.com"
    }
}
```

GetParametersForImport

以下示例显示了您使用 GetParametersForImport 操作时生成的一个 Amazon CloudTrail 日志条目。此操作返回您在将密钥材料导入 KMS 密钥时使用的公有密钥和导入令牌。当您使用 GetParametersForImport 操作或使用 Amazon KMS 控制台下载公有密钥和导入令牌时，会记录相同的 CloudTrail 条目。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2020-07-25T23:58:23Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GetParametersForImport",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "wrappingAlgorithm": "RSAES_OAEP_SHA_256",
        "wrappingKeySpec": "RSA_2048"
    },
    "responseElements": null,
    "requestID": "b5786406-e3c7-43d6-8d3c-6d5ef96e2278",
    "eventID": "4023e622-0c3e-4324-bdef-7f58193bba87",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
```

```
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

ImportKeyMaterial

以下示例显示了您使用 ImportKeyMaterial 操作时生成的一个 Amazon CloudTrail 日志条目。当您使用 `ImportKeyMaterial` 操作或使用 Amazon KMS 控制台导入密钥材料到 Amazon KMS key 中时，会记录相同的 CloudTrail 条目。

在 2022 年 12 月或之后记录的此操作的 CloudTrail 日志条目将受影响的 KMS 密钥的密钥 ARN 包含在 `responseElements.keyId` 值中，即使此操作未返回密钥 ARN。

```
{
    "eventVersion": "1.11",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2025-05-21T05:42:31Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ImportKeyMaterial",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "validTo": "May 21, 2025, 5:47:45 AM",
        "expirationModel": "KEY_MATERIAL_EXPIRES",
        "importType": "NEW_KEY_MATERIAL",
        "keyMaterialDescription": "ExampleKeyMaterialA"
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "keyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    },
```

```
    "requestID": "89e10ee7-a612-414d-95a2-a128346969fd",
    "eventID": "c7abd205-a5a2-4430-bbfa-fc10f3e2d79f",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_256_GCM_SHA384",
        "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
    }
}
```

ListAliases

以下示例显示了 ListAliases 操作的一个 Amazon CloudTrail 日志条目。因为此操作不使用任何特定别名或 Amazon KMS key，`resources` 字段为空。有关在 Amazon KMS 中查看别名的信息，请参阅 查找 KMS 密钥的别名和别名 ARN。

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:51:45Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ListAliases",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
```

```
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "limit": 5,
        "marker":
 "eyJiIjoiYWxpYXMvZTU0Y2MxOTMtYTMwNC00YzEwLTliZWItYTJjZjA3NjA2OTJhIiwiYSI6ImFsaWFzL2U1NGNjMTkzL
    },
    "responseElements": null,
    "requestID": "bfe6c190-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "a27dda7b-76f1-4ac3-8b40-42dfba77bcd6",
    "readOnly": true,
    "resources": [],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```


ListGrants

以下示例显示了 ListGrant 操作的一个 Amazon CloudTrail 日志条目。有关 Amazon KMS 的授权的信息，请参阅 Amazon KMS 中的授权。


```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:49Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ListGrants",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "marker":
 "eyJncmFudElkIjoiMWY4M2U2ZmM0YTY2NDgxYjQ2Yzc4MTdhM2Y4YmQwMDFkZDNiYmQ1MGVlYTMyY2RmOWFiNWY1Nzc1N
\u003d\u003d",
        "limit": 10
    },
```

```
    "responseElements": null,
    "requestID": "e5c23960-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "d24380f5-1b20-4253-8e92-dd0492b3bd3d",
    "readOnly": true,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

ListKeyRotations

以下示例显示了 ListKeyRotations 操作的一个 Amazon CloudTrail 日志条目。有关 KMS 密钥的密钥材料的自动轮换和按需轮换的信息，请参阅 旋转 Amazon KMS keys。

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2025-05-21T05:42:35Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ListKeyRotations",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "includeKeyMaterial": "ALL_KEY_MATERIAL"
    },
    "responseElements": null,
    "requestID": "99c88d32-f2db-455e-8a9a-23855258a452",
    "eventID": "8ce0e74b-b9c7-45a2-96ef-83136d38068e",
    "readOnly": true,
    "resources": [
```

```
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES256-GCM-SHA384",
        "clientProvidedHostHeader": "kms.us-east-1.amazonaws.com"
    }
}
```

PutKeyPolicy

以下示例显示了通过调用 PutKeyPolicy 操作生成的 Amazon CloudTrail 日志条目。有关更新密钥策略的信息，请参阅 更改密钥策略。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-09-01T20:06:16Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "PutKeyPolicy",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "policyName": "default",
```

```
        "policy": "{\n  \"Version\" : \"2012-10-17\",\n  \"Id\" : \"key-default-1\",\n
 \"Statement\" : [ {\n    \"Sid\" : \"Enable IAM User Permissions\",\n    \"Effect\" :
 \"Allow\",\n    \"Principal\" : {\n      \"AWS\" : \"arn:aws:iam::111122223333:root
\"\n    },\n    \"Action\" : \"kms:*\",\n    \"Resource\" : \"*\"\n  } ]\n}",
        "bypassPolicyLockoutSafetyCheck": false
    },
    "responseElements": null,
    "requestID": "7bb906fa-dc21-4350-b65c-808ff0f72f55",
    "eventID": "c217db1f-903f-4a2f-8f88-9580182d6313",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

ReEncrypt

以下示例显示了 ReEncrypt 操作的一个 Amazon CloudTrail 日志条目。此日志条目中的 `resources` 字段按此顺序指定两个 Amazon KMS keys，源 KMS 密钥和目标 KMS 密钥。

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2025-05-22T19:34:55Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ReEncrypt",
    "awsRegion": "us-west-2",
```

```
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "sourceEncryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "sourceEncryptionContext": {
            "Project": "Alpha",
            "Department": "Engineering"
        },
        "destinationKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
        "destinationEncryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "destinationEncryptionContext": {
            "Level": "3A"
        }
    },
    "responseElements": null,
    "additionalEventData": {
        "destinationKeyMaterialId":
 "96083e4fb6dbc41d77578a213a6b6669c044dd4c143e96755396d2bf11fd6068",
        "sourceKeyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
    },
    "requestID": "03769fd4-acf9-4b33-adf3-2ab8ca73aadf",
    "eventID": "542d9e04-0e8d-4e05-bf4b-4bdeb032e6ec",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        },
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
```

```
        "cipherSuite": "TLS_AES_256_GCM_SHA384",
        "clientProvidedHostHeader": "kms.us-west-2.amazonaws.com"
    }
}
```

ReplicateKey

以下示例显示了通过调用 ReplicateKey 操作生成的 Amazon CloudTrail 日志条目。ReplicateKey 操作和 CreateKey 操作中的 ReplicateKey 请求结果。

有关删除多区域密钥的信息，请参阅 创建多区域副本密钥。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-11-18T01:29:18Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ReplicateKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "replicaRegion": "us-west-2",
        "bypassPolicyLockoutSafetyCheck": false,
        "description": ""
    },
    "responseElements": {
        "replicaKeyMetadata": {
            "aWSAccountId": "111122223333",
            "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "creationDate": "Nov 18, 2020, 1:29:18 AM",
            "enabled": false,
            "description": "",
```

```
            "keyUsage": "ENCRYPT_DECRYPT",
            "keyState": "Creating",
            "origin": "AWS_KMS",
            "keyManager": "CUSTOMER",
            "keySpec": "SYMMETRIC_DEFAULT",
            "customerMasterKeySpec": "SYMMETRIC_DEFAULT",
            "encryptionAlgorithms": [
                "SYMMETRIC_DEFAULT"
            ],
            "multiRegion": true,
            "multiRegionConfiguration": {
                "multiRegionKeyType": "REPLICA",
                "primaryKey": {
                    "arn": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
                    "region": "us-east-1"
                },
                "replicaKeys": [
                    {
                        "arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
                        "region": "us-west-2"
                    }
                ]
            }
        },
        "replicaPolicy": "{\n  \"Version\":\"2012-10-17\",\n  \"Statement\":[{\n
 \"Effect\":\"Allow\",\n    \"Principal\":{\"AWS\":\"arn:aws:iam::123456789012:user/
Alice\"},\n    \"Action\":\"kms:*\",\n    \"Resource\":\"*\"\n  }, {\n    \"Effect
\":\"Allow\",\n    \"Principal\":{\"AWS\":\"arn:aws:iam::012345678901:user/Bob\"},\n
    \"Action\":\"kms:CreateGrant\",\n    \"Resource\":\"*\"\n  }, {\n    \"Effect\":
\"Allow\",\n    \"Principal\":{\"AWS\":\"arn:aws:iam::012345678901:user/Charlie\"},\n
  \"Action\":\"kms:Encrypt\",\n    \"Resource\":\"*\"\n}]\n}",
    },
    "requestID": "abcdef68-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "fedcba44-6773-4f96-8763-1993aec9ae6a",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
east-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
```

```
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

RetireGrant

以下示例显示了通过调用 RetireGrant 操作生成的 Amazon CloudTrail 日志条目。有关停用授权的信息，请参阅 停用和撤销授权。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-09-01T19:39:33Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "RetireGrant",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData": {
        "grantId": "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a"
    },
    "requestID": "1d274d57-5697-462c-a004-f25fcc29fa26",
    "eventID": "0771bcfb-3e24-4332-9ac8-e1c06563eecf",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
```

```
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

RevokeGrant

以下示例显示了通过调用 RevokeGrant 操作生成的 Amazon CloudTrail 日志条目。有关撤销授权的信息，请参阅 停用和撤销授权。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-09-01T19:35:17Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "RevokeGrant",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "grantId": "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a"
    },
    "responseElements": null,
    "requestID": "59d94c03-c5b7-428d-ae6e-f2c4b47d2917",
    "eventID": "07a23a39-6526-4ae2-b31e-d35fbe9e24ee",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
```

```
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

RotateKey

这些示例显示用于轮换 Amazon KMS keys 的操作的 Amazon CloudTrail 日志条目。有关轮换 KMS 密钥的信息，请参阅 旋转 Amazon KMS keys。

以下示例显示了一个轮换对称加密 KMS 密钥且启用了自动密钥轮换的操作的 CloudTrail 日志条目。有关启用自动轮换的信息，请参阅旋转 Amazon KMS keys。

有关记录 `EnableKeyRotation` 操作的 CloudTrail 日志条目的示例，请参阅 EnableKeyRotation。

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "Amazon Internal"
    },
    "eventTime": "2025-05-20T20:44:37Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "RotateKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "Amazon Internal",
    "userAgent": "Amazon Internal",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "a24b3967-ddad-417f-9b22-2332b918db06",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
```

```
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
        "rotationType": "AUTOMATIC",
        "keyOrigin": "AWS_KMS",
        "previousKeyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0",
        "currentKeyMaterialId":
 "96083e4fb6dbc41d77578a213a6b6669c044dd4c143e96755396d2bf11fd6068",
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "eventCategory": "Management"
}
```

以下示例展示了一条由 RotateKeyOnDemand 操作启动的按需轮换 CloudTrail 日志条目。有关按需轮换对称加密 KMS 密钥的信息，请参阅执行按需密钥轮换。

有关记录 RotateKeyOnDemand 操作的 CloudTrail 日志条目的示例，请参阅 RotateKeyOnDemand。

```
{
    "eventVersion": "1.11",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "Amazon Internal"
    },
    "eventTime": "2025-05-20T20:44:37Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "RotateKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "Amazon Internal",
    "userAgent": "Amazon Internal",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "a24b3967-ddad-417f-9b22-2332b918db06",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsServiceEvent",
```

```
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "serviceEventDetails": {
        "rotationType": "ON_DEMAND",
        "keyOrigin": "EXTERNAL",
        "previousKeyMaterialId":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0",
        "currentKeyMaterialId":
 "96083e4fb6dbc41d77578a213a6b6669c044dd4c143e96755396d2bf11fd6068",
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "eventCategory": "Management"
}
```


RotateKeyOnDemand

以下示例显示了 RotateKeyOnDemand 操作的一个 Amazon CloudTrail 日志条目。有关轮换密钥时写入的 CloudTrail 日志条目的示例，请参阅 RotateKey。有关按需轮换 KMS 密钥的密钥材料的更多信息，请参阅执行按需密钥轮换。


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2024-02-20T17:41:57Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "RotateKeyOnDemand",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "9e1dee86-eb84-42fd-8f25-e3fc7dbb32c8",
```

```
    "eventID": "00a09fbc-20d6-4a58-9b92-7da85984ab77",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES256-GCM-SHA384",
        "clientProvidedHostHeader": "kms.us-east-1.amazonaws.com"
    }
}
```

ScheduleKeyDeletion

这些示例显示用于 ScheduleKeyDeletion 操作的 Amazon CloudTrail 日志条目。

有关删除密钥时写入的 CloudTrail 日志条目的示例，请参阅 DeleteKey。关于删除 Amazon KMS keys 的信息，请查阅 删除 Amazon KMS key。

以下示例记录对单区域 KMS 密钥的 `ScheduleKeyDeletion` 请求。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2021-03-23T18:58:30Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ScheduleKeyDeletion",
```

```
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "pendingWindowInDays": 20,
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "keyState": "PendingDeletion",
        "deletionDate": "Apr 12, 2021 18:58:30 PM"
    },
    "requestID": "ee408f36-ea01-422b-ac14-b0f147c68334",
    "eventID": "3c4226b0-1e81-48a8-a333-7fa5f3cbd118",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

以下示例记录对拥有副本密钥的多区域 KMS 密钥的 ScheduleKeyDeletion 请求。

由于 Amazon KMS 在删除所有副本密钥之前不会删除多区域密钥，在 responseElements 字段中，keyState 是 PendingReplicaDeletion，deletionDate 字段会被省略。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
```

```
    "eventTime": "2021-10-28T17:59:05Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ScheduleKeyDeletion",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "pendingWindowInDays": 30,
        "keyId": "mrk-1234abcd12ab34cd56ef1234567890ab"
    },
    "responseElements": {
        "keyId": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "keyState": "PendingReplicaDeletion",
        "pendingWindowInDays": 30
    },
    "requestID": "12341411-d846-42a6-a476-b1cbe3011f89",
    "eventID": "abcda5f-396d-494c-9380-0c47860df5f1",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

以下示例在 Amazon CloudHSM 自定义密钥存储中记录对 KMS 密钥的 ScheduleKeyDeletion 请求。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
```

```
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2021-10-26T23:25:25Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "ScheduleKeyDeletion",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
        "pendingWindowInDays": 30
    },
    "responseElements": {
        "keyId": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321",
        "deletionDate": "Nov 2, 2021, 11:25:25 PM",
        "keyState": "PendingDeletion",
        "pendingWindowInDays": 30
    },
    "additionalEventData": {
        "customKeyStoreId": "cks-1234567890abcdef0",
        "clusterId": "cluster-1a23b4cdefg",
        "backingKeys": "[{\"backingKeyId\":\"backing-key-id\"}]"
    },
    "requestID": "abcd9f60-2c9c-4a0b-a456-d5d998f7f321",
    "eventID": "ca01996a-01b0-4edd-bbbb-25d7b6d1a6fa",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

Sign

这些示例显示用于 Sign 操作的 Amazon CloudTrail 日志条目。

以下示例显示用于 Sign 操作的 CloudTrail 日志条目，其使用非对称 RSA KMS 密钥生成文件的数字签名。


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-03-07T22:36:44Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Sign",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "messageType": "RAW",
        "keyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
        "signingAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
    },
    "responseElements": null,
    "requestID": "8d0b35e0-46cf-48b9-be99-bf2ebc9ab9fb",
    "eventID": "107b3cac-b125-4556-9702-12a2b9afcff7",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
```

```
    "eventCategory": "Management"
}
```

SynchronizeMultiRegionKey

以下示例显示了 Amazon KMS 同步多区域密钥时生成的一个 Amazon CloudTrail 日志条目。同步涉及将多区域主键的共享属性复制到其副本密钥的跨区域调用。Amazon KMS 会定期同步多区域密钥，以确保所有相关的多区域密钥具有相同的密钥材料。

CloudTrail 日志条目的 `resources` 元素包含多区域主键的密钥 ARN，包括其 Amazon Web Services 区域。此日志条目中未列出相关的多区域副本密钥及其区域。

在 2022 年 12 月或之后记录的此操作的 CloudTrail 日志条目将受影响的 KMS 密钥的密钥 ARN 包含在 `responseElements.keyId` 值中，即使此操作未返回密钥 ARN。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "AWS Internal"
    },
    "eventTime": "2020-11-18T02:04:37Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "SynchronizeMultiRegionKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "AWS Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "12345681-de97-42e9-bed0-b02ae1abd8dc",
    "eventID": "abcdec99-2b5c-4670-9521-ddb8f031e146",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
```

```
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

TagResource

以下示例显示了对 TagResource 操作进行的添加标签键为 Department 且标签值为 IT 的标签的调用的 Amazon CloudTrail 日志条目。

有关轮换密钥时写入的 UntagResource CloudTrail 日志条目的示例，请参阅 UntagResource。有关标记 Amazon KMS keys 的更多信息，请参阅 标签在 Amazon KMS。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-07-01T21:19:25Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "TagResource",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "tags": [
            {
                "tagKey": "Department",
                "tagValue": "IT"
            }
        ]
    },
    "responseElements": null,
    "requestID": "b942584a-f77d-4787-9feb-b9c5be6e746d",
```

```
    "eventID": "0a091b9b-0df5-4cf9-b667-6f2879532b8f",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

UntagResource

以下示例显示了对 UntagResource 操作进行的删除标签键为 Dept 的标签的调用的 Amazon CloudTrail 日志条目。

在 2022 年 12 月或之后记录的此操作的 CloudTrail 日志条目将受影响的 KMS 密钥的密钥 ARN 包含在 responseElements.keyId 值中，即使此操作未返回密钥 ARN。

有关 TagResource CloudTrail 日志条目的示例，请参阅 TagResource。有关标记 Amazon KMS keys 的更多信息，请参阅 标签在 Amazon KMS。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-07-01T21:19:19Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "UntagResource",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
```

```
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "tagKeys": [
            "Dept"
        ]
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "cb1d507b-6015-47f4-812b-179713af8068",
    "eventID": "0b00f4b0-036e-411d-aa75-87eb4a35a4b3",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

UpdateAlias

以下示例显示了 UpdateAlias 操作的一个 Amazon CloudTrail 日志条目。`resources` 元素包含别名和 KMS 密钥资源的字段。有关在 Amazon KMS 中创建别名的信息，请参阅 创建别名。

在 2022 年 12 月或之后记录的此操作的 CloudTrail 日志条目将受影响的 KMS 密钥的密钥 ARN 包含在 `responseElements.keyId` 值中，即使此操作未返回密钥 ARN。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
         "principalId": "EX_PRINCIPAL_ID",
         "arn": "arn:aws:iam::111122223333:user/Alice",
         "accountId": "111122223333",
         "accessKeyId": "EXAMPLE_KEY_ID",
         "userName": "Alice"
    },
```

```
    "eventTime": "2020-11-13T23:18:15Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "UpdateAlias",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "aliasName": "alias/my_alias",
        "targetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "d9472f40-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "f72d3993-864f-48d6-8f16-e26e1ae8dff0",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:alias/my_alias"
        },
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

UpdateCustomKeyStore

以下示例显示了通过调用 UpdateCustomKeyStore 操作来更新自定义密钥存储的集群 ID 而生成的 Amazon CloudTrail 日志条目。有关编辑自定义密钥存储的信息，请参阅 编辑 Amazon CloudHSM 密钥存储设置。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
```

```
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2021-10-21T20:17:32Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "UpdateCustomKeyStore",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "customKeyStoreId": "cks-1234567890abcdef0",
        "clusterId": "cluster-1a23b4cdefg"
    },
    "responseElements": null,
    "additionalEventData": {
        "customKeyStoreName": "ExampleKeyStore",
        "clusterId": "cluster-1a23b4cdefg"
    },
    "requestID": "abcde9e1-f1a3-4460-a423-577fb6e695c9",
    "eventID": "114b61b9-0ea6-47f5-a9d2-4f2bdd0017d5",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333"
}
```

UpdateKeyDescription

以下示例显示了通过调用 UpdateKeyDescription 操作生成的 Amazon CloudTrail 日志条目。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
```

```
    },
    "eventTime": "2022-09-01T19:22:40Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "UpdateKeyDescription",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "description": "New key description"
    },
    "responseElements": null,
    "requestID": "8c3c1f8b-336d-4896-b034-4eb9916bc9b3",
    "eventID": "f5f3d548-2e9e-4658-8427-9dcb5b1ea791",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```

UpdatePrimaryRegion

以下示例显示了通过对多区域密钥调用 UpdatePrimaryRegion 操作生成的 Amazon CloudTrail 日志条目。

UpdatePrimaryRegion 操作将写入两个 CloudTrail 日志条目：一个位于具有多区域主键的区域中，被转换为副本密钥，一个位于具有多区域副本密钥的区域中，被转换为主键。

在 2022 年 12 月或之后记录的此操作的 CloudTrail 日志条目将受影响的 KMS 密钥的密钥 ARN 包含在 `responseElements.keyId` 值中，即使此操作未返回密钥 ARN。

下面的示例显示了多区域密钥从主键更改为副本密钥所在区域 (us-west-2) 中的 UpdatePrimaryRegion 的 CloudTrail 日志条目。`primaryRegion` 字段显示了现在托管主键的区域 (ap-northeast-1)。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2021-03-10T20:23:37Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "UpdatePrimaryRegion",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
        "primaryRegion": "ap-northeast-1"
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "ee408f36-ea01-422b-ac14-b0f147c68334",
    "eventID": "3c4226b0-1e81-48a8-a333-7fa5f3cbd118",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333"
}
```


下面的示例代表了多区域密钥从副本密钥更改为主键所在区域 (ap-northeast-1) 中的 UpdatePrimaryRegion 的 CloudTrail 日志条目。此日志条目没有标识以前的主要区域。

```
{
    "eventVersion": "1.08",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice",
            "invokedBy": "kms.amazonaws.com"
    },
    "eventTime": "2021-03-10T20:23:37Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "UpdatePrimaryRegion",
    "awsRegion": "ap-northeast-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "AWS Internal",
    "requestParameters": {
        "keyId": "arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "primaryRegion": "ap-northeast-1"
    },
    "responseElements": {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "requestID": "ee408f36-ea01-422b-ac14-b0f147c68334",
    "eventID": "091e6be5-737f-43c6-8431-e3679d6d0619",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "eventCategory": "Management",
    "recipientAccountId": "111122223333"
}
```


VerifyMac

以下示例显示了 VerifyMac 操作的一个 Amazon CloudTrail 日志条目。


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
```

```
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
     },
    "eventTime": "2022-03-31T19:25:54Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "VerifyMac",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "macAlgorithm": "HMAC_SHA_384",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "requestID": "f35da560-edff-4d6e-9b40-fb306fa9ef1e",
    "eventID": "6b464487-6dea-44cd-84ad-225d7450c975",
    "readOnly": true,
    "resources": [
        {
           "accountId": "111122223333",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


验证

这些示例显示用于 Verify 操作的 Amazon CloudTrail 日志条目。

以下示例显示用于 Verify 操作的 CloudTrail 日志条目，其使用非对称 RSA KMS 密钥验证数字签名。


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
```

```
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2022-03-07T22:50:41Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Verify",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "signingAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256",
        "keyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
        "messageType": "RAW"
    },
    "responseElements": null,
    "requestID": "c73ab82a-af82-4750-ae2c-b6bb790e9c28",
    "eventID": "3b4331cd-5b7b-4de5-bf5f-82ec22f0dac0",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-west-2:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321"
        }
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "111122223333",
    "eventCategory": "Management"
}
```


Amazon EC2 示例一

以下示例记录了 IAM 主体如何在 Amazon EC2 管理控制台中使用默认卷密钥创建加密卷。

以下示例显示了一个 CloudTrail 日志条目，该日志条目显示了用户 Alice 如何在 Amazon EC2 管理控制台中使用默认卷密钥创建加密卷。该 EC2 日志文件记录包含一个 `volumeId` 字段，其值为 `"vol-13439757"`。Amazon KMS 记录包含一个 `encryptionContext` 字段，其值为 `"aws:ebs:id": "vol-13439757"`。同样，这两个记录的 `principalId` 和 `accountId` 都相互匹配。这些记录反映了一个事实，即创建加密卷生成的数据密钥被用于加密卷内容。

```
{
  "Records": [
     {
      "eventVersion": "1.02",
      "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
        },
      "eventTime": "2014-11-05T20:50:18Z",
      "eventSource": "ec2.amazonaws.com",
      "eventName": "CreateVolume",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "Amazon Internal",
      "requestParameters": {
        "size": "10",
        "zone": "us-east-1a",
        "volumeType": "gp2",
        "encrypted": true
      },
      "responseElements": {
        "volumeId": "vol-13439757",
        "size": "10",
        "zone": "us-east-1a",
        "status": "creating",
        "createTime": 1415220618876,
        "volumeType": "gp2",
        "iops": 30,
        "encrypted": true
      },
      "requestID": "1565210e-73d0-4912-854c-b15ed349e526",
      "eventID": "a3447186-135f-4b00-8424-bc41f1a93b4f",
      "eventType": "AwsApiCall",
      "recipientAccountId": "123456789012"
    },
    {
      "eventVersion": "1.02",
      "userIdentity": {
        "type": "IAMUser",
```

```
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
      },
      "eventTime": "2014-11-05T20:50:19Z",
      "eventSource": "kms.amazonaws.com",
      "eventName": "GenerateDataKeyWithoutPlaintext",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "&AWS; Internal",
      "requestParameters": {
        "encryptionContext": {
          "aws:ebs:id": "vol-13439757"
        },
        "numberOfBytes": 64,
        "keyId": "alias/aws/ebs"
      },
      "responseElements": null,
      "requestID": "create-123456789012-758241111-1415220618",
      "eventID": "4bd2a696-d833-48cc-b72c-05e61b608399",
      "readOnly": true,
      "resources": [
        {
          "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
          "accountId": "111122223333"
        }
      ],
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    }
  ]
}
```


Amazon EC2 示例二

在以下示例中，运行 Amazon EC2 实例的 IAM 主体会创建并挂载一个使用 KMS 密钥加密的数据卷。此操作会生成多个 CloudTrail 日志记录。有关 Amazon EBS 和加密的更多信息，请参阅 Requirements for Amazon EBS encryption。

创建卷时，Amazon EC2 将代表客户从 Amazon KMS (GenerateDataKeyWithoutPlaintext) 获取加密的数据密钥。然后会创建一个授权 (CreateGrant)，从而允许它解密数据密钥。装载卷时，Amazon EC2 会调用 Amazon KMS 来解密数据密钥 (Decrypt)。

Amazon EC2 实例的 instanceId、"i-81e2f56c" 将出现在 RunInstances 事件中。使用相同的实例 ID 来限定所创建授权的 granteePrincipal ("111122223333:aws:ec2-infrastructure:i-81e2f56c") 以及 Decrypt 调用中的委托人的代入角色 ("arn:aws:sts::111122223333:assumed-role/aws:ec2-infrastructure/i-81e2f56c")。

保护数据卷的 KMS 密钥的密钥 ARN、arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab 将出现在全部三个 Amazon KMS 调用（CreateGrant、GenerateDataKeyWithoutPlaintext 和 Decrypt）中。


```
{
  "Records": [
    {
      "eventVersion": "1.02",
      "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
      },
      "eventTime": "2014-11-05T21:35:27Z",
      "eventSource": "ec2.amazonaws.com",
      "eventName": "RunInstances",
      "awsRegion": "us-west-2",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "Amazon Internal",
      "requestParameters": {
        "instancesSet": {
          "items": [
            {
              "imageId": "ami-b66ed3de",
              "minCount": 1,
              "maxCount": 1
            }
          ]
        },
```

```
        "groupSet": {
          "items": [
            {
              "groupId": "sg-98b6e0f2"
            }
          ]
        },
        "instanceType": "m3.medium",
        "blockDeviceMapping": {
          "items": [
            {
              "deviceName": "/dev/xvda",
              "ebs": {
                "volumeSize": 8,
                "deleteOnTermination": true,
                "volumeType": "gp2"
              }
            },
            {
              "deviceName": "/dev/sdb",
              "ebs": {
                "volumeSize": 8,
                "deleteOnTermination": false,
                "volumeType": "gp2",
                "encrypted": true
              }
            }
          ]
        },
        "monitoring": {
          "enabled": false
        },
        "disableApiTermination": false,
        "instanceInitiatedShutdownBehavior": "stop",
        "clientToken": "XdKUT141516171819",
        "ebsOptimized": false
      },
      "responseElements": {
        "reservationId": "r-5ebc9f74",
        "ownerId": "111122223333",
        "groupSet": {
          "items": [
            {
              "groupId": "sg-98b6e0f2",
```

```
            "groupName": "launch-wizard-2"
          }
        ]
      },
      "instancesSet": {
        "items": [
          {
            "instanceId": "i-81e2f56c",
            "imageId": "ami-b66ed3de",
            "instanceState": {
              "code": 0,
              "name": "pending"
            },
            "amiLaunchIndex": 0,
            "productCodes": {

            },
            "instanceType": "m3.medium",
            "launchTime": 1415223328000,
            "placement": {
              "availabilityZone": "us-east-1a",
              "tenancy": "default"
            },
            "monitoring": {
              "state": "disabled"
            },
            "stateReason": {
              "code": "pending",
              "message": "pending"
            },
            "architecture": "x86_64",
            "rootDeviceType": "ebs",
            "rootDeviceName": "/dev/xvda",
            "blockDeviceMapping": {

            },
            "virtualizationType": "hvm",
            "hypervisor": "xen",
            "clientToken": "XdKUT1415223327917",
            "groupSet": {
              "items": [
                {
                  "groupId": "sg-98b6e0f2",
                  "groupName": "launch-wizard-2"
```

```
                }
              ]
            },
            "networkInterfaceSet": {

            },
            "ebsOptimized": false
          }
        ]
      }
    },
    "requestID": "41c4b4f7-8bce-4773-bf0e-5ae3bb5cbce2",
    "eventID": "cd75a605-2fee-4fda-b847-9c3d330ebaae",
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
  },
  {
    "eventVersion": "1.02",
    "userIdentity": {
          "type": "IAMUser",
          "principalId": "EX_PRINCIPAL_ID",
          "arn": "arn:aws:iam::111122223333:user/Alice",
          "accountId": "111122223333",
          "accessKeyId": "EXAMPLE_KEY_ID",
          "userName": "Alice"
    },
    "eventTime": "2014-11-05T21:35:35Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "CreateGrant",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
      "constraints": {
        "encryptionContextSubset": {
          "aws:ebs:id": "vol-f67bafb2"
        }
      },
      "granteePrincipal": "111122223333:aws:ec2-infrastructure:i-81e2f56c",
      "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": {
      "grantId": "abcde1237f76e4ba7987489ac329fbfba6ad343d6f7075dbd1ef191f0120514a"
```

```
      },
      "requestID": "41c4b4f7-8bce-4773-bf0e-5ae3bb5cbce2",
      "eventID": "c1ad79e3-0d3f-402a-b119-d5c31d7c6a6c",
      "readOnly": false,
      "resources": [
        {
          "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
          "accountId": "111122223333"
        }
      ],
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    },
    {
      "eventVersion": "1.02",
      "userIdentity": {
          "type": "IAMUser",
          "principalId": "EX_PRINCIPAL_ID",
          "arn": "arn:aws:iam::111122223333:user/Alice",
          "accountId": "111122223333",
          "accessKeyId": "EXAMPLE_KEY_ID",
          "userName": "Alice"
      },
      "eventTime": "2014-11-05T21:35:32Z",
      "eventSource": "kms.amazonaws.com",
      "eventName": "GenerateDataKeyWithoutPlaintext",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "userAgent": "Amazon Internal",
      "requestParameters": {
        "encryptionContext": {
          "aws:ebs:id": "vol-f67bafb2"
        },
        "numberOfBytes": 64,
        "keyId": "alias/aws/ebs"
      },
      "responseElements": null,
      "requestID": "create-111122223333-758247346-1415223332",
      "eventID": "ac3cab10-ce93-4953-9d62-0b6e5cba651d",
      "readOnly": true,
      "resources": [
        {
```

```
          "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
          "accountId": "111122223333"
        }
      ],
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    },
    {
      "eventVersion": "1.02",
      "userIdentity": {
        "type": "AssumedRole",
        "principalId": "111122223333:aws:ec2-infrastructure:i-81e2f56c",
        "arn": "arn:aws:sts::111122223333:assumed-role/aws:ec2-infrastructure/
i-81e2f56c",
        "accountId": "111122223333",
        "accessKeyId": "",
        "sessionContext": {
          "attributes": {
            "mfaAuthenticated": "false",
            "creationDate": "2014-11-05T21:35:38Z"
          },
          "sessionIssuer": {
            "type": "Role",
            "principalId": "111122223333:aws:ec2-infrastructure",
            "arn": "arn:aws:iam::111122223333:role/aws:ec2-infrastructure",
            "accountId": "111122223333",
            "userName": "aws:ec2-infrastructure"
          }
        }
      },
      "eventTime": "2014-11-05T21:35:47Z",
      "eventSource": "kms.amazonaws.com",
      "eventName": "Decrypt",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "192.0.2.0",
      "requestParameters": {
        "encryptionContext": {
          "aws:ebs:id": "vol-f67bafb2"
        }
      },
      "responseElements": null,
      "requestID": "b4b27883-6533-11e4-b4d9-751f1761e9e5",
      "eventID": "edb65380-0a3e-4123-bbc8-3d1b7cff49b0",
```

```
      "readOnly": true,
      "resources": [
        {
          "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
          "accountId": "111122223333"
        }
      ],
      "eventType": "AwsApiCall",
      "recipientAccountId": "111122223333"
    }
  ]
}
```

使用 Amazon CloudWatch 监控 KMS 密钥

您可以使用 Amazon CloudWatch 监控 Amazon KMS keys，这是一项 Amazon 服务，可从 Amazon KMS 收集原始数据，并将数据处理为便于读取的近乎实时的指标。这些数据会保存两周，从而使您能够访问历史信息，并更好地了解您 KMS 密钥的使用情况及其随时间推移而发生的变化。

您可以使用 Amazon CloudWatch 提示自己注意重要事件，例如：

- KMS 密钥中导入的密钥材料临近到期日期。
- 仍在使用待删除的 KMS 密钥。
- KMS 密钥中的密钥材料已自动轮换。
- KMS 密钥已删除。

您也可以创建 Amazon CloudWatch 告警，当您的请求速率达到配额值的特定百分比时向您发出提示。有关详细信息，请参阅 Amazon 安全博客中的使用 Service Quotas 和 Amazon CloudWatch 管理您的 Amazon KMS API 请求速度。

Amazon KMS 指标和维度

Amazon KMS 预定义 Amazon CloudWatch 指标，使您可以更轻松地监控关键数据并创建警报。您可以使用 Amazon Web Services 管理控制台 和 Amazon CloudWatch API 查看 Amazon KMS 指标。

本部分列出了各个 Amazon KMS 指标及各指标的维度，并提供有关根据这些指标和维度创建 CloudWatch 警报的基本指导。

> **Note**
>
> 维度组名称：
> 要在 Amazon CloudWatch 控制台中查看指标，请在 Metrics（指标）部分选择维度组名称。然后，您可以按 Metric name（指标名称）进行筛选。本主题包括各个 Amazon KMS 指标的指标名称和维度组名称。

您可以使用 Amazon Web Services 管理控制台 和 Amazon CloudWatch API 查看 Amazon KMS 指标。有关更多信息，请参阅《Amazon CloudWatch 用户指南》中的查看可用指标。

主题

SuccessfulRequest

特定 KMS 密钥上的成功密码操作请求数量。通过使用 `SuccessfulRequest` 指标，您可以在 CloudWatch 中对 Amazon KMS API 使用情况执行密钥级别的筛选。此指标的 `Sum` 统计数据用于定义该时间段内成功请求的总数。

此指标用来确定占用请求配额最大或产生 API 费用最大的 KMS 密钥。您还可以创建基于 `SuccesfulRequest` 的 CloudWatch 警报，从而在出现异常的 API Amazon KMS 使用模式时通知您。这些警报有助于识别可能在无意中超出请求配额或产生意外费用的效率低下的工作流。

的维度`SuccessfulRequest`

维度	描述
KeyArn	每个 KMS 密钥的值。
操作	每个 Amazon KMS API 操作的值。此指标仅适用于密码操作。

对于 ReEncrypt 操作，SuccessfulRequest 指标将包括源 KMS 密钥和目标 KMS 密钥的维度。

维度	描述
SourceKeyArn	解密加密文字的 KMS 密钥的值。
DestinationKeyArn	重新加密数据的 KMS 密钥的值。
操作	每个 Amazon KMS API 操作（在此例中为 ReEncrypt）的值。

SecondsUntilKeyMaterialExpiration

KMS 密钥中导入的密钥材料过期之前剩余的秒数。此指标仅对具有导入的密钥材料（密钥材料来源为 EXTERNAL）以及到期日期的 KMS 密钥有效。

此指标用来跟踪导入的密钥材料在到期之前的剩余有效期。当该时间低于您定义的阈值时，则应重新导入密钥材料并设置新的到期日期，以确保能够使用该 KMS 密钥。SecondsUntilKeyMaterialExpiration 指标特定于单个 KMS 密钥。您不能使用此指标来监控多个 KMS 密钥或将来可能创建的多个 KMS 密钥。有关创建 CloudWatch 警报以监控此指标的帮助，请参阅 针对导入密钥材料的到期创建 CloudWatch 警报。

该指标最有用的统计数据是 Minimum，它会告诉您指定统计周期内所有数据点的最小剩余时间。此指标的唯一有效单位是 Seconds。

维度组名称：Per-Key Metrics（各密钥指标）

`SecondsUntilKeyMaterialExpiration` 的维度

维度	描述；与 Amazon 有关
KeyId	每个 KMS 密钥的值。

当您为 KMS 密钥计划删除时，Amazon KMS 会强制执行一段等待期，然后再删除 KMS 密钥。您可以利用这段等待期来确保现在或将来都不需要 KMS 密钥。您还可以配置 CloudWatch 告警，使其在有人员或应用程序在等待期间试图于加密操作中使用 KMS 密钥时发出警告。如果您收到来自此类告警的通知，您可能需要取消删除 KMS 密钥。

有关说明，请参阅创建检测待删除 KMS 密钥的使用的警报。

CloudHSMKeyStoreThrottle

Amazon KMS 对每个 Amazon CloudHSM 密钥存储中 KMS 密钥的密码操作请求进行节流的数量（响应为 `ThrottlingException`）。此指标仅适用于 Amazon CloudHSM 密钥存储。

`CloudHSMKeyStoreThrottle` 指标仅适用于 Amazon CloudHSM 密钥存储中的 KMS 密钥，并且仅适用于密码操作请求。当请求速率超过 Amazon CloudHSM 密钥存储的自定义密钥存储请求配额时，Amazon KMS 会对这些请求进行节流。此指标还包括 Amazon CloudHSM 集群实施对节流。

维度组名称：Keystore Throttle Metrics（密钥存储限制指标）

维度	描述
CustomKeyStoreId	每个 Amazon CloudHSM 密钥存储的值。
KmsOperation	每个 Amazon KMS API 操作的值。此指标仅适用于 Amazon CloudHSM 密钥存储中 KMS 密钥的密码操作。
KeySpec	每个 KMS 密钥类型的值。对于 Amazon CloudHSM 密钥存储中的 KMS 密钥，唯一受支持的密钥规范是 SYMMETRIC_DEFAULT。

ExternalKeyStoreThrottle

Amazon KMS 限制的每个外部密钥存储中对 KMS 密钥进行加密操作的请求数（以 `ThrottlingException` 响应）。此指标仅适用于外部密钥存储。

ExternalKeyStoreThrottle 指标仅适用于外部密钥存储中的 KMS 密钥，并且仅适用于密码操作请求。当请求率超过外部密钥存储的自定义密钥存储请求配额时，Amazon KMS 会对这些请求进行节流。此指标不包括外部密钥存储代理或外部密钥管理器的节流。

此指标用来检查和调整自定义密钥存储请求配额的值。如果此指标表明 Amazon KMS 频繁对您对这些 KMS 密钥的请求节流，则可以考虑申请上调自定义密钥存储请求限额值。如需帮助，请参阅《服务限额用户指南》中的 申请上调限额。

如果您经常收到 KMSInvalidStateException 错误，其中包含一条说明请求“due to a very high request rate（由于请求率很高）”而被拒绝的消息，或者请求“because the external key store proxy did not respond in time（由于外部密钥存储代理未及时响应）”而被拒绝，则可能表明您的外部密钥管理器或外部密钥存储代理无法跟上当前的请求速率。如可能，请降低您的请求速率。您也可以考虑请求降低自定义密钥存储请求限额值。降低此限额值可能会增加节流的风险（和 ExternalKeyStoreThrottle 指标值），但这表明多余的请求在发送到外部密钥存储代理或外部密钥管理器之前会很快被 Amazon KMS 拒绝。要申请下调限额，请访问 Amazon Web Services 支持 中心 并创建工单。

维度组名称：Keystore Throttle Metrics（密钥存储限制指标）

维度	描述
CustomKeyStoreId	每个外部密钥存储的值。
KmsOperation	每个 Amazon KMS API 操作的值。此指标仅适用于外部密钥存储中 KMS 密钥的密码操作。
KeySpec	每个 KMS 密钥类型的值。对于外部密钥存储中 KMS 密钥的唯一支持的密钥规范是 SYMMETRIC_DEFAULT。

XksProxyCertificateDaysToExpire

距离您的外部密钥存储代理端点（XksProxyUriEndpoint）的 TLS 证书过期的天数。此指标仅适用于外部密钥存储。

使用此指标创建 CloudWatch 警报，该警报会通知您，TLS 证书即将过期。证书过期后，Amazon KMS 将无法与外部密钥存储代理通信。在您续订证书之前，外部密钥存储中所有受 KMS 密钥保护的数据都将不可访问。

证书警报可防止证书过期，证书过期可能会影响您访问加密资源。设置警报，让您的组织有时间在证书过期之前续订证书。

维度组名称：XKS Proxy Certificate Metrics（XKS 代理证书指标）

维度	描述
CustomKeyStoreId	每个外部密钥存储的值。
CertificateName	TLS 证书中的主题名称（CN）。

您可以根据外部密钥存储和外部密钥存储中的 KMS 密钥的指标创建 CloudWatch 警报。有关说明，请参阅监控外部密钥存储。

XksProxyCredentialAge

自当前外部密钥存储代理身份验证凭证（`XksProxyAuthenticationCredential`）与外部密钥存储关联以来的天数。此计数从您在创建或更新外部密钥存储的过程中输入身份验证凭证时开始计算。此指标仅适用于外部密钥存储。

此值旨在提醒您身份验证凭证的使用时间。但是，由于我们从您将凭证与外部密钥存储关联时开始计算，而不是从您在外部密钥存储代理上创建身份验证凭证时开始计算，这可能无法准确地表明代理中的凭证使用时间。

使用此指标创建 CloudWatch 警报，该警报会提醒您轮换外部密钥存储代理身份验证凭证。

维度组名称：Per-Keystore Metrics（各密钥存储指标）

维度	描述
CustomKeyStoreId	每个外部密钥存储的值。

您可以根据外部密钥存储和外部密钥存储中的 KMS 密钥的指标创建 CloudWatch 警报。有关说明，请参阅监控外部密钥存储。

XksProxyErrors

与对外部密钥存储代理的 Amazon KMS 请求相关的异常数量。此计数包括外部密钥存储代理返回至 Amazon KMS 的异常，以及外部密钥存储代理在 250 毫秒超时间隔内未响应 Amazon KMS 时发生的超时错误。此指标仅适用于外部密钥存储。

使用此指标跟踪外部密钥存储中 KMS 密钥的错误率。这可显示最常见的错误，因此您可以确定工程工作的优先顺序。例如，产生较高不可重试错误率的 KMS 密钥，可能表明您的外部密钥存储配置存在问题。若要查看您的外部密钥存储配置，请参阅 查看外部密钥存储。若要编辑您的外部密钥存储设置，请参阅 编辑外部密钥存储属性。

维度组名称：XKS Proxy Error Metrics（XKS 代理错误指标）

维度	描述
CustomKeyStoreId	每个外部密钥存储的值。
KmsOperation	生成对 XKS 代理的请求的每个 Amazon KMS API 操作的值。
XksOperation	每个外部密钥存储代理 API 操作的值。
KeySpec	每个 KMS 密钥类型的值。对于外部密钥存储中 KMS 密钥的唯一支持的密钥规范是 SYMMETRIC_DEFAULT。
ErrorType	值： • 可重试错误：可能是暂时性错误，例如网络错误。 • 不可重试错误：可能表示自定义密钥存储配置或外部组件存在问题。 • 不适用：请求成功；没有错误
ExceptionName	值： • 异常名称 • 无：请求成功；没有错误

您可以根据外部密钥存储和外部密钥存储中的 KMS 密钥的指标创建 CloudWatch 警报。有关说明，请参阅监控外部密钥存储。

XksExternalKeyManagerStates

处于以下各个运行状况状态的外部密钥管理器实例的数量：`Active`、`Degraded` 和 `Unavailable`。此指标的信息来自与每个外部密钥存储关联的外部密钥存储代理。此指标仅适用于外部密钥存储。

以下是与外部密钥存储相关联的外部密钥管理器实例的运行状况。每个外部密钥存储代理都可能会使用不同的指标来衡量外部密钥管理器的运行状况。有关详细信息，请参阅外部密钥存储代理的相关文档。

- `Active`：外部密钥管理器运行正常。
- `Degraded`：外部密钥管理器运行状况不佳，但仍可以传输流量
- `Unavailable`：外部密钥管理器无法传输流量。

使用此指标创建 CloudWatch 警报，该警报会提醒您注意外部密钥管理器实例已降级和不可用。要确定哪些外部密钥管理器实例处于各个状态，请查阅您的外部密钥存储代理日志。

维度组名称：XKS External Key Manager Metrics（XKS 外部密钥管理器指标）

维度	描述
CustomKeyStoreId	每个外部密钥存储的值。
XksExternalKeyManagerState	每个运行状态的值。

您可以根据外部密钥存储和外部密钥存储中的 KMS 密钥的指标创建 CloudWatch 警报。有关说明，请参阅监控外部密钥存储。

XksProxyLatency

外部密钥存储代理响应 Amazon KMS 请求所用的毫秒数。如果请求超时，则记录的值为 250 毫秒的超时限制。此指标仅适用于外部密钥存储。

使用此指标来评估您的外部密钥存储代理和外部密钥管理器的性能。例如，如果代理在加密和解密操作时经常超时，请咨询您的外部代理管理员。

响应缓慢还可能表明您的外部密钥管理器无法处理当前的请求流量。Amazon KMS 建议您的外部密钥管理器每秒能够处理高达 1800 个加密操作请求。如果您的外部密钥管理器无法处理每秒 1800 个请求的速率，请考虑请求降低自定义密钥存储中 KMS 密钥的请求限额。使用外部密钥存储中的 KMS 密钥进行加密操作的请求将采用快速失效机制，并出现节流异常，而不是由外部密钥存储代理或外部密钥管理器处理后拒绝。

维度组名称：XKS Proxy Latency Metrics（XKS 代理延迟指标）

维度	描述
CustomKeyStoreId	每个外部密钥存储的值。
KmsOperation	生成对 XKS 代理的请求的每个 Amazon KMS API 操作的值。
XksOperation	每个外部密钥存储代理 API 操作的值。
KeySpec	每个 KMS 密钥类型的值。对于外部密钥存储中 KMS 密钥的唯一支持的密钥规范是 SYMMETRIC_DEFAULT。

您可以根据外部密钥存储和外部密钥存储中的 KMS 密钥的指标创建 CloudWatch 警报。有关说明，请参阅监控外部密钥存储。

针对导入密钥材料的到期创建 CloudWatch 警报

您可以创建 CloudWatch 警报，在 KMS 密钥中导入的密钥材料即将到期时通知您。例如，警报可以在距离到期时间少于 30 天时通知您。

当您将密钥材料导入 KMS 密钥中时，可以选择性地指定密钥材料的到期时间。当密钥材料过期后，Amazon KMS 将删除密钥材料，并且 KMS 密钥将变为不可用。要再次使用该 KMS 密钥，您必须重新导入密钥材料。但是，如果您在密钥材料到期之前将其重新导入，则可以避免中断使用该 KMS 密钥的进程。

此警报使用 Amazon KMS 发布到 CloudWatch 的 `SecondsUntilKeyMaterialExpires` 指标，其中包含已到期的导入密钥材料的 KMS 密钥。每个警报都使用此指标来监控特定 KMS 密钥的导入密钥材料。您无法为所有具有到期密钥材料的 KMS 密钥创建单个警报，也无法为未来可能创建的 KMS 密钥创建警报。

要求

用于监控已导入密钥材料的到期时间的 CloudWatch 警报需要以下资源。

- 带有已到期导入密钥材料的 KMS 密钥。
- Amazon SNS 主题。有关详细信息，请参阅 Amazon CloudWatch CloudWatch 用户指南中的创建 Amazon SNS 主题。

创建警报

使用以下必填值，按照基于静态阈值创建 CloudWatch 警报中的说明进行操作。对于其他字段，请接受默认值并按要求提供名称。

Field	Value
选择指标	选择 KMS，然后选择每密钥指标。 选择带有 KMS 密钥和 `SecondsUntilKeyMaterialExpires` 指标的行。然后选择 Select metric（选择指标）。 指标列表仅显示导入密钥材料已到期的 KMS 密钥的 `SecondsUntilKeyMaterialExpires` 指标。如果您在账户和区域中没有带这些属性的 KMS 密钥，则此列表为空。
统计数据	最小值
周期	1 minute
阈值类型	静态
当 ...	当 *`metric-name`* 大于 1 时

为外部密钥存储创建 CloudWatch 警报

您可以根据外部密钥存储指标创建 Amazon CloudWatch 警报，以便在指标值超过您指定的阈值时通知您。警报可以将消息发送到亚马逊简单通知服务 (Amazon SNS) Simple Notification Scaling 主题或亚马逊 A EC2 EC2 uto Scaling 政策。有关 CloudWatch 警报的详细信息，请参阅亚马逊 CloudWatch 用户指南中的使用亚马逊 CloudWatch 警报。

在创建亚马逊 CloudWatch 警报之前，您需要一个亚马逊 SNS 主题。有关详情，请参阅亚马逊 CloudWatch 用户指南中的创建 Ama zon SNS 主题。

主题

- 创建证书到期警报
- 创建响应超时警报
- 创建可重试错误警报
- 创建不可重试错误警报

创建证书到期警报

此警报使用 Amazon KMS 发布到的XksProxyCertificateDaysToExpire指标 CloudWatch 来记录与您的外部密钥存储代理端点关联的 TLS 证书的预期到期时间。您无法为账户中的所有外部密钥存储创建单个警报，也无法为将来可能会创建的外部密钥存储创建警报。

我们建议将警报设置为在证书将要过期的前 10 天提醒您，但您应设置最适合您需求的阈值。

创建警报

使用以下必填值按照基于静态阈值创建 CloudWatch 警报中的说明进行操作。对于其他字段，请接受默认值并按要求提供名称。

Field	Value
选择指标	选择 KMS，然后选择 XKS Proxy Certificate Metrics（XKS 代理证书指标）。 选中要监控的 `XksProxyCertificateName` 旁的复选框。 然后选择 Select metric（选择指标）。
Statistic	最小值
周期	5 分钟
阈值类型	静态
当 ...	无论何时`Lower`都XksProxyCertificateDaysToExpire是`10`。

创建响应超时警报

此警报使用 Amazon KMS 发布 CloudWatch到的XksProxyLatency指标来记录外部密钥存储代理响应请求所花费的毫秒数。 Amazon KMS 您无法为账户中的所有外部密钥存储创建单个警报，也无法为将来可能会创建的外部密钥存储创建警报。

Amazon KMS 期望外部密钥存储代理在 250 毫秒内对每个请求做出响应。我们建议设置警报，以在外部密钥存储代理响应时间超过 200 毫秒时提醒您，但您应该设置最适合您需求的阈值。

创建警报

使用以下必填值按照基于静态阈值创建 CloudWatch 警报中的说明进行操作。对于其他字段，请接受默认值并按要求提供名称。

Field	Value
选择指标	选择 KMS，然后选择 XKS Proxy Latency Metrics（XKS 代理延迟指标）。 选中要监控的 `KmsOperation` 旁的复选框。 然后选择 Select metric（选择指标）。
Statistic	平均值
周期	5 分钟
阈值类型	静态
当 ...	无论何时`Greater`都XksProxyLatency是`200`。

创建可重试错误警报

此警报使用 Amazon KMS 发布到的XksProxyErrors指标 CloudWatch 来记录与您的外部密钥存储代理 Amazon KMS 请求相关的异常数量。您无法为账户中的所有外部密钥存储创建单个警报，也无法为将来可能会创建的外部密钥存储创建警报。

可重试错误会降低您的可靠性百分比，并可能表示网络错误。我们建议设置警报，以在一分钟内记录超过五个可重试错误时提醒您，但您应该设置最适合您需求的阈值。

使用以下必填值按照基于静态阈值创建 CloudWatch 警报中的说明进行操作。对于其他字段，请接受默认值并按要求提供名称。

Field	Value
选择指标	选择 Query（查询）选项卡。 对于 Namespace（命名空间），选择 AWS/KMS。 对于 Metric name（指标名称），输入 SUM(XksProxyErrors) 。 对于 Filter by（筛选条件），输入 ErrorType = Retryable 。 选择运行。然后选择 Select metric（选择指标）。
标签	*Retryable errors*
周期	1 minute
阈值类型	静态
当 ...	每当 q1 Greater 5 时。

创建不可重试错误警报

此警报使用 Amazon KMS 发布到的XksProxyErrors指标 CloudWatch 来记录与您的外部密钥存储代理 Amazon KMS 请求相关的异常数量。您无法为账户中的所有外部密钥存储创建单个警报，也无法为将来可能会创建的外部密钥存储创建警报。

不可重试错误可能表示您的外部密钥存储的配置存在问题。我们建议设置警报，以在一分钟内记录超过五个不可重试错误时提醒您，但您应该设置最适合您需求的阈值。

使用以下必填值按照基于静态阈值创建 CloudWatch 警报中的说明进行操作。对于其他字段，请接受默认值并按要求提供名称。

Field	Value
选择指标	选择 Query（查询）选项卡。 对于 Namespace（命名空间），选择 AWS/KMS。 对于 Metric name（指标名称），输入 SUM(XksProxyErrors) 。

Field	Value
	对于 Filter by（筛选条件），输入 `ErrorType = Non-retryable` 。 选择运行。然后选择 Select metric（选择指标）。
标签	*Non-retryable errors*
周期	1 minute
阈值类型	静态
当 ...	每当 `q1 Greater 5` 时。

使用 Amazon EventBridge 来监控 KMS 密钥

您可以使用 Amazon EventBridge（旧称 Amazon CloudWatch Events）提醒您注意 KMS 密钥生命周期中的以下重要事件。

- KMS 密钥中的密钥材料执行自动轮换或按需轮换。
- KMS 密钥中已导入的密钥材料到期。
- 计划删除的 KMS 密钥已被删除。

Amazon KMS 与 Amazon EventBridge 集成，以通知您影响 KMS 密钥的重要事件。每个事件以 JSON（JavaScript 对象表示法）格式表示，包含事件名称、事件发生的日期和时间、受影响的对象等等。您可以收集这些事件，并设立将事件路由到一个或多个目标（如 Amazon Lambda 函数、Amazon SNS 主题、Amazon SQS 队列、Amazon Kinesis Data Streams 中的流或内置目标）的规则。

有关将 EventBridge 与其他类型的事件（包括 Amazon CloudTrail 在记录读/写 API 请求时发出的事件）结合使用的更多信息，请参阅 Amazon EventBridge 用户指南。

以下主题介绍 Amazon KMS 生成的 EventBridge 事件。

KMS CMK 轮换

对于对称加密 KMS 密钥中的密钥材料，Amazon KMS 支持自动轮换和按需轮换。

每当 Amazon KMS 轮换密钥材料时，它会向 EventBridge 发送一个 `KMS CMK Rotation` 事件。Amazon KMS 将尽力生成此事件。

以下是该事件的示例。

```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "KMS CMK Rotation",
  "source": "aws.kms",
  "account": "111122223333",
  "time": "2025-05-23T03:11:54Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  ],
  "detail": {
    "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "key-origin": "AWS_KMS",
    "rotation-type": "ON_DEMAND",
    "previous-key-material-id":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0",
    "current-key-material-id":
 "96083e4fb6dbc41d77578a213a6b6669c044dd4c143e96755396d2bf11fd6068"
  }
}
```

KMS 导入的密钥材料过期

当您将密钥材料导入 KMS 密钥中时，您可以选择性地指定密钥材料的过期时间。当关键材料过期时，Amazon KMS 会删除密钥材料并发送相应的 KMS Imported Key Material Expiration 事件到 EventBridge。Amazon KMS 将尽力生成此事件。

以下是该事件的示例。

```
{
  "version": "0",
  "id": "9da9af57-9253-4406-87cb-7cc400e43465",
  "detail-type": "KMS Imported Key Material Expiration",
  "source": "aws.kms",
  "account": "111122223333",
  "time": "2025-05-23T03:11:54Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
```

```
  ],
  "detail": {
    "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "key-material-id":
 "123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef0"
  }
}
```

KMS CMK 删除

当您为 KMS 密钥计划删除时，Amazon KMS 会强制执行一段等待期，然后再删除 KMS 密钥。等待期结束后，Amazon KMS 将删除 KMS 密钥并发送 `KMS CMK Deletion` 事件到 EventBridge。Amazon KMS 将保证此 EventBridge 事件。由于重试，它可能会在几秒钟内生成多个删除同一 KMS 密钥的事件。

以下是该事件的示例。

```
{
  "version": "0",
  "id": "e9ce3425-7d22-412a-a699-e7a5fc3fbc9a",
  "detail-type": "KMS CMK Deletion",
  "source": "aws.kms",
  "account": "111122223333",
  "time": "2025-05-23T03:11:54Z",
  "region": "us-west-2",
  "resources": [
    "arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
  ],
  "detail": {
    "key-id": "1234abcd-12ab-34cd-56ef-1234567890ab"
  }
}
```

中的别名 Amazon KMS

别名是 Amazon KMS key的友好名称。例如，别名允许您将 KMS 密钥引用为 `test-key`，而不是 `1234abcd-12ab-34cd-56ef-1234567890ab`。

您可以使用别名在 Amazon KMS 控制台、操作和加密DescribeKey操作（例如 E n crypt 和）中标识 KMS 密钥。GenerateDataKey别名还使您能够轻松识别 Amazon 托管式密钥。这些 KMS 密钥的别名始终具有 `aws/<service-name>` 形式。例如，亚马逊 DynamoD Amazon 托管式密钥 B 的别名是。`aws/dynamodb`您可以为项目建立类似的别名标准，例如在别名前加上项目或类别的名称。

您还可以根据 KMS 密钥的别名允许和拒绝访问 KMS 密钥，而无需编辑策略或管理授权。此功能是基于属性的访问控制 (ABAC) Amazon KMS 支持的一部分。有关更多信息，请参阅 使用别名控制对 KMS 密钥的访问。

别名的大部分功能来自于您随时更改与别名关联的 KMS 密钥的能力。别名可以使您的代码更易于编写和维护。例如，假设您使用别名来引用特定 KMS 密钥，并且您想要更改 KMS 密钥。在这种情况下，只需将别名与其他 KMS 密钥关联即可。您不需要更改您的代码。

别名还您更容易在不同 Amazon Web Services 区域中重用相同代码。在多个区域中创建具有相同名称的别名，并将每个别名与其区域中的 KMS 密钥关联。当代码在每个区域中运行时，别名将引用该区域中关联的 KMS 密钥。有关示例，请参阅了解如何在您的应用程序中使用别名。

您可以在 Amazon KMS 控制台中使用 CreateAliasAPI 或使用AWS::KMS::Alias Amazon CloudFormation 模板为 KMS 密钥创建别名。

该 Amazon KMS API 提供对每个账户和区域中的别名的完全控制。API 包括创建别名 (CreateAlias)、查看别名和别名 ARNs (ListAliases)、更改与别名关联的 KMS 密钥 (UpdateAlias) 以及删除别名 (DeleteAlias) 的操作。

别名的作用方式

了解别名在 Amazon KMS中的作用方式。

别名是一种独立的 Amazon 资源

> 别名不是 KMS 密钥的属性。您对别名执行的操作不会影响其关联的 KMS 密钥。您可以为 KMS 密钥创建别名，然后更新别名，使其与其他 KMS 密钥相关联。您甚至可以删除别名，而不会对关联的 KMS 密钥产生任何影响。但是，如果您删除 KMS 密钥，则会删除与该 KMS 密钥关联的所有别名。

如果您在 IAM policy 中指定别名作为资源，则该策略将引用别名，而不是关联的 KMS 密钥。

每个别名都有两种格式

创建别名时，需要指定别名。 Amazon KMS 为您创建别名 ARN。

- 别名 ARN 是唯一标识别名的 Amazon Resource Name (ARN)。

```
# Alias ARN
arn:aws:kms:us-west-2:111122223333:alias/<alias-name>
```

- 别名名称在账户和所在区域中是唯一的。在 Amazon KMS API 中，别名总是以前缀为。`alias/` Amazon KMS 控制台中省略了该前缀。

```
# Alias name
alias/<alias-name>
```

别名不是密钥

别名可以在 CloudTrail 日志和其他输出中以纯文本形式显示。不要在别名名称中包含机密或敏感信息。

每个别名一次与一个 KMS 密钥关联

别名及其 KMS 密钥必须位于同一账户和区域中。

您可以将别名与同一 Amazon Web Services 账户 区域中的任何客户托管密钥相关联。但是，您无权将别名与 Amazon 托管式密钥 关联。

例如，此ListAliases输出显示`test-key`别名恰好与一个目标 KMS 密钥相关联，该密钥由`TargetKeyId`属性表示。

```
{
     "AliasName": "alias/test-key",
     "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/test-key",
     "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
     "CreationDate": 1593622000.191,
     "LastUpdatedDate": 1593622000.191
}
```

多个别名可以与同一 KMS 密钥关联

例如，您可以将 `test-key` 和 `project-key` 别名与同一个 KMS 密钥关联。

```
{
```

```
    "AliasName": "alias/test-key",
    "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/test-key",
    "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "CreationDate": 1593622000.191,
    "LastUpdatedDate": 1593622000.191
},
{
    "AliasName": "alias/project-key",
    "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/project-key",
    "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
    "CreationDate": 1516435200.399,
    "LastUpdatedDate": 1516435200.399
}
```

别名在账户和区域中必须是唯一的

例如，您在每个账户和区域中只能有一个 `test-key` 别名。别名区分大小写，但仅在大小写上不同的别名很容易出错。您不能更改别名名称。但是，您可以删除别名并使用所需名称创建新别名。

您可以在不同的区域中创建具有相同名称的别名。

例如，您可以在美国东部（弗吉尼亚北部）拥有 `finance-key` 别名，在欧洲（法兰克福）拥有 `finance-key` 别名。每个别名都将与其区域中的 KMS 密钥相关联。如果您的代码引用 `alias/finance-key` 之类的别名名称，您可以在多个区域中运行它。在每个区域中，它使用不同的 KMS 密钥。有关更多信息，请参阅 了解如何在您的应用程序中使用别名。

您可以更改与别名关联的 KMS 密钥

您可以使用该UpdateAlias操作将别名与其他 KMS 密钥相关联。例如，如果 `finance-key` 别名与 `1234abcd-12ab-34cd-56ef-1234567890ab` KMS 密钥关联，您可以对其进行更新，使其与 `0987dcba-09fe-87dc-65ba-ab0987654321` KMS 密钥关联。

但是，当前 KMS 密钥和新的 KMS 密钥必须是相同的类型（要么都是对称的，要么都是非对称的，要么都是 HMAC），并且它们必须具有相同的密钥用途（ENCRYPT_DECRYPT、SIGN_VERIFY 或 GENERATE_VERIFY_MAC）。此限制可防止使用别名的代码中出现错误。如果您必须将别名与其他类型的密钥关联，并且您已降低风险，则可以删除并重新创建别名。

有些 KMS 密钥没有别名

在 Amazon KMS 控制台中创建 KMS 密钥时，必须为其指定一个新的别名。但是，使用该CreateKey操作创建 KMS 密钥时，不需要别名。此外，您还可以使用UpdateAlias操作来更改与别名关联的 KMS 密钥，使用该DeleteAlias操作来删除别名。因此，有些 KMS 密钥可能有多个别名，有些可能没有别名。

Amazon 在您的账户中创建别名

Amazon 在您的账户中为Amazon 托管式密钥创建别名。这些别名具有 `alias/aws/<service-name>` 形式的名称，例如 `alias/aws/s3`。

有些 Amazon 别名没有 KMS 密钥。这些预定义的别名通常与您开始使用服务 Amazon 托管式密钥时相关联。

使用别名标识 KMS 密钥

在加密操作中，您可以使用别名或别名 ARN 来标识 KMS 密钥、DescribeKey和。GetPublicKey（如果 KMS 密钥位于不同的 Amazon Web Services 账户中，您必须使用其密钥 ARN 或别名 ARN。）别名不是 KMS 密钥在其他 Amazon KMS 操作中的有效标识符。有关每个 Amazon KMS API 操作的有效密钥标识符的信息，请参阅《AP Amazon Key Management Service l 参考》中对`KeyId`参数的描述。

您不能使用别名名称或别名 ARN 来标识 IAM policy 中的 KMS 密钥。要根据别名控制对 KMS 密钥的访问权限，请使用 k ms: RequestAlias 或 k ms: ResourceAliases 条件密钥。有关更多信息，请参阅 ABAC for Amazon KMS。

控制对别名的访问

创建或更改别名时，会影响别名及其关联的 KMS 密钥。因此，管理别名的委托人必须具有对别名和所有受影响的 KMS 密钥调用别名操作的权限。您可以通过使用密钥策略、IAM policy 和授权来提供这些权限。

> **Note**
>
> 请谨慎授予委托人管理标签和别名的权限。更改标签或别名可以允许或拒绝对客户托管密钥的权限。有关详细信息，请参阅 ABAC for Amazon KMS 和 使用别名控制对 KMS 密钥的访问。

有关控制对所有 Amazon KMS 操作的访问权限的信息，请参见权限参考。

创建和管理别名的权限如下所示。

kms: CreateAlias

要创建别名，委托人需要别名和相关 KMS 密钥的以下权限。

- 别名的 kms:CreateAlias。在附加到允许创建别名的委托人的 IAM policy 中提供此权限。

  以下示例策略语句在 Resource 元素中指定特定别名。但是您可以列出多个别名 ARNs 或指定别名模式，例如 “test*”。您还可以指定 "*" 的 Resource 值以允许委托人在账户和区域中创建任何别名。创建别名的权限也可以包含在对账户和区域中的所有资源的 kms:Create* 权限中。

```
{
  "Sid": "IAMPolicyForAnAlias",
  "Effect": "Allow",
  "Action": [
    "kms:CreateAlias",
    "kms:UpdateAlias",
    "kms:DeleteAlias"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:alias/test-key"
}
```

- KMS 密钥的 kms:CreateAlias。此权限必须在密钥策略或者从密钥策略委派的 IAM policy 中提供。

```
{
  "Sid": "Key policy for 1234abcd-12ab-34cd-56ef-1234567890ab",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:user/KMSAdminUser"},
  "Action": [
    "kms:CreateAlias",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

您可以使用条件键限制可以与别名关联的 KMS 密钥。例如，您可以使用 k ms: KeySpec 条件密钥来允许委托人仅在非对称 KMS 密钥上创建别名。有关您可以用于限制对 KMS 密钥资源的 kms:CreateAlias 权限的条件键完整列表，请参阅 Amazon KMS 权限。

kms: ListAliases

要列出账户和区域中的别名，委托人必须在 IAM policy 中具有 kms:ListAliases 权限。由于此策略与任何特定 KMS 密钥或别名资源无关，因此策略中资源元素的值必须为 "*"。

例如，以下 IAM policy 语句授予委托人列出账户和区域中所有 KMS 密钥和别名的权限。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Action": [
      "kms:ListKeys",
      "kms:ListAliases"
    ],
    "Resource": "*"
  }
}
```

kms: UpdateAlias

要更改与别名关联的 KMS 密钥，委托人需要三个权限元素：一个用于别名，一个用于当前 KMS 密钥，另一个用于新的 KMS 密钥。

例如，假设您想要将 `test-key` 别名从具有密钥 ID 1234abcd-12ab-34cd-56ef-1234567890ab 的 KMS 密钥更改为具有密钥 ID 0987dcba-09fe-87dc-65ba-ab0987654321 的 KMS 密钥。在这种情况下，请包括类似于本部分中的示例的策略语句。

- 别名的 `kms:UpdateAlias`。您可以在附加到委托人的 IAM policy 中提供此权限。以下 IAM policy 指定了一个特定的别名。但是您可以列出多个别名 ARNs 或指定别名模式，例如`"test*"`。您还可以指定 `"*"` 的 `Resource` 值以允许委托人在账户和区域中更新任何别名。

```
{
  "Sid": "IAMPolicyForAnAlias",
  "Effect": "Allow",
  "Action": [
    "kms:UpdateAlias",
    "kms:ListAliases",
    "kms:ListKeys"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:alias/test-key"
}
```

- 当前与别名关联的 KMS 密钥的 `kms:UpdateAlias`。此权限必须在密钥策略或者从密钥策略委派的 IAM policy 中提供。

```
{
  "Sid": "Key policy for 1234abcd-12ab-34cd-56ef-1234567890ab",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:user/KMSAdminUser"},
  "Action": [
    "kms:UpdateAlias",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

- 操作将其与别名关联的 KMS 密钥的 `kms:UpdateAlias`。此权限必须在密钥策略或者从密钥策略委派的 IAM policy 中提供。

```
{
  "Sid": "Key policy for 0987dcba-09fe-87dc-65ba-ab0987654321",
  "Effect": "Allow",
  "Principal": {"AWS": "arn:aws:iam::111122223333:user/KMSAdminUser"},
  "Action": [
    "kms:UpdateAlias",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

您可以使用条件键在 `UpdateAlias` 操作中限制其中一个或两个 KMS 密钥。例如，您可以使用 k ms: ResourceAliases 条件密钥来允许委托人仅在目标 KMS 密钥已具有特定别名时才更新别名。有关您可以用于限制对 KMS 密钥资源的 `kms:UpdateAlias` 权限的条件键完整列表，请参阅 Amazon KMS 权限。

kms: DeleteAlias

要删除别名，委托人需要别名和相关 KMS 密钥的权限。

与往常一样，在授予委托人删除资源的权限时，您应该谨慎操作。但是，删除别名不会影响关联的 KMS 密钥。虽然这可能会导致依赖别名的应用程序出现故障，但如果错误地删除了别名，您可以重新创建别名。

- 别名的 `kms:DeleteAlias`。在附加到允许删除别名的委托人的 IAM policy 中提供此权限。

以下示例策略语句在 Resource 元素中指定别名。但是您可以列出多个别名 ARNs 或指定别名模式，例如"test*"，您也可以将Resource值指定为"*"以允许委托人删除账户和区域中的任何别名。

```
{
  "Sid": "IAMPolicyForAnAlias",
  "Effect": "Allow",
  "Action": [
    "kms:CreateAlias",
    "kms:UpdateAlias",
    "kms:DeleteAlias"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:alias/test-key"
}
```

- 关联的 KMS 密钥的 kms:DeleteAlias。此权限必须在密钥策略或者从密钥策略委派的 IAM policy 中提供。

```
{
  "Sid": "Key policy for 1234abcd-12ab-34cd-56ef-1234567890ab",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::111122223333:user/KMSAdminUser"
  },
  "Action": [
    "kms:CreateAlias",
    "kms:UpdateAlias",
    "kms:DeleteAlias",
    "kms:DescribeKey"
  ],
  "Resource": "*"
}
```

限制别名权限

当资源为 KMS 密钥时，您可以使用条件键限制别名权限。例如，以下 IAM policy 允许对特定账户和区域中的 KMS 密钥执行别名操作。但是，它使用 k ms: KeyOrigin 条件密钥进一步限制对密钥材料的 KMS 密钥的权限 Amazon KMS。

有关您可以用于限制对 KMS 密钥资源的别名权限的条件键完整列表，请参阅 Amazon KMS 权限。

```
{
  "Sid": "IAMPolicyKeyPermissions",
  "Effect": "Allow",
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
  "Action": [
    "kms:CreateAlias",
    "kms:UpdateAlias",
    "kms:DeleteAlias"
  ],
  "Condition": {
    "StringEquals": {
      "kms:KeyOrigin": "AWS_KMS"
    }
  }
}
```

您不能在资源为别名的策略语句中使用条件键。若要限制委托人可以管理的别名，请使用控制对别名访问的 IAM policy 语句的 Resource 元素的值。例如，以下政策声明允许委托人创建、更新或删除 Amazon Web Services 账户 和区域中的任何别名，除非别名以开头Restricted。

```
{
  "Sid": "IAMPolicyForAnAliasAllow",
  "Effect": "Allow",
  "Action": [
    "kms:CreateAlias",
    "kms:UpdateAlias",
    "kms:DeleteAlias"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:alias/*"
},
{
  "Sid": "IAMPolicyForAnAliasDeny",
  "Effect": "Deny",
  "Action": [
    "kms:CreateAlias",
    "kms:UpdateAlias",
    "kms:DeleteAlias"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:alias/Restricted*"
}
```

创建别名

您可以在 Amazon KMS 控制台中创建别名，也可以使用 Amazon KMS API 操作来创建别名。

别名必须为 1-256 个字符的字符串。它只能包含字母数字字符、正斜杠 (/)、下划线 (_) 和连字符 (-)。客户托管密钥的别名名称不能以开头 `alias/aws/` 开头。`alias/aws/` 前缀专门预留供 Amazon 托管式密钥 使用。

您可以为新的 KMS 密钥或现有的 KMS 密钥创建别名。您可以添加别名，以便在项目或应用程序中使用特定 KMS 密钥。

您也可以使用 Amazon CloudFormation 模板为 KMS 密钥创建别名。有关更多信息，请参阅《Amazon CloudFormation 用户指南》中的 AWS::KMS::Alias。

使用控制 Amazon KMS 台

在 Amazon KMS 控制台中创建 KMS 密钥时，必须为新的 KMS 密钥创建别名。要为现有 KMS 密钥创建别名，请使用 KMS 密钥详细信息页面上的 Aliases（别名）选项卡。

1. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/km s 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择客户托管密钥。您无法管理 Amazon 托管式密钥 或 Amazon 拥有的密钥的别名。
4. 在表中，选择 KMS 密钥的密钥 ID 和别名。然后，在 KMS 密钥详细信息页面上，选择 Aliases（别名）选项卡。

   如果 KMS 密钥具有多个别名，则表中的 Aliases（别名）列会显示一个别名和一个别名摘要，例如（再加 n 个）。选择别名摘要将直接进入 KMS 密钥详细信息页面上的 Aliases（别名）选项卡中。
5. 在 Aliases（别名）选项卡上，选择 Create alias（创建别名）。输入别名名称，然后选择 Create alias（创建别名）。

> **Important**
>
> 不要在此字段中包含机密或敏感信息。此字段可能会以纯文本形式显示在 CloudTrail 日志和其他输出中。

> **Note**
>
> 不要添加 alias/ 前缀。控制台会为您自动添加。如果您输入 alias/ExampleAlias，实际的别名名称将是 alias/alias/ExampleAlias。

使用 Amazon KMS API

要创建别名，请使用CreateAlias操作。与在控制台中创建 KMS 密钥的过程不同，该CreateKey操作不会为新的 KMS 密钥创建别名。

> **Important**
>
> 不要在此字段中包含机密或敏感信息。此字段可能会以纯文本形式显示在 CloudTrail 日志和其他输出中。

您可以使用 CreateAlias 操作为没有别名的新 KMS 密钥创建别名。您也可以使用 CreateAlias 操作将别名添加到任何现有 KMS 密钥中或重新创建意外删除的别名。

在 Amazon KMS API 操作中，别名必须以开头，alias/后跟一个名称，例如alias/ExampleAlias。别名在账户和 区域中必须是唯一的。要查找已在使用的别名，请使用ListAliases操作。别名名称区分大小写。

TargetKeyId可以是相同 Amazon Web Services 区域中的任何客户托管密钥。要标识 KMS 密钥，请使用其密钥 ID 或密钥 ARN。您不能使用另一个别名。

以下示例将创建 example-key 别名，并将其与指定的 KMS 密钥相关联。这些示例使用 Amazon Command Line Interface (Amazon CLI)。有关使用多种编程语言的示例，请参阅CreateAlias与 Amazon SDK 或 CLI 配合使用。

```
$ aws kms create-alias \
    --alias-name alias/example-key \
    --target-key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

CreateAlias 不返回任何输出。要查看新别名，请使用 ListAliases 操作。有关更多信息，请参阅 使用 Amazon KMS API。

查找 KMS 密钥的别名和别名 ARN

别名便于在 Amazon KMS 控制台中识别 KMS 密钥。您可以在 Amazon KMS 控制台中或使用ListAliases操作查看 KMS 密钥的别名。该DescribeKey操作返回 KMS 密钥的属性，但不包括别名。

以下过程演示如何使用 Amazon KMS 控制台和 Amazon KMS API 查看和识别与 KMS 密钥关联的别名。 Amazon KMS API 示例使用 Amazon Command Line Interface (Amazon CLI)，但您可以使用任何支持的编程语言。

使用控制 Amazon KMS 台

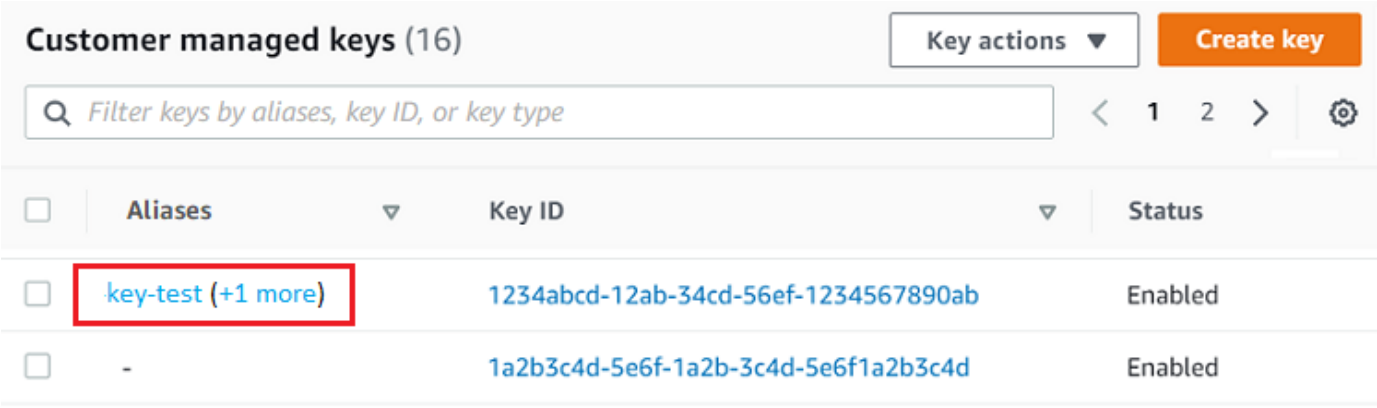
Amazon KMS 控制台显示与 KMS 密钥关联的别名。

1. 在 https://console.aws.amazon.com/km Amazon KMS s 处打开控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 要查看您账户中自己所创建和管理的密钥，请在导航窗格中选择 Customer managed keys (客户托管密钥)。要查看您的账户中为您 Amazon 创建和管理的密钥，请在导航窗格中选择Amazon 托管密钥。
4. Aliases（别名）列显示每个 KMS 密钥的别名。如果 KMS 密钥没有别名，则 Aliases（别名）列中会显示短划线 (-)。

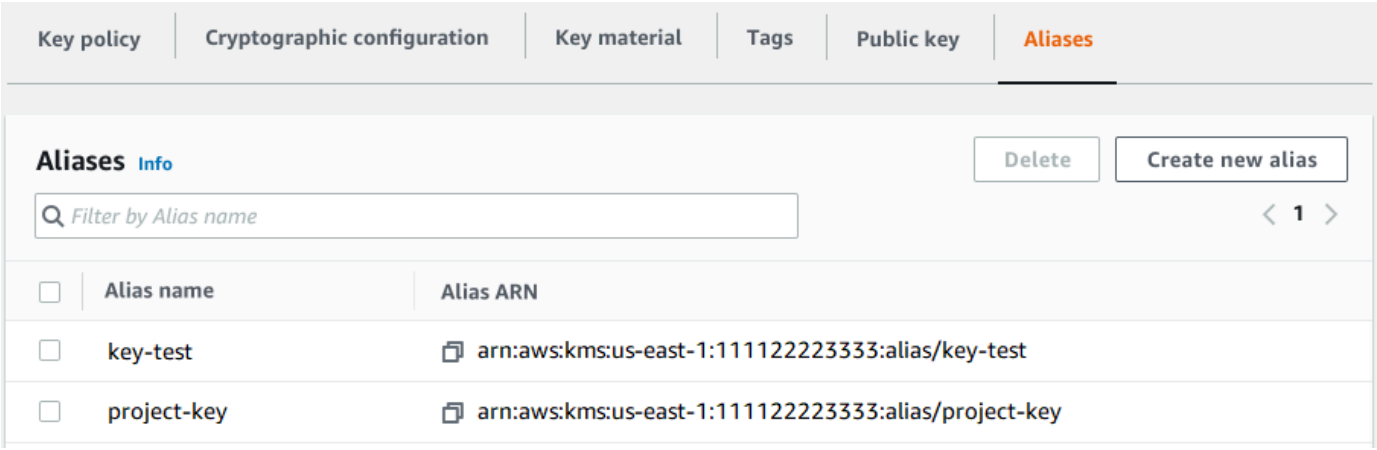
   如果 KMS 密钥具有多个别名，则 Aliases（别名）列还会包含别名摘要，例如（再加 n 个）。例如，以下 KMS 密钥有两个别名，其中一个是 `key-test`。

   要查找 KMS 密钥的所有别名的名称和别名 ARN，请使用 Aliases（别名）选项卡。

   - 要直接转到 Aliases（别名）选项卡上的 Aliases（别名）列中，选择别名摘要（再加 n 个）。仅当 KMS 密钥具有多个别名时，才会显示别名摘要。
   - 或者，选择 KMS 密钥的别名或密钥 ID（这将打开 KMS 密钥的详细信息页面），然后选择 Aliases（别名）选项卡。这些选项卡在 General configuration（常规配置）部分下。

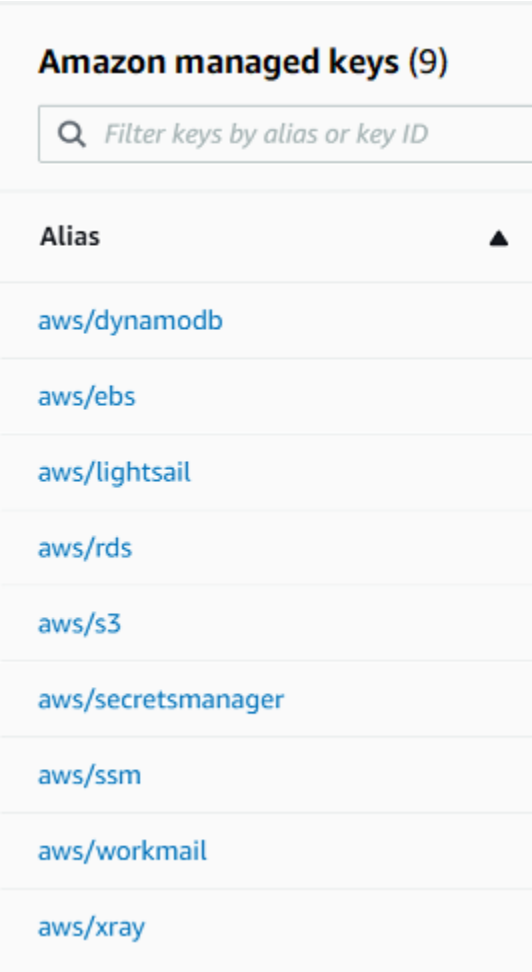


5. Aliases（别名）选项卡显示 KMS 密钥的所有别名的名称和别名 ARN。您还可以在此选项卡上创建和删除 KMS 密钥的别名。



Amazon 托管式密钥

您可以使用别名来识别 Amazon 托管式密钥，如本示例Amazon 托管式密钥页所示。 Amazon 托管式密钥 的别名始终具有以下格式：aws/*<service-name>*。例如，亚马逊 DynamoD Amazon 托管式密钥 B 的别名是。aws/dynamodb



使用 Amazon KMS API

该ListAliases操作返回账户和区域中别名的别名和别名 ARN。输出包括客户托管密钥 Amazon 托管式密钥 和客户托管密钥的别名。 Amazon 托管式密钥 的别名始终具有格式 aws/*<service-name>* ，如 aws/dynamodb。

该响应可能还包括没有 TargetKeyId 字段的别名。这些是 Amazon 已创建但尚未与 KMS 密钥关联的预定义别名。

```
$ aws kms list-aliases
{
    "Aliases": [
        {
            "AliasName": "alias/access-key",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/access-key",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1516435200.399,
            "LastUpdatedDate": 1516435200.399
```

```
        },
        {
            "AliasName": "alias/ECC-P521-Sign",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/ECC-P521-Sign",
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1693622000.704,
            "LastUpdatedDate": 1693622000.704
        },
        {
            "AliasName": "alias/ImportedKey",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/ImportedKey",
            "TargetKeyId": "1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d",
            "CreationDate": 1493622000.704,
            "LastUpdatedDate": 1521097200.235
        },
        {
            "AliasName": "alias/finance-project",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/finance-project",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1604958290.014,
            "LastUpdatedDate": 1604958290.014
        },
        {
            "AliasName": "alias/aws/dynamodb",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/dynamodb",
            "TargetKeyId": "0987ab65-43cd-21ef-09ab-87654321cdef",
            "CreationDate": 1521097200.454,
            "LastUpdatedDate": 1521097200.454
        },
        {
            "AliasName": "alias/aws/ebs",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/ebs",
            "TargetKeyId": "abcd1234-09fe-ef90-09fe-ab0987654321",
            "CreationDate": 1466518990.200,
            "LastUpdatedDate": 1466518990.200
        }
    ]
}
```


要获取与特定 KMS 密钥关联的所有别名，请使用 `ListAliases` 操作的可选 `KeyId` 参数。`KeyId` 参数采用 KMS 密钥的密钥 ID 或者密钥 ARN。

此示例获取与 `0987dcba-09fe-87dc-65ba-ab0987654321` KMS 密钥关联的所有别名。

```
$ aws kms list-aliases --key-id 0987dcba-09fe-87dc-65ba-ab0987654321
{
    "Aliases": [
        {
            "AliasName": "alias/access-key",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/access-key",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": "2018-01-20T15:23:10.194000-07:00",
            "LastUpdatedDate": "2018-01-20T15:23:10.194000-07:00"
        },
        {
            "AliasName": "alias/finance-project",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/finance-project",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
            "CreationDate": 1604958290.014,
            "LastUpdatedDate": 1604958290.014
        }
    ]
}
```

KeyId 参数不采用通配符，但您可以使用编程语言的功能筛选响应。

例如，以下 Amazon CLI 命令仅获取的别名。 Amazon 托管式密钥

```
$ aws kms list-aliases --query 'Aliases[?starts_with(AliasName, `alias/aws/`)]'
```

例如，以下命令仅获取 access-key 别名。别名名称区分大小写。

```
$ aws kms list-aliases --query 'Aliases[?AliasName==`alias/access-key`]'
[
    {
        "AliasName": "alias/access-key",
        "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/access-key",
        "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
        "CreationDate": "2018-01-20T15:23:10.194000-07:00",
        "LastUpdatedDate": "2018-01-20T15:23:10.194000-07:00"
    }
]
```

更新别名

由于别名是独立的资源，因此您可以更改与别名关联的 KMS 密钥。例如，如果test-key别名与一个 KMS 密钥关联，则可以使用该UpdateAlias操作将其与其他 KMS 密钥关联。这是手动轮换 KMS 密钥而不更改其密钥材料的几种方法之一。您还可以更新 KMS 密钥，以使将一个 KMS 密钥用于新资源的应用程序现在使用不同的 KMS 密钥。

您无法在 Amazon KMS 控制台中更新别名。另外，您不能使用 UpdateAlias（或任何其他操作）更改别名名称。要更改别名名称，请删除当前别名，然后为 KMS 密钥创建新的别名。

更新别名时，当前 KMS 密钥和新 KMS 密钥必须为相同类型（均为对称、非对称或 HMAC）。它们还必须拥有相同的密钥用法（ENCRYPT_DECRYPT、SIGN_VERIFY 或 GENERATE_VERIFY_MAC）。此限制可防止使用别名的代码中出现加密错误。

以下示例首先使用ListAliases操作来显示test-key别名当前与 KMS 密钥相关联1234abcd-12ab-34cd-56ef-1234567890ab。

```
$ aws kms list-aliases --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
{
    "Aliases": [
        {
            "AliasName": "alias/test-key",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/test-key",
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "CreationDate": 1593622000.191,
            "LastUpdatedDate": 1593622000.191
        }
    ]
}
```

接下来，它使用 UpdateAlias 操作将与 test-key 别名关联的 KMS 密钥更改为 KMS 密钥 0987dcba-09fe-87dc-65ba-ab0987654321。您不需要指定当前关联的 KMS 密钥，只需指定新的（“目标”）KMS 密钥。别名名称区分大小写。

```
$ aws kms update-alias --alias-name 'alias/test-key' --target-key-id
 0987dcba-09fe-87dc-65ba-ab0987654321
```

要验证别名现在是否已与目标 KMS 密钥关联，请再次使用 ListAliases 操作。此 Amazon CLI 命令使用--query参数仅获取test-key别名。TargetKeyId 和 LastUpdatedDate 字段将更新。

```
$ aws kms list-aliases --query 'Aliases[?AliasName==`alias/test-key`]'
[
    {
        "AliasName": "alias/test-key",
        "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/test-key",
        "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321",
        "CreationDate": 1593622000.191,
        "LastUpdatedDate": 1604958290.154
    }
]
```

删除别名

您可以在 Amazon KMS 控制台中删除别名，也可以使用DeleteAlias操作来删除别名。删除别名之前，请确保别名未使用。虽然删除别名不会影响关联的 KMS 密钥，但它可能会为使用别名的任何应用程序造成问题。如果错误地删除了别名，您可以创建具有相同名称的新别名，并将其与相同或不同的 KMS 密钥关联。

如果您删除 KMS 密钥，则会删除与该 KMS 密钥关联的所有别名。

使用控制 Amazon KMS 台

要在 Amazon KMS 控制台中删除别名，请使用 KMS 密钥详情页面上的 “别名” 选项卡。您可以一次删除一个 KMS 密钥的多个别名。

1. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/km s 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择客户托管密钥。您无法管理 Amazon 托管式密钥 或 Amazon 拥有的密钥的别名。
4. 在表中，选择 KMS 密钥的密钥 ID 和别名。然后，在 KMS 密钥详细信息页面上，选择 Aliases（别名）选项卡。

   如果 KMS 密钥具有多个别名，则表中的 Aliases（别名）列会显示一个别名和一个别名摘要，例如（再加 n 个）。选择别名摘要将直接进入 KMS 密钥详细信息页面上的 Aliases（别名）选项卡中。
5. 在 Aliases（别名）选项卡上，选中要删除的别名旁边的复选框。然后选择删除。

使用 Amazon KMS API

要删除别名，请使用DeleteAlias操作。此操作一次删除一个别名。别名名称区分大小写，且前面必须带有 `alias/` 前缀。

例如，以下命令删除 `test-key` 别名。此命令不返回任何输出。

```
$ aws kms delete-alias --alias-name alias/test-key
```

要验证别名是否已删除，请使用ListAliases操作。以下命令使用中的`--query`参数仅 Amazon CLI 获取`test-key`别名。响应中的空括号表示 `ListAliases` 响应不包含 `test-key` 别名。要消除括号，请使用 `--output text` 参数和值。

```
$ aws kms list-aliases --query 'Aliases[?AliasName==`alias/test-key`]'
[]
```

使用别名控制对 KMS 密钥的访问

您可以根据与 KMS 密钥关联的别名来控制对 KMS 密钥的访问。为此，请使用 kms: RequestAlias 和 kms: ResourceAliases 条件密钥。此功能是基于属性的访问控制 (ABAC) Amazon KMS 支持的一部分。

`kms:RequestAlias` 条件键基于请求中的别名允许或拒绝对 KMS 密钥的访问。`kms:ResourceAliases` 条件键基于与 KMS 密钥关联的别名允许或拒绝对 KMS 密钥的访问。

这些功能不允许您通过使用策略语句的 `resource` 元素中的别名来标识 KMS 密钥。当别名是 `resource` 元素的值时，策略将应用于别名资源，而不是可能与其关联的任何 KMS 密钥。

> **Note**
>
> 标签和别名的更改最多可能需要 5 分钟的时间才能影响 KMS 密钥授权。最近的更改可能会在 API 操作中显示，然后才会影响授权。

使用别名控制对 KMS 密钥的访问权限时，请考虑以下事项：

- 使用别名来强化最低权限访问的最佳实践。仅为 IAM 委托人授予他们对必须使用或管理的 KMS 密钥的所需权限。例如，使用别名标识用于项目的 KMS 密钥。然后授予项目团队仅使用带有项目别名的 KMS 密钥的权限。

- 谨慎为委托人提供 kms:CreateAlias、kms:UpdateAlias 或 kms:DeleteAlias 权限，以允许他们添加、编辑和删除标签。当您使用别名控制对 KMS 密钥的访问时，更改别名可以授予委托人使用他们没有权限使用的 KMS 密钥的权限。它还可以拒绝对其他委托人执行其工作所需的 KMS 密钥的访问。
- 查看您 Amazon Web Services 账户 中当前有权管理别名的委托人，并在必要时调整权限。不具有更改密钥策略或创建授权权限的密钥管理员可以控制对 KMS 密钥的访问，前提是他们有权管理别名。

  例如，控制台密钥管理员的默认密钥策略包括对 kms:CreateAlias、kms:DeleteAlias 和 kms:UpdateAlias 权限。IAM policy 可能会授予对您的 Amazon Web Services 账户中所有 KMS 密钥的别名权限。例如，AWSKeyManagementServicePowerUser托管策略允许委托人创建、删除和列出所有 KMS 密钥的别名，但不能更新它们。
- 在设置依赖于别名的策略之前，请查看您 Amazon Web Services 账户的 KMS 密钥的别名。请确保您的策略仅适用于您要包含的别名。使用CloudTrail 日志和CloudWatch 警报提醒您注意可能影响您的 KMS 密钥访问权限的别名更改。此外，ListAliases响应还包括每个别名的创建日期和上次更新日期。
- 别名策略条件使用模式匹配；它们不绑定到别名的特定实例。使用基于别名的条件键的策略会影响与模式匹配的所有新别名和现有别名。如果删除并重新创建与策略条件匹配的别名，则该条件将应用于新别名，就像对旧别名一样。

kms:RequestAlias 条件键依赖于操作请求中明确指定的别名。kms:ResourceAliases 条件键取决于与 KMS 密钥关联的别名，即使它们未出现在请求中。

kms: RequestAlias

基于标识请求中的 KMS 密钥的别名，允许或拒绝对 KMS 密钥的访问。您可以在密钥策略或 IAM 策略中使用 kms: RequestAlias 条件密钥。它适用于在请求中使用别名标识 KMS 密钥的操作，即加密操作DescribeKey、和GetPublicKey。它对别名操作无效，例如CreateAlias或DeleteAlias。

在条件键中，指定别名名称或别名名称模式。您不能指定别名 ARN。

例如，以下密钥策略语句允许委托人在 KMS 密钥上使用指定的操作。权限仅在请求使用包含可标识 KMS 密钥的 alpha 别名时有效。

```
{
  "Sid": "Key policy using a request alias condition",
  "Effect": "Allow",
  "Principal": {
```

```
    "AWS": "arn:aws:iam::111122223333:role/alpha-developer"
  },
  "Action": [
    "kms:Decrypt",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:RequestAlias": "alias/*alpha*"
    }
  }
}
```

以下来自授权委托人的示例请求将满足条件。但是，使用密钥 ID、密钥 ARN 或者其他别名的请求将无法满足条件，即使这些值标识了相同的 KMS 密钥。

```
$ aws kms describe-key --key-id "arn:aws:kms:us-west-2:111122223333:alias/project-
alpha"
```

kms: ResourceAliases

基于与 KMS 密钥关联的别名允许或拒绝访问 KMS 密钥，即使请求中未使用别名也是如此。k ms: ResourceAliases 条件密钥允许您指定别名或别名模式，例如`alias/test*`，这样您就可以在 IAM 策略中使用它来控制对同一区域中多个 KMS 密钥的访问权限。它对任何使用 KMS 密钥的 Amazon KMS 操作都有效。

例如，以下 IAM 策略允许委托人分两 Amazon Web Services 账户次调用 KMS 密钥的指定操作。但是，该权限仅适用于与开头为 `restricted` 的别名相关联的 KMS 密钥。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AliasBasedIAMPolicy",
      "Effect": "Allow",
      "Action": [
```

```
        "kms:Decrypt",
        "kms:GenerateDataKey*",
        "kms:DescribeKey"
      ],
      "Resource": [
        "arn:aws:kms:*:111122223333:key/*",
        "arn:aws:kms:*:444455556666:key/*"
      ],
      "Condition": {
        "ForAnyValue:StringLike": {
          "kms:ResourceAliases": "alias/restricted*"
        }
      }
    }
  ]
}
```

kms:ResourceAliases 条件是资源的条件，而不是请求的。因此，未指定别名的请求仍然可以满足条件。

以下示例请求（指定匹配别名）满足条件。

```
$ aws kms enable-key-rotation --key-id "alias/restricted-project"
```

但是，下面的示例请求也满足条件，前提是指定的 KMS 密钥具有以 restricted 开头的别名，即使该别名未在请求中使用。

```
$ aws kms enable-key-rotation --key-id "1234abcd-12ab-34cd-56ef-1234567890ab"
```

了解如何在您的应用程序中使用别名

您可以使用别名在应用程序代码中表示 KMS 密钥。 Amazon KMS 加密运算中的KeyId参数DescribeKey、和GetPublicKey接受别名或别名 ARN。

例如，以下 GenerateDataKey 命令使用别名名称 (alias/finance) 来标识 KMS 密钥。别名名称是 KeyId 参数的值。

```
$ aws kms generate-data-key --key-id alias/finance --key-spec AES_256
```

如果 KMS 密钥位于其他密钥中 Amazon Web Services 账户，则必须在这些操作中使用密钥 ARN 或别名 ARN。使用别名 ARN 时，请记住 KMS 密钥的别名是在拥有 KMS 密钥的账户中定义的，并且在每个区域中可能会有所不同。有关查找别名 ARN 的帮助，请参阅 查找 KMS 密钥的别名和别名 ARN。

例如，以下 GenerateDataKey 命令使用不在调用者账户中的 KMS 密钥。ExampleAlias 别名与指定账户和区域中的 KMS 密钥相关联。

```
$ aws kms generate-data-key --key-id arn:aws:kms:us-west-2:444455556666:alias/
ExampleAlias --key-spec AES_256
```

别名的最强大用途之一是在多个 Amazon Web Services 区域中运行的应用程序中。例如，您可能有一个使用 RSA 非对称 KMS 密钥进行签名和验证的全局应用程序。

- 在美国西部（俄勒冈）(us-west-2) 区域中，您需要使用 arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab。
- 在欧洲（法兰克福）(eu-central-1) 区域中，您需要使用 arn:aws:kms:eu-central-1:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321
- 在亚太地区（新加坡）(ap-southeast-1) 区域中，您需要使用 arn:aws:kms:ap-southeast-1:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d。

您可以在每个区域中创建不同版本的应用程序，也可以使用字典或 switch 语句为每个区域选择正确的 KMS 密钥。但在每个区域中创建具有相同别名名称的别名要容易得多。请记住，别名名称区分大小写。

```
aws --region us-west-2 kms create-alias \
    --alias-name alias/new-app \
    --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab

aws --region eu-central-1 kms create-alias \
    --alias-name alias/new-app \
    --key-id arn:aws:kms:eu-central-1:111122223333:key/0987dcba-09fe-87dc-65ba-
ab0987654321

aws --region ap-southeast-1 kms create-alias \
    --alias-name alias/new-app \
    --key-id arn:aws:kms:ap-
southeast-1:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d
```

然后，在代码中使用别名。当代码在每个区域中运行时，别名将引用该区域中关联的 KMS 密钥。例如，此代码将使用别名名称调用 Sign 操作。

```
aws kms sign --key-id alias/new-app \
    --message $message \
    --message-type RAW \
    --signing-algorithm RSASSA_PSS_SHA_384
```

但是，别名可能有被删除或更新为与其他 KMS 密钥关联的风险。在这种情况下，应用程序尝试使用别名名称验证签名将失败，您可能需要重新创建或更新别名。

要降低此风险，请谨慎授予委托人管理您在应用程序中使用的别名的权限。有关更多信息，请参阅 控制对别名的访问。

对于在多个 Amazon Web Services 区域中加密数据的应用程序，还有几个其他解决方案，包括 Amazon Encryption SDK。

在日志中 Amazon CloudTrail 查找别名

在 Amazon KMS API 操作 Amazon KMS key 中，您可以使用别名来表示。当您这样做时，KMS 密钥的别名和密钥 ARN 将记录在事件的 Amazon CloudTrail 日志条目中。别名显示在 `requestParameters` 字段中。密钥 ARN 显示在 `resources` 字段中。即使某项 Amazon 服务 Amazon 托管式密钥 在您的账户中使用了，也是如此。

例如，以下GenerateDataKey请求使用`project-key`别名表示 KMS 密钥。

```
$ aws kms generate-data-key --key-id alias/project-key --key-spec AES_256
```

当此请求记录在 CloudTrail 日志中时，日志条目包括所使用的实际 KMS 密钥的别名和密钥 ARN。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "ABCDE",
        "arn": "arn:aws:iam::111122223333:role/ProjectDev",
        "accountId": "111122223333",
        "accessKeyId": "FFHIJ",
        "userName": "example-dev"
    },
```

```
    "eventTime": "2020-06-29T23:36:41Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "205.205.123.000",
    "userAgent": "aws-cli/1.18.89 Python/3.6.10
 Linux/4.9.217-0.1.ac.205.84.332.metal1.x86_64 botocore/1.17.12",
    "requestParameters": {
        "keyId": "alias/project-key",
        "keySpec": "AES_256"
    },
    "responseElements": null,
    "requestID": "d93f57f5-d4c5-4bab-8139-5a1f7824a363",
    "eventID": "d63001e2-dbc6-4aae-90cb-e5370aca7125",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

有关在 CloudTrail 日志中记录 Amazon KMS 操作的详细信息，请参阅使用 Amazon CloudTrail 记录 Amazon KMS API 调用。

标签在 Amazon KMS

标签是您可以为 Amazon 资源分配（或 Amazon 可以分配）的可选元数据标签。每个标签都包含一个标签键和一个标签值，它们都是区分大小写的字符串。标签值可为空 (null) 字符串。资源上的每个标签必须具有不同的标签密钥，但您可以为多个 Amazon 资源添加相同的标签。每个资源最多可以有 50 个用户创建的标签。

不要在标签键或标签值中包含机密或敏感信息。许多人都可以访问标签 Amazon Web Services 服务，包括账单。

在中 Amazon KMS，您可以在创建 KMS 密钥时向客户托管密钥添加标签，也可以标记或取消标记现有 KMS 密钥，除非这些密钥处于待删除状态。您不能在其他 Amazon Web Services 账户中标记别名、自定义密钥存储库 Amazon 托管式密钥、Amazon 拥有的密钥、或 KMS 密钥。标签是可选的，但它们可能非常有用。

例如，您可以添加一个 `"Project"="Alpha"` 标签添加到您用于 Alpha 项目的所有 KMS 密钥和 Amazon S3 存储桶。

```
TagKey   = "Project"
TagValue = "Alpha"
```

有关标签的一般信息，包括格式和语法，请参阅中的Amazon Web Services 一般参考标记 Amazon 资源。

标签可帮助您：

- 识别和整理您的 Amazon 资源。许多 Amazon 服务都支持标记，因此您可以为来自不同服务的资源分配相同的标签，以表明这些资源是相关的。例如，您可以为 KMS 密钥和亚马逊弹性区块存储 (Amazon EBS) 存储卷或密钥分配相同的标签。 Amazon Secrets Manager 您还可以使用标签来标识 KMS 密钥以实现自动化。
- 追踪您的 Amazon 成本。向 Amazon 资源添加标签时， Amazon 会生成一份成本分配报告，其中包含按标签汇总的使用量和成本。您可以使用此功能来跟踪 Amazon KMS 项目、应用程序或成本中心的成本。

  有关对成本分配使用标签的更多信息，请参阅 Amazon Billing 用户指南中的使用成本分配标签。有关适用于标签键和标签值的规则的信息，请参阅 Amazon Billing 用户指南中的用户定义的标签限制。

- 控制对 Amazon 资源的访问权限。根据标签允许和拒绝访问 KMS 密钥是 Amazon KMS 支持基于属性的访问控制 (ABAC) 的一部分。有关 Amazon KMS keys 根据标签控制访问权限的信息，请参阅使用标签控制对 KMS 密钥的访问。有关使用标签控制 Amazon 资源访问权限的更多一般信息，请参阅 IAM 用户指南中的使用 Amazon 资源标签控制对资源的访问权限。

Amazon KMS 当您使用、或ListResourceTags操作时 TagResource，UntagResource会在 Amazon CloudTrail 日志中写入一个条目。

主题

- 控制对标签的访问
- 向 KMS 密钥添加标签
- 编辑与 KMS 密钥关联的标签
- 删除与 KMS 密钥关联的标签
- 查看与 KMS 密钥关联的标签
- 使用标签控制对 KMS 密钥的访问

控制对标签的访问

要在 Amazon KMS 控制台中或使用 API 添加、查看和删除标签，委托人需要标记权限。您可以在密钥策略中提供这些权限。您还可以在 IAM policy（包括 VPC 端点策略）中提供这些权限，但仅当密钥策略允许时。AWSKeyManagementServicePowerUser托管策略允许委托人对账户可以访问的所有 KMS 密钥进行标记、取消标记和列出标签。

您也可以通过对标签使用 Amazon 全局条件键来限制这些权限。在中 Amazon KMS，这些条件可以控制对标记操作（例如TagResource和 UntagResource）的访问权限。

> **Note**
>
> 请谨慎授予委托人管理标签和别名的权限。更改标签或别名可以允许或拒绝对客户托管密钥的权限。有关详细信息，请参阅 ABAC for Amazon KMS 和 使用标签控制对 KMS 密钥的访问。

有关示例策略和更多信息，请参阅 IAM 用户指南中的根据标签键控制访问。

用于创建和管理标签的权限如下所示。

kms: TagResource

允许委托人添加或编辑标签。要在创建 KMS 密钥时添加标签，委托人必须在 IAM policy 中具有不限于特定 KMS 密钥的权限。

kms: ListResourceTags

允许委托人查看 KMS 密钥上的标签。

kms: UntagResource

允许委托人从 KMS 密钥中删除标签。

标记策略中的权限

您可以在密钥策略或 IAM policy 中提供权限标记。例如，以下示例密钥策略向选定用户授予标记 KMS 密钥的权限。它为所有可以担任示例管理员或开发人员角色的用户授予查看标签的权限。

JSON

```
{
  "Version":"2012-10-17",
  "Id": "example-key-policy",
  "Statement": [
    {
      "Sid": "EnableIAMUserPermissions",
      "Effect": "Allow",
      "Principal": {"AWS": "arn:aws:iam::111122223333:root"},
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "AllowAllTaggingPermissions",
      "Effect": "Allow",
      "Principal": {"AWS": [
        "arn:aws:iam::111122223333:user/LeadAdmin",
        "arn:aws:iam::111122223333:user/SupportLead"
      ]},
      "Action": [
          "kms:TagResource",
          "kms:ListResourceTags",
          "kms:UntagResource"
      ],
```

```
      "Resource": "*"
    },
    {
      "Sid": "AllowRolesViewTags",
      "Effect": "Allow",
      "Principal": {"AWS": [
        "arn:aws:iam::111122223333:role/Administrator",
        "arn:aws:iam::111122223333:role/Developer"
      ]},
      "Action": "kms:ListResourceTags",
      "Resource": "*"
    }
  ]
}
```

要授予委托人对多个 KMS 密钥的标记权限，您可以使用 IAM policy。为使此策略生效，每个 KMS 密钥的密钥策略都必须允许账户使用 IAM policy 来控制对 KMS 密钥的访问。

例如，以下 IAM policy 允许委托人创建 KMS 密钥。它还允许他们在指定账户中的所有 KMS 密钥上创建和管理标签。这种组合允许委托人在创建 KMS 密钥时使用CreateKey操作的 Tags 参数向 KMS 密钥添加标签。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPolicyCreateKeys",
      "Effect": "Allow",
      "Action": "kms:CreateKey",
      "Resource": "*"
    },
    {
      "Sid": "IAMPolicyTags",
      "Effect": "Allow",
      "Action": [
        "kms:TagResource",
        "kms:UntagResource",
        "kms:ListResourceTags"
      ],
```

```
      "Resource": "arn:aws:kms:*:111122223333:key/*"
    }
  ]
}
```

限制标签权限

您可以通过使用策略条件限制标记权限。以下策略条件可应用于 kms:TagResource 和 kms:UntagResource 权限。例如，您可以使用 aws:RequestTag/*tag-key* 条件来允许委托人仅添加特定标签，或阻止委托人添加具有特定标签键的标签。或者，您可以使用 kms:KeyOrigin 条件来防止委托人标记或取消标记具有导入密钥材料的 KMS 密钥。

- aws：RequestTag
- a@@ ws:ResourceTag/tag-key（仅限 IAM 策略）
- aws：TagKeys
- kms: CallerAccount
- kms: KeySpec
- kms: KeyUsage
- kms: KeyOrigin
- kms: ViaService

作为使用标签控制对 KMS 密钥的访问的最佳实践，请使用 aws:RequestTag/*tag-key* 或 aws:TagKeys 条件键来确定允许哪些标签（或标签键）。

例如，以下 IAM policy 与上一个类似。但是，此策略允许主体为具有 Project 标签键的标签创建标签 (TagResource) 并删除标签 UntagResource。

由于TagResource和UntagResource请求可以包含多个标签，因此您必须使用 aws: TagKeys 条件指定ForAllValues或ForAnyValue设置运算符。ForAnyValue 运算符要求请求中至少有一个标签键与策略中的其中一个标签键匹配。ForAllValues 运算符要求请求中所有的标签键与策略中的其中一个标签键匹配。true如果请求中没有标签，则ForAllValues运算符也会返回，但 TagResource 如果未指定标签，则会 UntagResource失败。有关集合运算符的详细信息，请参阅 IAM 用户指南中的使用多个键和值。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPolicyCreateKey",
      "Effect": "Allow",
      "Action": "kms:CreateKey",
      "Resource": "*"
    },
    {
      "Sid": "IAMPolicyViewAllTags",
      "Effect": "Allow",
      "Action": "kms:ListResourceTags",
      "Resource": "arn:aws:kms:*:111122223333:key/*"
    },
    {
      "Sid": "IAMPolicyManageTags",
      "Effect": "Allow",
      "Action": [
        "kms:TagResource",
        "kms:UntagResource"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
          "ForAllValues:StringEquals": {"aws:TagKeys": "Project"}
      }
    }
  ]
}
```

向 KMS 密钥添加标签

标签有助于识别和整理您的 Amazon 资源。您可以在创建 KMS 密钥时向客户托管密钥添加标签，或者向现有的 KMS 密钥添加标签。您无法标记 Amazon 托管式密钥。

以下过程演示如何使用 Amazon KMS 控制台和 Amazon KMS API 向客户托管密钥添加标签。Amazon KMS API 示例使用 Amazon Command Line Interface (Amazon CLI)，但您可以使用任何支持的编程语言。

主题

创建 KMS 密钥时添加标签

在使用 Amazon KMS 控制台或CreateKey操作创建密钥时，可以向 KMS 密钥添加标签。要在创建 KMS 密钥时添加标签，除了创建 KMS 密钥所需的权限之外，您还必须具有 IAM 策略中的 `kms:TagResource` 权限。权限至少必须涵盖账户和区域中的所有 KMS 密钥。有关更多信息，请参阅控制对标签的访问。

使用控制 Amazon KMS 台

要在控制台中创建 KMS 密钥时添加标签，除了添加标签和创建 KMS 密钥所需的权限之外，您还必须具有在控制台中查看 KMS 密钥所需的权限。权限至少必须涵盖账户和区域中的所有 KMS 密钥。

1. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/km s 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择客户托管密钥。（您无法管理 Amazon 托管式密钥的标签）
4. 选择密钥类型，然后选择 Next（下一步）。
5. 输入别名和可选的描述。
6. 输入一个标签键和可选的标签值。要添加其他标签，请选择 Add tag（添加标签）。要删除标签，请选择 Remove（删除）。完成新 KMS 密钥的标记后，选择 Next（下一步）。
7. 完成 KMS 密钥的创建。

使用 Amazon KMS API

要在使用CreateKey操作创建密钥时指定标签，请使用操作的`Tags`参数。

`CreateKey` 的 `Tags` 参数的值是区分大小写的标签键和标签值对的集合。KMS 密钥上的每个标签都必须具有不同的标签名称。标签值可为 null 或空字符串。

例如，以下 Amazon CLI 命令创建带有`Project:Alpha`标签的对称加密 KMS 密钥。指定多个键值对时，请使用空格分隔每个对。

```
$ aws kms create-key --tags TagKey=Project,TagValue=Alpha
```

当此命令成功时，它会返回一个 KeyMetadata 对象以及有关新 KMS 密钥的信息。但是，KeyMetadata 不包括标签。要获取标签，请使用ListResourceTags操作。

向现有的 KMS 密钥添加标签

您可以在 Amazon KMS 控制台中或使用TagResource操作为现有的客户托管 KMS 密钥添加标签。要添加标签，您需要具有向 KMS 密钥添加标签的权限。您可以从 KMS 密钥的密钥策略中获取此权限，或者如果密钥策略允许，还可以从包含 KMS 密钥的 IAM policy 获取此权限。

使用控制 Amazon KMS 台

1. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/km s 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择客户托管密钥。（您无法管理 Amazon 托管式密钥的标签）
4. 您可以使用表筛选条件仅显示带有特定标签的 KMS 密钥。有关详细信息，请参阅使用 Amazon KMS 控制台查看标签。
5. 选中 KMS 密钥的别名旁的复选框。
6. 选择 Key actions、Add or edit tags。
7. 在 KMS 密钥的详细信息页面上，选择 Tags（标签）选项卡。
   - 要创建您的第一个标签，请选择 Create tag（创建标签），键入标签键（必需）和标签值（可选），然后选择 Save（保存）。

     如果将标签值留空，则实际标签值为 null 或空字符串。
   - 要添加标签，请选择 Edi（编辑），选择 Add tag（添加标签），键入标签键和标签值，然后选择 Save（保存）。
8. 要保存您的更改，请选择保存更改。

使用 Amazon KMS API

该TagResource操作将向 KMS 密钥添加一个或多个标签。此操作不能用于在不同的 Amazon Web Services 账户中添加标签。您也可以使用该 TagResource 操作来编辑现有标签。有关更多信息，请参阅 the section called “编辑标签”。

要添加标签，请指定新标签键和标签值。KMS 密钥上的每个标签都必须具有不同的标签键。标签值可为 null 或空字符串。

例如，以下命令将 **Purpose** 和 **Department** 标签添加到示例 KMS 密钥中。

```
$ aws kms tag-resource \
        --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
        --tags TagKey=Purpose,TagValue=Pretest TagKey=Department,TagValue=Finance
```

此命令成功执行后，不会返回任何输出。要查看 KMS 密钥上的标签，请使用ListResourceTags操作。

编辑与 KMS 密钥关联的标签

标签有助于识别和整理您的 Amazon 资源。您可以在 Amazon KMS 控制台中或使用TagResource操作来编辑与客户托管的 KMS 密钥关联的标签。您无法编辑的标签 Amazon 托管式密钥。

以下过程演示如何编辑与 KMS 密钥关联的标签。 Amazon KMS API 示例使用 Amazon Command Line Interface （ Amazon CLI ），但您可以使用任何受支持的编程语言。

使用控制 Amazon KMS 台

1. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/km s 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择客户托管密钥。（您无法编辑的标签 Amazon 托管式密钥）
4. 您可以使用表筛选条件仅显示带有特定标签的 KMS 密钥。有关详细信息，请参阅使用 Amazon KMS 控制台查看标签。
5. 选中 KMS 密钥的别名旁的复选框。
6. 选择 Key actions、Add or edit tags。
7. 在 KMS 密钥的详细信息页面上，选择 Tags（标签）选项卡。
   - 要更改标签的名称或值，请选择 Edit (编辑)，进行更改，然后选择 Save (保存)。
8. 要保存您的更改，请选择保存更改。

使用 Amazon KMS API

该TagResource操作向客户托管密钥添加一个或多个标签；。但是，您也可以使用 TagResource 来更改现有标签的标签值。此操作不能用于添加或编辑不同 Amazon Web Services 账户中的标签。

要编辑标签，请指定现有标签键和新标签值。KMS 密钥上的每个标签都必须具有不同的标签键。标签值可为 null 或空字符串。

例如，此命令会将 Purpose 标签的值从 Pretest 更改为 Test。

```
$ aws kms tag-resource \
        --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
        --tags TagKey=Purpose,TagValue=Test
```

删除与 KMS 密钥关联的标签

标签有助于识别和整理您的 Amazon 资源。您可以在 Amazon KMS 控制台中或使用UntagResource操作删除与客户托管的 KMS 密钥关联的标签。您无法编辑或移除的标签 Amazon 托管式密钥。

以下过程演示了如何删除 KMS 密钥的标签。 Amazon KMS API 示例使用 Amazon Command Line Interface （ Amazon CLI ），但您可以使用任何受支持的编程语言。

使用控制 Amazon KMS 台

1. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/km s 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择客户托管密钥。（ 您无法管理 Amazon 托管式密钥的标签 ）
4. 您可以使用表筛选条件仅显示带有特定标签的 KMS 密钥。有关详细信息，请参阅使用 Amazon KMS 控制台查看标签。
5. 选中 KMS 密钥的别名旁的复选框。
6. 选择 Key actions、Add or edit tags。
7. 在 KMS 密钥的详细信息页面上，选择 Tags（ 标签 ）选项卡。
   - 要删除标签，请选择 Edit (编辑)。在标签行上，选择 Remove (删除)，然后选择 Save (保存)。
8. 要保存您的更改，请选择保存更改。

使用 Amazon KMS API

该UntagResource操作会从 KMS 密钥中删除标签。要标识要删除的标签，请指定标签键。此操作不能用于从其他 Amazon Web Services 账户中的 KMS 密钥中删除标签。

当它成功时，UntagResource 操作不返回任何输出。此外，如果在 KMS 密钥上未找到指定的标签键，则不会抛出异常或返回响应。要确认操作是否奏效，请使用该ListResourceTags操作。

例如，此命令将从指定的 KMS 密钥中删除 **Purpose** 标签及其值。

```
$ aws kms untag-resource --key-id 1234abcd-12ab-34cd-56ef-1234567890ab --tag-keys
 Purpose
```

查看与 KMS 密钥关联的标签

标签有助于识别和整理您的 Amazon 资源。您可以在 Amazon KMS 控制台中或使用ListResourceTags操作查看与客户托管的 KMS 密钥关联的标签。

以下过程演示如何查找与特定 KMS 密钥关联的标签。 Amazon KMS API 示例使用 Amazon Command Line Interface (Amazon CLI)，但您可以使用任何支持的编程语言。

使用控制 Amazon KMS 台

1. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/km s 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择客户托管密钥。（您无法管理 Amazon 托管式密钥的标签）
4. 您可以使用表筛选条件仅显示带有特定标签的 KMS 密钥。

   要仅显示具有特定标签的 KMS 密钥，请选择筛选框，选择标签键，然后从实际标签值中进行选择。还可以键入全部或部分标签值。

   生成的表格将显示带有所选标签的所有 KMS 密钥。但是，它不显示标签。要查看标签，请选择 KMS 密钥的密钥 ID 或别名，然后在其详细信息页面上选择 Tags（标签）选项卡。别名显示在 General configuration（常规配置）部分下。

   此筛选条件同时需要标签键和标签值。它不会通过仅键入标签键或仅键入标签值来找到 KMS 密钥。要按全部或部分标签键或值筛选标签，请使用ListResourceTags操作获取带标签的 KMS 密钥，然后使用您的编程语言的筛选功能。
5. 选中 KMS 密钥的别名旁的复选框。
6. 选择 Key actions、Add or edit tags。
7. 在 KMS 密钥的详细信息页面上，选择 Tags（标签）选项卡。

使用 Amazon KMS API

该ListResourceTags操作获取 KMS 密钥的标签。`KeyId` 参数是必需的。此操作不能用于查看其他 Amazon Web Services 账户中的 KMS 密钥上的标签。

例如，以下命令获取示例 KMS 密钥的标签。

```
$ aws kms list-resource-tags --key-id 1234abcd-12ab-34cd-56ef-1234567890ab

  "Truncated": false,
  "Tags": [
      {
        "TagKey": "Project",
        "TagValue": "Alpha"
     },
     {
       "TagKey": "Purpose",
       "TagValue": "Test"
     },
     {
       "TagKey": "Department",
       "TagValue": "Finance"
     }
  ]
}
```

使用标签控制对 KMS 密钥的访问

您可以 Amazon KMS keys 根据 KMS 密钥上的标签控制对的访问权限。例如，您可以编写 IAM policy，以允许委托人仅启用和禁用具有特定标签的 KMS 密钥。或者，您可以使用 IAM policy 防止委托人在加密操作中使用 KMS 密钥，除非 KMS 密钥具有特定标签。

此功能是基于属性的访问控制 (ABAC) Amazon KMS 支持的一部分。有关使用标签控制 Amazon 资源访问的信息，请参阅 ABAC 有什么用 Amazon？ 以及使用 IAM 用户指南中的资源标签控制对资源的访问权限。 Amazon 要获取有关解决与 ABAC 相关的访问问题的帮助，请参阅 对 ABAC 进行故障排除 Amazon KMS。

> **Note**
>
> 标签和别名的更改最多可能需要 5 分钟的时间才能影响 KMS 密钥授权。最近的更改可能会在 API 操作中显示，然后才会影响授权。

Amazon KMS 支持 aws:ResourceTag/tag-key 全局条件上下文密钥，它允许您根据 KMS 密钥上的标签控制对 KMS 密钥的访问。由于多个 KMS 密钥可以具有相同的标签，此功能可使您将权限应用于一组选定的 KMS 密钥。您还可以通过更改 KMS 密钥的标签轻松更改集合中的 KMS 密钥。

在中 Amazon KMS，`aws:ResourceTag/tag-key`条件密钥仅在 IAM 策略中受支持。密钥策略（仅适用于一个 KMS 密钥）或不使用特定 KMS 密钥的操作（例如ListKeys或ListAliases操作）不支持它。

使用标签控制访问提供了一种简单、可扩展且灵活的方式来管理权限。但是，如果设计和管理不当，它可能会无意中允许或拒绝对您的 KMS 密钥的访问。如果您使用标签来控制访问，请考虑以下做法。

- 使用标签来强化最低权限访问的最佳实践。仅为 IAM 委托人授予他们对必须使用或管理的 KMS 密钥的所需权限。例如，使用标签来标记用于项目的 KMS 密钥。然后授予项目团队仅使用带有项目标签的 KMS 密钥的权限。
- 谨慎为委托人提供 `kms:TagResource` 和 `kms:UntagResource` 权限，以允许他们添加、编辑和删除标签。当您使用标签控制对 KMS 密钥的访问时，更改标签可以授予委托人使用他们没有权限使用的 KMS 密钥的权限。它还可以拒绝对其他委托人执行其工作所需的 KMS 密钥的访问。不具有更改密钥策略或创建授权权限的密钥管理员可以控制对 KMS 密钥的访问，前提是他们有权管理标签。

  如有可能，请使用策略条件，例如 `aws:RequestTag/tag-key` 或 `aws:TagKeys`，以将委托人的标记权限限制为特定 KMS 密钥上的特定标签或标签模式。
- 查看您中当前拥有标记和取消标记权限 Amazon Web Services 账户 的委托人，并在必要时对其进行调整。例如，控制台密钥管理员的默认密钥策略包括对该 KMS 密钥的 `kms:TagResource` 和 `kms:UntagResource` 权限。IAM policy 可能允许对所有 KMS 密钥的标记和取消标记权限。例如，AWSKeyManagementServicePowerUser托管策略允许委托人标记、取消标记和列出所有 KMS 密钥上的标签。
- 在设置依赖于标签的策略之前，请查看您的 KMS 密钥上的标签 Amazon Web Services 账户。请确保您的策略仅适用于您要包含的标签。使用CloudTrail 日志和CloudWatch 警报提醒您注意可能影响您的 KMS 密钥访问权限的标签更改。
- 基于标签的策略条件使用模式匹配；它们不绑定到标签的特定实例。使用基于标签的条件键的策略会影响与模式匹配的所有新标签和现有标签。如果删除并重新创建与策略条件匹配的标签，则该条件将应用于新标签，就像对旧标签一样。

例如，请考虑以下 IAM policy。它允许委托人仅对您账户中位于亚太（新加坡）地区GenerateDataKeyWithoutPlaintext且有标签的 KMS 密钥调用和解密操作。`"Project"="Alpha"`您可以将此策略附加到示例 Alpha 项目中的角色。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "IAMPolicyWithResourceTag",
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKeyWithoutPlaintext",
        "kms:Decrypt"
      ],
      "Resource": "arn:aws:kms:ap-southeast-1:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Project": "Alpha"
        }
      }
    }
  ]
}
```

以下示例 IAM policy 允许委托人使用账户中的 KMS 密钥执行特定加密操作。但它禁止主体对具有 "Type"="Reserved" 标签或不包含 "Type" 标签的 KMS 密钥使用这些加密操作。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "IAMAllowCryptographicOperations",
      "Effect": "Allow",
      "Action": [
        "kms:Encrypt",
        "kms:GenerateDataKey*",
        "kms:Decrypt",
        "kms:ReEncrypt*"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*"
```

```
    },
    {
      "Sid": "IAMDenyOnTag",
      "Effect": "Deny",
      "Action": [
        "kms:Encrypt",
        "kms:GenerateDataKey*",
        "kms:Decrypt",
        "kms:ReEncrypt*"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Type": "Reserved"
        }
      }
    },
    {
      "Sid": "IAMDenyNoTag",
      "Effect": "Deny",
      "Action": [
        "kms:Encrypt",
        "kms:GenerateDataKey*",
        "kms:Decrypt",
        "kms:ReEncrypt*"
      ],
      "Resource": "arn:aws:kms:*:111122223333:key/*",
      "Condition": {
        "Null": {
          "aws:ResourceTag/Type": "true"
        }
      }
    }
  ]
}
```

密钥存储

密钥存储是用于存储和使用加密密钥的安全位置。Amazon KMS 中的默认密钥存储还支持生成和管理其存储的密钥的方法。默认情况下，对于您在 Amazon KMS 中创建的 Amazon KMS keys，密钥材料将在符合 FIPS 140-3 加密模块认证计划的硬件安全模块（HSM）中生成并受其保护。KMS 密钥的密钥材料绝不会让 HSM 处于未加密状态。

使用 Amazon KMS 来创建和管理加密密钥时，Amazon KMS 支持用多种类型的密钥存储来保护密钥材料。Amazon KMS 提供的所有密钥存储选项均按照 FIPS 140-3 安全 3 级进行持续验证，旨在防止任何人（包括 Amazon 操作员）在未经您许可的情况下访问或使用您的纯文本密钥。

Amazon KMS 标准密钥存储

默认情况下，KMS 密钥是使用标准 Amazon KMS HSM 创建的。从您的角度来看，这种 HSM 类型可以看作是一组多租户 HSM 实例集，以实现可扩展性最高、成本最低且最容易管理的密钥存储。如果要创建将在一个或多个 Amazon Web Services 服务内使用的 KMS 密钥，以便该服务可以代表您加密数据，则需要创建对称密钥。如果您在自己的应用程序设计中使用 KMS 密钥，则可以选择创建对称加密密钥、非对称密钥或 HMAC 密钥。

在标准密钥存储选项中，Amazon KMS 会创建密钥，然后使用服务内部管理的密钥对其进行加密。然后，您密钥的加密版本的多个副本将存储在系统中，系统旨在保障持久性。在标准密钥存储类型中生成和保护密钥材料可让您充分利用 Amazon KMS 的可扩展性、可用性和耐久性，同时将 Amazon 密钥存储的操作负担和成本降至最低。

带导入的密钥材料的 Amazon KMS 标准密钥存储

您可以选择将密钥材料导入 Amazon KMS，而不是要求 Amazon KMS 生成并存储给定密钥的唯一副本，这样便能生成自己的 256 位对称加密密钥、RSA 或椭圆曲线（ECC）密钥，或HMAC 散列消息认证码密钥，并将其应用于 KMS 密钥标识符（keyId）。这有时被称为自带密钥（BYOK）。必须使用 Amazon KMS 颁发的公有密钥、支持的加密包装算法和 Amazon KMS 提供的基于时间的导入令牌来保护从本地密钥管理系统导入的密钥材料。此过程将验证您的加密导入密钥在离开您的环境后，是否只能通过 Amazon KMS HSM 解密。

如果您对生成密钥的系统有特定要求，或者想要在 Amazon 外保存密钥副本作为备份，则导入的密钥材料可能会很有用。请注意，您需要对导入的密钥材料的整体可用性和持久性负责。尽管 Amazon KMS 有导入的密钥的副本，并且可以在您需要确保高度可用，但导入的密钥还提供了一个特殊的删除 API – DeleteImportedKeyMaterial。此 API 将立即删除 Amazon KMS 拥有的导入的密钥材料的所有副本，且 Amazon 无法选择恢复密钥。此外，您还可以为导入的密钥设置过期时间，过期后密钥将无法

使用。要使密钥在 Amazon KMS 中再次可用，则必须重新导入密钥材料并将其分配给相同的 keyId。这种针对导入密钥的删除操作与 Amazon KMS 代表您生成和存储的标准密钥不同。在标准情况下，密钥删除过程有一个强制性的等待期，在此期间，计划删除的密钥将首先被禁止使用。此操作允许您在可能需要该密钥才能访问数据的任何应用程序或 Amazon 服务的日志中查看拒绝访问错误。如果您看到此类访问请求，可以选择取消计划删除并重新启用该密钥。经过可配置的等待期（7 到 30 天）后，KMS 才会真正删除密钥材料、keyID 以及与密钥关联的所有元数据。有关可用性和持久性的更多信息，请参阅《Amazon KMS 开发人员指南》中的保护导入的密钥材料。

导入的密钥材料还有一些其他限制需要注意。由于 Amazon KMS 无法生成新的密钥材料，因此无法对导入密钥配置自动轮换。您需要使用新的 keyId 创建新 KMS 密钥，然后导入新的密钥材料才能实现有效的轮换。此外，使用导入的对称密钥在 Amazon KMS 中创建的加密文字无法使用 Amazon 外的本地密钥副本轻松解密。这是因为 Amazon KMS 使用的经过身份验证的加密格式会将额外的元数据附加到加密文字中，以便在解密操作期间确保加密文字是由之前加密操作中使用的预期 KMS 密钥创建的。大多数外部加密系统不了解如何通过解析这些元数据来获得原始加密文字，从而使用其对称密钥副本。在导入的非对称密钥（例如 RSA 或 ECC）下创建的加密文字可以有匹配的（公有或私有）密钥部分的情况下，在 Amazon KMS 外使用，因为 Amazon KMS 没有向加密文字添加额外的元数据。

Amazon KMS 自定义密钥存储

不过，如果您需要加强对 HSM 的控制，则可以创建自定义密钥存储。

自定义密钥存储是 Amazon KMS 中的密钥存储，由您拥有和管理的 Amazon KMS 外的密钥管理器提供支持。自定义密钥存储将 Amazon KMS 的方便、全面的密钥管理界面与拥有和控制密钥材料和加密操作的能力结合。在自定义密钥存储中使用 KMS 密钥时，加密操作实际上是通过您的密钥管理器使用您的加密密钥来执行。因此，您对加密密钥的可用性和持久性以及 HSM 的运行承担更多责任。

拥有您的 HSM 可能有助于满足某些监管要求，这些要求尚不允许多租户 Web 服务（例如标准 KMS 密钥存储）保留您的加密密钥。自定义密钥存储并不比使用 Amazon 托管 HSM 的 KMS 密钥存储更安全，但其管理影响不同，且成本更高。因此，您对加密密钥的可用性和持久性以及 HSM 的运行承担更多责任。无论是将标准密钥存储与 Amazon KMS HSM 一起使用，还是使用自定义密钥存储，该服务都旨在确保任何人（包括 Amazon 员工）都无法在未经您许可的情况下检索或使用您的明文密钥。Amazon KMS 支持两种类型的自定义密钥存储：Amazon CloudHSM 密钥存储和外部密钥存储。

不支持的 特征

Amazon KMS 在自定义密钥存储中不支持以下功能。

- 非对称 KMS 密钥
- HMAC KMS 密钥

- 具有导入密钥材料的 KMS 密钥
- 自动密钥轮换
- 多区域密钥

Amazon CloudHSM 密钥存储

您可以在 Amazon CloudHSM 密钥存储中创建 KMS 密钥，这种情况下，将在您拥有并管理的 Amazon CloudHSM 集群中生成、存储和使用根用户密钥。向 Amazon KMS 发出的使用密钥执行某些密码操作的请求，将转发到您的 Amazon CloudHSM 集群以执行操作。虽然 Amazon CloudHSM 集群由 Amazon 托管，但它是由您直接管理和操作的单租户解决方案。Amazon CloudHSM 集群中 KMS 密钥的大部分可用性和性能由您拥有。要查看 Amazon CloudHSM 自定义密钥库是否符合您的要求，请在 Amazon 安全博客上阅读 Are Amazon KMS custom key stores right for you?。

外部密钥存储

您可以将 Amazon KMS 配置为使用外部密钥存储（XKS），这种情况下，将在 Amazon Web Services 云 外的密钥管理系统中生成、存储和使用根用户密钥。向 Amazon KMS 提出的使用密钥进行某些加密操作的请求将转到您的外部托管系统以执行操作。具体而言，请求会被转发到您网络中的 XKS 代理，然后该代理会将请求转发到您使用的任何加密系统。XKS 代理是开源规范，任何人都可以集成。许多商业密钥管理供应商都支持 XKS 代理规范。由于外部密钥存储由您或第三方托管，因此您负责系统中密钥的所有可用性、持久性和性能。要查看外部密钥存储是否符合您的要求，请阅读 Amazon 新闻博客上的 Announcing Amazon KMS External Key Store (XKS)。

Amazon CloudHSM 密钥存储

Amazon CloudHSM 密钥存储是由 Amazon CloudHSM 集群支持的自定义密钥存储。当您在自定义密钥存储中创建 Amazon KMS key时，Amazon KMS 将在您拥有并管理的 Amazon CloudHSM 集群中为该 KMS 密钥生成并存储不可提取的密钥材料。在自定义密钥存储中使用 KMS 密钥时，会在集群中的 HSM 中执行加密操作。此功能在您的 Amazon Web Services 账户 中将 Amazon KMS 的便利性和广泛集成与 Amazon CloudHSM 集群的更多控制相结合。

Amazon KMS 提供了完整的控制台和 API 支持，以创建、使用和管理您的自定义密钥存储。在自定义密钥存储中使用 KMS 密钥的方式，与使用任何 KMS 密钥相同。例如，您可以使用 KMS 密钥生成数据密钥和加密数据。您还可以将自定义密钥存储中的 KMS 密钥与支持客户托管密钥的 Amazon 服务一起使用。

我是否需要自定义密钥存储？

对于大多数用户，默认 Amazon KMS 密钥存储（受 FIPS 140-3 验证的加密模块保护）满足其安全要求。无需添加额外的维护责任层或额外服务的依赖项。

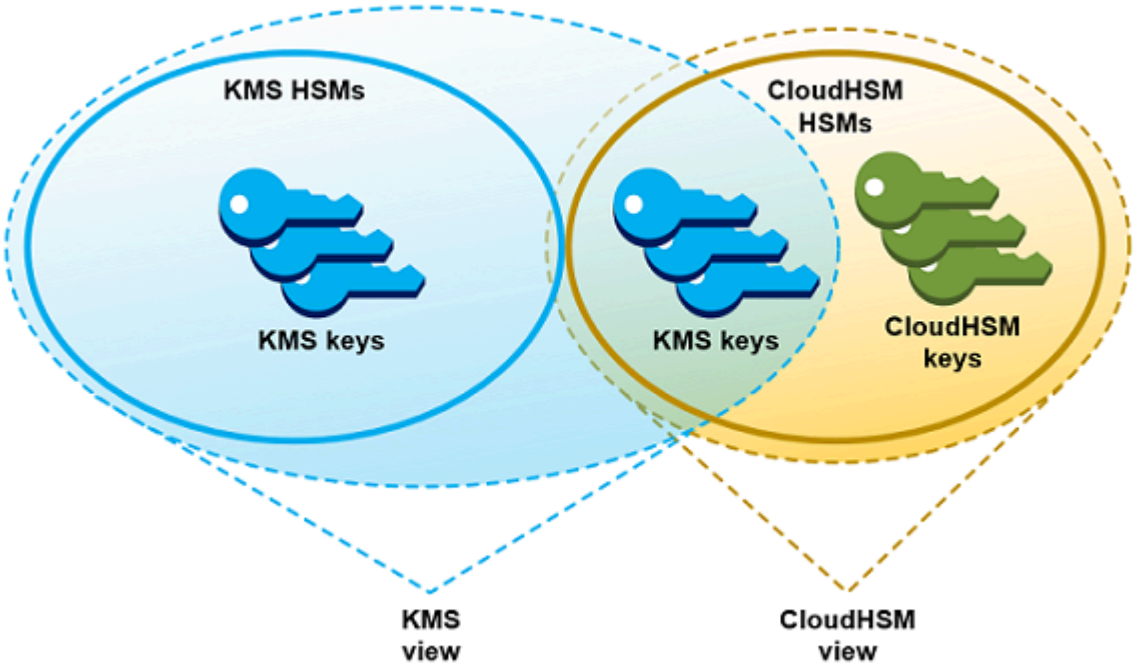
但是，如果您的组织具有以下任何要求，则可能要考虑创建自定义密钥存储：

- 在单租户 HSM 或您可以直接控制的 HSM 中，您有明确要求保护的密钥。
- 您需要能够立即从 Amazon KMS 中移除密钥材料。
- 您需要能够独立于 Amazon KMS 或 Amazon CloudTrail 审计密钥的所有使用情况。

自定义密钥存储如何工作？

每个自定义密钥存储均与您的 Amazon Web Services 账户 中的一个 Amazon CloudHSM 集群关联。在将自定义密钥存储连接到其集群时，Amazon KMS 将创建网络基础设施来支持连接。然后使用集群中的专用加密用户的凭证登录到集群中的密钥 Amazon CloudHSM 客户端。

在 Amazon KMS 中创建和管理自定义密钥存储，而在 Amazon CloudHSM 中创建和管理 HSM 集群。当您在 Amazon KMS 自定义密钥存储中创建 Amazon KMS keys 时，您可以在 Amazon KMS 中查看和管理 KMS 密钥。但是，您还可以在 Amazon CloudHSM 中查看和管理其密钥材料，就像您对集群中的其他密钥执行该操作一样。



您可以使用您的自定义密钥存储中的 Amazon KMS 生成的密钥材料创建对称加密 KMS 密钥。然后，使用用于 Amazon KMS 密钥存储中的 KMS 密钥的相同方法来查看和管理自定义密钥存储中的 KMS

密钥。您可以使用 IAM 和密钥策略控制访问，创建标签和别名，启用和禁用 KMS 密钥，以及计划密钥删除。您可以使用 KMS 密钥执行加密操作并将其与 Amazon 服务（已与 Amazon KMS 集成）一起使用。

此外，您可以完全控制 Amazon CloudHSM 集群，包括创建和删除 HSM 以及管理备份。您可以使用 Amazon CloudHSM 客户端和支持的软件库来查看、审计和管理 KMS 密钥的密钥材料。在自定义密钥存储断开后，Amazon KMS 无法访问它，并且用户无法使用自定义密钥存储中的 KMS 密钥执行加密操作。增加的控制层使自定义密钥存储成为需要它的组织的强大解决方案。

从何处开始？

要创建和管理 Amazon CloudHSM 密钥存储，请使用 Amazon KMS 和 Amazon CloudHSM 的功能。

1. 在 Amazon CloudHSM 中开始。创建活动 Amazon CloudHSM 集群或选择现有集群。集群必须在不同的可用区中具有至少两个活动 HSM。然后，在该集群中为 Amazon KMS 创建一个专用加密用户 (CU) 账户。
2. 在 Amazon KMS 中，创建一个自定义密钥存储（与选定的 Amazon CloudHSM 集群关联）。Amazon KMS 提供一个完整的管理界面，使您能够创建、查看、编辑和删除您的自定义密钥存储。
3. 在准备好使用您的自定义密钥存储后，将它连接到其关联的 Amazon CloudHSM 集群。Amazon KMS 创建其所需的网络基础设施来支持连接。之后，它将使用专用加密用户账户凭证登录集群，以便它可以在集群中生成和管理密钥材料。
4. 现在，您可以在自定义密钥存储中创建对称加密 KMS 密钥。在创建 KMS 密钥时指定自定义密钥存储。

如果您在任何时候遇到困难，都可以在对自定义密钥存储进行故障排除主题中查找帮助。如果您的问题未得到解答，请使用本指南的每页底部的反馈链接或在 Amazon Key Management Service 开发论坛上发布问题。

限额

Amazon KMS 允许每个 Amazon Web Services 账户 和区域中存在多达 10 个自定义密钥存储，包括 Amazon CloudHSM 密钥存储和外部密钥存储，无论这些密钥存储的连接状态如何。此外，在 Amazon CloudHSM 密钥存储中使用 KMS 密钥 存在 Amazon KMS 请求限额。

定价

有关自定义密钥存储中的 Amazon KMS 自定义密钥存储和客户托管密钥的成本的信息，请参阅 Amazon Key Management Service 定价。有关 Amazon CloudHSM 集群和 HSM 的成本的信息，请参阅 Amazon CloudHSM 定价。

区域

Amazon KMS 在所有支持 Amazon KMS 的 Amazon Web Services 区域 中均支持 Amazon CloudHSM 密钥存储，亚太地区（墨尔本）、中国（北京）、中国（宁夏）和欧洲（西班牙）除外。

不支持的 特征

Amazon KMS 在自定义密钥存储中不支持以下功能。

- 非对称 KMS 密钥
- HMAC KMS 密钥
- 具有导入密钥材料的 KMS 密钥
- 自动密钥轮换
- 多区域密钥

Amazon CloudHSM 密钥存储概念

本主题介绍了 Amazon CloudHSM 密钥存储中使用的一些术语和概念。

Amazon CloudHSM 密钥存储

Amazon CloudHSM 密钥存储是与您拥有和管理的 Amazon CloudHSM 集群关联的自定义密钥存储。Amazon CloudHSM 集群由 FIPS 140-2 或 FIPS 140-3 3 级认证的硬件安全模块（HSM）提供支持。

当您在 Amazon CloudHSM 密钥存储中创建 KMS 密钥时，Amazon KMS 将在关联的 Amazon CloudHSM 集群中生成 256 位、持久且不可导出的高级加密标准（AES）对称密钥。此密钥材料绝不会让您的 HSM 处于未加密状态。在 Amazon CloudHSM 密钥存储中使用 KMS 密钥时，会在集群中的 HSM 中执行加密操作。

Amazon CloudHSM 密钥存储将 Amazon KMS 的方便、全面的密钥管理界面与 Amazon Web Services 账户 中的 Amazon CloudHSM 集群提供的额外控制相结合。此集成功能可让您在 Amazon KMS 中创建、管理和使用 KMS 密钥，同时保持对存储 KMS 密钥的密钥材料的 HSM 的完全控制，包括管理集群、HSM 和备份。您可以使用 Amazon KMS 控制台和 API 来管理 Amazon CloudHSM 密钥

存储及其 KMS 密钥。您还可以使用 Amazon CloudHSM 控制台、API、客户端软件和关联的软件库来管理关联的集群。

您可以查看和管理您的 Amazon CloudHSM 密钥存储、编辑其属性，并将其与关联的 Amazon CloudHSM 集群连接和断开连接。如果您需要删除 Amazon CloudHSM 密钥存储，则必须先删除 Amazon CloudHSM 密钥存储中的 KMS 密钥，方法为计划其删除并等到宽限期结束。删除 Amazon CloudHSM 密钥存储会从 Amazon KMS 中移除资源，但不会影响您的 Amazon CloudHSM 集群。

Amazon CloudHSM 集群

每个 Amazon CloudHSM 密钥存储均与一个 Amazon CloudHSM 集群关联。当您在 Amazon CloudHSM 密钥存储中创建 Amazon KMS key 时，Amazon KMS 将在关联的集群中创建其密钥材料。当您在 Amazon CloudHSM 密钥存储中使用 KMS 密钥时，将在关联的集群中执行加密操作。

每个 Amazon CloudHSM 集群只能与一个 Amazon CloudHSM 密钥存储关联。您选择的集群无法与另一个 Amazon CloudHSM 密钥存储关联或无法和与其他 Amazon CloudHSM 密钥存储关联的集群共享备份历史记录。集群必须初始化且处于活动状态，并且必须与 Amazon CloudHSM 密钥存储位于相同的 Amazon Web Services 账户 和区域中。您可以创建新集群或使用现有集群。Amazon KMS 不需要集群的专用权。要在 Amazon CloudHSM 密钥存储中创建 KMS 密钥，其关联的集群必须包含至少两个活动 HSM。所有其他操作都只需要一个 HSM。

您在创建 Amazon CloudHSM 密钥存储时指定 Amazon CloudHSM 集群，并且无法更改它。但是，您可以替换与原始集群共享备份历史记录的任何集群。这样，您就可以删除该集群（如有必要），然后将它替换为从其备份之一创建的集群。您可以完全控制关联的 Amazon CloudHSM 集群，从而能管理用户和密钥、创建和删除 HSM 以及使用和管理备份。

当您准备好使用 Amazon CloudHSM 密钥存储时，您可以将其连接到关联的 Amazon CloudHSM 集群。您可以随时连接和断开自定义密钥存储。连接自定义密钥存储后，您可以创建和使用其 KMS 密钥。断开密钥存储后，您可以查看和管理 Amazon CloudHSM 密钥存储及其 KMS 密钥。但您无法创建新 KMS 密钥或在 Amazon CloudHSM 密钥存储中使用 KMS 密钥以进行加密操作。

`kmsuser` 加密用户

为了代表创建和管理关联的 Amazon CloudHSM 集群中的密钥材料，Amazon KMS 在名为 `kmsuser` 的集群中使用专用的 Amazon CloudHSM 加密用户（CU）。`kmsuser` CU 是一个标准 CU 账户，它将自动同步到集群中的所有 HSM 并保存在集群备份中。

在创建 Amazon CloudHSM 密钥存储之前，您在 CloudHSM CLI 中使用 user create 命令在您的 Amazon CloudHSM 集群中创建 `kmsuser` CU 账户。然后，当您创建 Amazon CloudHSM 密钥存

储时，您向 Amazon KMS 提供 kmsuser 账户密码。当您连接自定义密钥存储时，Amazon KMS 将以 kmsuser CU 身份登录到集群并轮换其密码。Amazon KMS 会在安全存储之前对您的 kmsuser 密码进行加密。轮换密码时，新密码也会以相同的方式加密和存储。

只要 Amazon CloudHSM 密钥存储处于连接状态，Amazon KMS 将保持以 kmsuser 身份登录。您不应将此 CU 账户用于其他目的。但是，您保留对 kmsuser CU 账户的最终控制权。您可以随时查找 kmsuser 拥有的密钥。必要时，您可以将自定义密钥存储断开连接，更改 kmsuser 密码，以 kmsuser 身份登录到集群，以及查看和管理 kmsuser 拥有的密钥。

有关创建 kmsuser CU 账户的说明，请参阅创建 kmsuser 加密用户。

在 Amazon CloudHSM 密钥存储中的 KMS 密钥

您可以使用 Amazon KMS 或 Amazon KMS API 在 Amazon CloudHSM 密钥存储中创建 Amazon KMS keys。您将使用对任何 KMS 密钥使用的同一方法。唯一的区别在于您必须标识 Amazon CloudHSM 密钥存储并指定密钥材料的源是 Amazon CloudHSM 集群。

当您在 Amazon CloudHSM 密钥存储中创建一个 KMS 密钥时，Amazon KMS 将在 Amazon KMS 中创建 KMS 密钥，并在其关联集群中生成 256 位、持久且不可导出的高级加密标准（AES）对称密钥材料。在加密操作中使用 Amazon KMS 密钥时，该操作将在 Amazon CloudHSM 集群中使用基于集群的 AES 密钥执行。尽管 Amazon CloudHSM 支持不同类型的对称和非对称密钥，但 Amazon CloudHSM 密钥存储仅支持 AES 对称加密密钥。

您可以在 Amazon KMS 控制台中查看 Amazon CloudHSM 密钥存储中的 KMS 密钥，并使用控制台选项显示自定义密钥存储 ID。您还可以使用 DescribeKey 操作来查找 Amazon CloudHSM 密钥存储 ID 和 Amazon CloudHSM 集群 ID。

Amazon CloudHSM 密钥存储中的 KMS 密钥与 Amazon KMS 中的任何 KMS 密钥的工作方式相同。授权用户需要相同的权限来使用和管理 KMS 密钥。您可使用相同的控制台过程和 API 操作来查看和管理 Amazon CloudHSM 密钥存储中的 KMS 密钥。这包括启用和禁用 KMS 密钥、创建和使用标签和别名以及设置和更改 IAM 和密钥策略。您可以使用 Amazon CloudHSM 密钥存储中的 KMS 密钥执行密码操作，并将其与支持使用客户自主管理型密钥的集成 Amazon 服务结合使用。但是，不能启用自动密钥轮换或者将密钥材料导入到 Amazon CloudHSM 密钥存储中的 KMS 密钥。

您还可以将相同的过程用于 Amazon CloudHSM 密钥存储中的 KMS 密钥的计划删除。在等待期限结束后，Amazon KMS 将从 KMS 中删除 KMS 密钥。然后，它将尽可能从关联的 Amazon CloudHSM 集群中删除 KMS 密钥的密钥材料。但是，您可能需要从集群及其备份中手动删除孤立密钥材料。

控制对 Amazon CloudHSM 密钥存储的访问

您可以使用 IAM policy 控制对 Amazon CloudHSM 密钥存储和 Amazon CloudHSM 集群的访问。您可以使用密钥策略、IAM policy 和授权来控制对 Amazon CloudHSM 密钥存储中的 Amazon KMS keys 的访问。我们建议您仅向用户、组和角色提供他们可能执行的任务所需的权限。

为了支持您的 Amazon CloudHSM 密钥存储，Amazon KMS 需要获取有关您的 Amazon CloudHSM 集群的信息的权限。它还需要创建将您的 Amazon CloudHSM 密钥存储连接到其 Amazon CloudHSM 集群的网络基础设施的权限。要获得这些权限，Amazon KMS 会在您的 Amazon Web Services 账户中创建 AWSServiceRoleForKeyManagementServiceCustomKeyStores 服务相关角色。有关更多信息，请参阅 授权 Amazon KMS 管理 Amazon CloudHSM 和 Amazon 资源 EC2 。

在设计您的 Amazon CloudHSM 密钥存储时，请确保使用和管理它的主体仅具有其所需的权限。以下列表描述了 Amazon CloudHSM 密钥存储管理员和用户所需的最低权限。

- 创建和管理 Amazon CloudHSM 密钥存储的主体需要以下权限才能使用 Amazon CloudHSM 密钥存储 API 操作。
  - `cloudhsm:DescribeClusters`
  - `kms:CreateCustomKeyStore`
  - `kms:ConnectCustomKeyStore`
  - `kms:DeleteCustomKeyStore`
  - `kms:DescribeCustomKeyStores`
  - `kms:DisconnectCustomKeyStore`
  - `kms:UpdateCustomKeyStore`
  - `iam:CreateServiceLinkedRole`
- 创建和管理与您的 Amazon CloudHSM 密钥存储关联的 Amazon CloudHSM 集群的主体需要创建和初始化 Amazon CloudHSM 集群的权限。这包括创建或使用 Amazon 虚拟私有云（VPC）、创建子网和创建 Amazon EC2 实例所需的权限。他们可能还需要创建和删除 HSM 以及管理备份。有关所需权限的列表，请参阅《Amazon CloudHSM User Guide》中的 Identity and access management for Amazon CloudHSM。
- 在您的 Amazon CloudHSM 密钥存储中创建和管理 Amazon KMS keys 的主体需要具有与在 Amazon KMS 中创建和管理任何 KMS 密钥的人员相同的权限。Amazon CloudHSM 密钥存储中的 KMS 密钥的默认密钥策略与 Amazon KMS 中的 KMS 密钥的默认密钥策略相同。基于属性的访问权限控制（ABAC）使用标签和别名来控制对 KMS 密钥的访问，对 Amazon CloudHSM 密钥存储中的 KMS 密钥也有效。

- 使用 Amazon CloudHSM 密钥存储中的 KMS 密钥进行加密操作的主体需要使用 KMS 密钥执行加密操作的权限，如 kms:Decrypt。您可以在密钥策略或 IAM policy 中提供这些权限。但是，他们无需任何额外权限即可在 Amazon CloudHSM 密钥存储中使用 KMS 密钥。

创建 Amazon CloudHSM 密钥存储

您可以在账户中创建一个或多个 Amazon CloudHSM 密钥存储。每个 Amazon CloudHSM 密钥存储均与相同 Amazon Web Services 账户 和区域中的一个 Amazon CloudHSM 集群关联。在创建 Amazon CloudHSM 密钥存储之前，您需要汇编先决条件。然后，在使用 Amazon CloudHSM 密钥存储之前，您必须连接它和它的 Amazon CloudHSM 集群。

> **备注**
>
> KMS 无法通过 IPv6 与 Amazon CloudHSM 密钥存储通信。
> 如果您尝试创建一个属性值与断开连接的现有 Amazon CloudHSM 密钥存储相同的 Amazon CloudHSM 密钥存储，Amazon KMS 不会创建新的 Amazon CloudHSM 密钥存储，也不会引发异常或显示错误。相反，Amazon KMS 将副本识别为重试的可能结果，并返回现有 Amazon CloudHSM 密钥存储的 ID。
> 您无需立即连接 Amazon CloudHSM 密钥存储。您可以将它保持断开状态，直到您准备好使用它为止。但是，要验证它是否已正确配置，您可能需要连接它，查看其连接状态，然后断开它。

主题

汇编先决条件

每个 Amazon CloudHSM 密钥存储均由一个 Amazon CloudHSM 集群提供支持。要创建 Amazon CloudHSM 密钥存储，您必须指定一个尚未与其他密钥存储关联的活动 Amazon CloudHSM 集群。您还需要在集群的 HSM 中创建一个专用的加密用户 (CU)，Amazon KMS 可使用该用户代表您创建和管理密钥。

在创建 Amazon CloudHSM 密钥存储之前，请执行以下操作：

选择一个 Amazon CloudHSM 集群

每个 Amazon CloudHSM 密钥存储正好与一个 Amazon CloudHSM 集群关联。在 Amazon CloudHSM 密钥存储中创建 Amazon KMS keys 时，Amazon KMS 会在 Amazon KMS 中创建 KMS 密钥元数据，例如 ID 和 Amazon 资源名称（ARN）。然后，它会在关联集群的 HSM 中创建密钥材料。您可以创建新的 Amazon CloudHSM 集群或使用现有的集群。Amazon KMS 不需要对集群的独占访问权限。

所选 Amazon CloudHSM 集群将与 Amazon CloudHSM 密钥存储永久关联。在创建 Amazon CloudHSM 密钥存储后，您可以为关联集群更改集群 ID，但您指定的集群必须与原始集群共享备份历史记录。要使用不相关的集群，您需要创建新的 Amazon CloudHSM 密钥存储。

您选择的 Amazon CloudHSM 集群必须具有以下特征：

- 集群必须处于活动状态。

  您必须创建集群，将其初始化，安装适用于您的平台的 Amazon CloudHSM 客户端软件，然后激活该集群。有关详细说明，请参阅《Amazon CloudHSM User Guide》中的 Getting started with Amazon CloudHSM。

- 未启用双向 TLS（mTLS）。

  KMS 不支持将 mTLS 用于集群。务必不要启用此设置。

- 集群必须与 Amazon CloudHSM 密钥存储位于相同的账户和区域中。您无法将一个区域中的 Amazon CloudHSM 密钥存储与另一个区域中的集群相关联。要在多个区域中创建密钥基础设施，必须在每个区域中创建 Amazon CloudHSM 密钥存储和集群。

- 集群不能与同一账户和区域中的其他自定义密钥存储关联。此账户和区域中的每个 Amazon CloudHSM 密钥存储都必须与不同的 Amazon CloudHSM 集群关联。您无法指定已与自定义密钥存储关联的集群，也无法指定与关联集群共享备份历史记录的集群。共享备份历史记录的集群具有相同的集群证书。要查看集群的集群证书，请使用 Amazon CloudHSM 控制台或 DescribeClusters 操作。

  如果您将 Amazon CloudHSM 集群备份到不同的区域，则它将被视为一个不同的集群，并且您可以将备份关联到其区域中的自定义密钥存储。但是，两个自定义密钥存储中的 KMS 密钥不可互操作，即使它们具有相同的备用密钥。Amazon KMS 会将元数据绑定到加密文字，以确保只能由加密所用 KMS 密钥进行解密。

- 必须在区域中的至少两个可用区中为集群配置私有子网。由于 Amazon CloudHSM 并非在所有可用区中都受支持，因此，我们建议您在该区域的所有可用区中创建私有子网。您无法为现有集群重新配置子网，但可以从备份创建集群（在集群配置中具有不同的子网）。

> **Important**
>
> 创建 Amazon CloudHSM 密钥存储后，不要删除为其 Amazon CloudHSM 集群配置的任何私有子网。如果 Amazon KMS 未找到集群配置中的所有子网，则尝试连接到自定义密钥存储会失败，并显示 SUBNET_NOT_FOUND 连接错误状态。有关更多信息，请参阅 如何修复连接故障。

- 集群的安全组 (cloudhsm-cluster-*<cluster-id>*-sg) 必须包含允许使用 IPv4 的 TCP 流量通过端口 2223-2225 的入站规则和出站规则。入站规则中的 Source (源) 和出站规则中的 Destination (目标) 必须匹配安全组 ID。在创建集群时，默认情况下会设置这些规则。请勿删除或更改它们。
- 集群必须在不同的可用区中包含至少两个活动 HSM。要验证 HSM 的数量，请使用 Amazon CloudHSM 控制台或 DescribeClusters 操作。如有必要，您可以添加 HSM。

查找信任锚点证书

在创建自定义密钥存储时，您必须将 Amazon CloudHSM 集群的信任锚点证书上传到 Amazon KMS。Amazon KMS 需要信任锚点证书才能将 Amazon CloudHSM 密钥存储连接到 Amazon CloudHSM 集群。

每个活动 Amazon CloudHSM 集群均有一个信任锚点证书。在初始化集群时，您将生成此证书，将它保存在 customerCA.crt 文件中，并将它复制到已连接到集群的主机。

为 **kmsuser** 创建 Amazon KMS 加密用户

为了管理您的 Amazon CloudHSM 密钥存储，Amazon KMS 会登录到选定集群中的 kmsuser 加密用户（CU）账户。在创建您的 Amazon CloudHSM 密钥存储之前，您必须创建 kmsuser CU。然后，在创建 Amazon CloudHSM 密钥存储时，您向 Amazon KMS 提供 kmsuser 密码。在将 Amazon CloudHSM 密钥存储连接到其关联的 Amazon CloudHSM 集群时，Amazon KMS 将以 kmsuser 身份登录并轮换 kmsuser 密码。

> **Important**
>
> 在创建 2FA CU 时，请勿指定 kmsuser 选项。如果这样做，Amazon KMS 将无法登录，并且您的 Amazon CloudHSM 密钥存储将无法连接到此 Amazon CloudHSM 集群。一旦指定 2FA，便无法撤消它。相反，您必须删除 CU 并重新创建它。

> **备注**
>
> 以下过程使用 Amazon CloudHSM 客户端 SDK 5 命令行工具 CloudHSM CLI。CloudHSM CLI 将 `key-handle` 替换为 `key-reference`。
> 2025 年 1 月 1 日，Amazon CloudHSM 将终止对客户端 SDK 3 命令行工具、CloudHSM 管理实用程序（CMU）和密钥管理实用程序（KMU）的支持。有关客户端 SDK 3 命令行工具和客户端 SDK 5 命令行工具之间区别的更多信息，请参阅《Amazon CloudHSM 用户指南》中的从客户端 SDK 3 CMU 和 KMU 迁移到客户端 SDK 5 CloudHSM CLI。

1. 按照《Amazon CloudHSM 用户指南》的开始使用 CloudHSM 命令行界面（CLI）主题中所述的入门步骤进行操作。
2. 使用 user create 命令创建名为 `kmsuser` 的 CU。

   密码必须由 7 到 32 个字母数字字符组成。它区分大小写，并且不能包含任何特殊字符。

   以下示例命令创建了一个 `kmsuser` CU。

```
aws-cloudhsm > user create --username kmsuser --role crypto-user
Enter password:
Confirm password:
{
 "error_code": 0,
 "data": {
   "username": "kmsuser",
   "role": "crypto-user"
 }
}
```

创建新的 Amazon CloudHSM 密钥存储

汇编先决条件后，您可以在 Amazon KMS 控制台中或使用 CreateCustomKeyStore 操作创建新的 Amazon CloudHSM 密钥存储。

使用 Amazon KMS 控制台

在 Amazon Web Services 管理控制台 中创建 Amazon CloudHSM 密钥存储时，您可以添加并创建先决条件作为工作流的一部分。但是，如果您事先已汇编这些先决条件，则此过程会更快。

1. 登录到 Amazon Web Services 管理控制台，然后通过以下网址打开 Amazon Key Management Service（Amazon KMS）控制台：https://console.aws.amazon.com/kms。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择自定义密钥存储、Amazon CloudHSM 密钥存储。
4. 选择创建密钥存储。
5. 为自定义密钥存储输入友好名称。该名称在您账户的所有自定义密钥存储中必须具备唯一性。

   > **Important**
   >
   > 不要在此字段中包含机密或敏感信息。此字段可能会以纯文本形式显示在 CloudTrail 日志和其他输出中。

6. 为 Amazon CloudHSM 密钥存储选择一个 Amazon CloudHSM 集群。或者，要创建新的 Amazon CloudHSM 集群，请选择 Create an Amazon CloudHSM cluster（创建 Amazon CloudHSM 集群）链接。

   该菜单显示您账户和区域中尚未与 Amazon CloudHSM 密钥存储关联的 Amazon CloudHSM 集群。该集群必须满足要求（与关联自定义密钥存储相关）。
7. 选择 Choose file（选择文件），然后为选定的 Amazon CloudHSM 集群上传信任锚点证书。这是您在初始化集群时创建的 `customerCA.crt` 文件。
8. 输入您在选定集群中创建的 `kmsuser` 加密用户 (CU) 的密码。
9. 选择创建。

当该过程成功时，新的 Amazon CloudHSM 密钥存储将显示在账户和区域的 Amazon CloudHSM 密钥存储列表中。如果该过程失败，则会显示一条错误消息，描述问题并提供有关如何解决该问题的帮助。如果您需要更多帮助，请参阅对自定义密钥存储进行故障排除。

如果您尝试创建一个属性值与断开连接的现有 Amazon CloudHSM 密钥存储相同的 Amazon CloudHSM 密钥存储，Amazon KMS 不会创建新的 Amazon CloudHSM 密钥存储，也不会引发异常或显示错误。相反，Amazon KMS 将副本识别为重试的可能结果，并返回现有 Amazon CloudHSM 密钥存储的 ID。

下一步：不会自动连接新的 Amazon CloudHSM 密钥存储。您必须先连接自定义密钥存储与其关联的 Amazon CloudHSM 集群，然后才能在 Amazon CloudHSM 密钥存储中创建 Amazon KMS keys。

使用 Amazon KMS API

您可以使用 CreateCustomKeyStore 操作创建一个新的 Amazon CloudHSM 密钥存储，该密钥存储与账户和区域中的 Amazon CloudHSM 集群关联。这些示例使用 Amazon Command Line Interface (Amazon CLI)，但您可以使用任何受支持的编程语言。

`CreateCustomKeyStore` 操作需要以下参数值。

- CustomKeyStoreName – 自定义密钥存储的友好名称，该名称在账户中是唯一的。

> **Important**
>
> 不要在此字段中包含机密或敏感信息。此字段可能会以纯文本形式显示在 CloudTrail 日志和其他输出中。

- CloudHsmClusterId – 满足要求（与 Amazon CloudHSM 密钥存储相关）的 Amazon CloudHSM 集群的集群 ID。
- KeyStorePassword – 指定集群中的 `kmsuser` CU 账户的密码。
- TrustAnchorCertificate – 您初始化集群时创建的 `customerCA.crt` 文件的内容。

以下示例使用虚构的集群 ID。在运行命令之前，请将其替换为有效的集群 ID。

```
$ aws kms create-custom-key-store
        --custom-key-store-name ExampleCloudHSMKeyStore \
        --cloud-hsm-cluster-id cluster-1a23b4cdefg \
        --key-store-password kmsPswd \
        --trust-anchor-certificate <certificate-goes-here>
```

如果您使用的是 Amazon CLI，则可指定信任锚点证书文件而不是其内容。在下面的示例中，`customerCA.crt` 文件位于根目录中。

```
$ aws kms create-custom-key-store
        --custom-key-store-name ExampleCloudHSMKeyStore \
        --cloud-hsm-cluster-id cluster-1a23b4cdefg \
        --key-store-password kmsPswd \
        --trust-anchor-certificate file://customerCA.crt
```

当此操作成功时，`CreateCustomKeyStore` 将返回自定义密钥存储 ID，如以下示例响应中所示。

```
{
    "CustomKeyStoreId": cks-1234567890abcdef0
}
```

如果操作失败，请更正异常指示的错误，然后重试。有关其他帮助，请参阅对自定义密钥存储进行故障排除。

如果您尝试创建一个属性值与断开连接的现有 Amazon CloudHSM 密钥存储相同的 Amazon CloudHSM 密钥存储，Amazon KMS 不会创建新的 Amazon CloudHSM 密钥存储，也不会引发异常或显示错误。相反，Amazon KMS 将副本识别为重试的可能结果，并返回现有 Amazon CloudHSM 密钥存储的 ID。

下一步：要使用 Amazon CloudHSM 密钥存储，可将它连接到其 Amazon CloudHSM 集群。

查看 Amazon CloudHSM 密钥存储

您可以使用 Amazon KMS 控制台或使用 DescribeCustomKeyStores 操作查看每个账户和区域中的 Amazon CloudHSM 密钥存储。

使用 Amazon KMS 控制台

在 Amazon Web Services 管理控制台 中查看 Amazon CloudHSM 密钥存储时，可查看以下内容：

- 自定义密钥存储名称和 ID
- 关联的 Amazon CloudHSM 集群的 ID
- 集群中的 HSM 的数量
- 当前连接状态

连接状态 [Status（状态）] 值 Disconnected（已断开连接）表示自定义密钥存储是新的且从未连接过，或者已有意断开与其 Amazon CloudHSM 集群的连接。但是，如果您尝试使用已连接的自定义密钥存储中的 KMS 密钥失败，则可能表示自定义密钥存储或其 Amazon CloudHSM 集群存在问题。有关帮助信息，请参阅 如何修复失败的 KMS 密钥。

要查看给定账户和区域中的 Amazon CloudHSM 密钥存储，请使用以下过程。

1. 登录到 Amazon Web Services 管理控制台，然后通过以下网址打开 Amazon Key Management Service（Amazon KMS）控制台：https://console.aws.amazon.com/kms。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。

3. 在导航窗格中，选择自定义密钥存储、Amazon CloudHSM 密钥存储。

要自定义显示，请单击显示在 Create key store (创建密钥存储) 按钮下方的齿轮图标。

使用 Amazon KMS API

要查看您的 Amazon CloudHSM 密钥存储，请使用 DescribeCustomKeyStores 操作。默认情况下，此操作将返回账户和区域中的所有自定义密钥存储。不过，您可以使用 CustomKeyStoreId 或 CustomKeyStoreName 参数（但不能同时使用两者）将输出限制到特定的自定义密钥存储。对于 Amazon CloudHSM 密钥存储，输出内容包含自定义密钥存储 ID 和名称、自定义密钥存储类型、关联 Amazon CloudHSM 集群的 ID 以及连接状态。如果连接状态指示错误，则输出还包含描述错误原因的错误代码。

本部分中的示例使用 Amazon Command Line Interface (Amazon CLI)，但您可以使用任何受支持的编程语言。

例如，以下命令返回账户和区域中的所有自定义密钥存储。您可以使用 Limit 和 Marker 参数来浏览输出中的自定义密钥存储。

```
$ aws kms describe-custom-key-stores
```

以下示例命令使用 CustomKeyStoreName 参数以仅获取具有 ExampleCloudHSMKeyStore 友好名称的自定义密钥存储。您可以在每个命令中使用 CustomKeyStoreName 或 CustomKeyStoreId 参数（但不能同时使用二者）。

以下示例输出表示已连接到其 Amazon CloudHSM 集群的 Amazon CloudHSM 密钥存储。

> **Note**
>
> 在 DescribeCustomKeyStores 响应中添加了 CustomKeyStoreType 字段，以区分 Amazon CloudHSM 密钥存储和外部密钥存储。

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleCloudHSMKeyStore
{
   "CustomKeyStores": [
      {
         "CloudHsmClusterId": "cluster-1a23b4cdefg",
         "ConnectionState": "CONNECTED",
```

```
            "CreationDate": "1.499288695918E9",
            "CustomKeyStoreId": "cks-1234567890abcdef0",
            "CustomKeyStoreName": "ExampleCloudHSMKeyStore",
            "CustomKeyStoreType": "AWS_CLOUDHSM",
            "TrustAnchorCertificate": "<certificate appears here>"
        }
    ]
}
```

ConnectionState 为 Disconnected 表示自定义密钥存储从未连接过，或者它已有意从其 Amazon CloudHSM 集群断开连接。但是，如果尝试使用已连接的 Amazon CloudHSM 密钥存储中的 KMS 密钥失败，则可能表示 Amazon CloudHSM 密钥存储或其 Amazon CloudHSM 集群存在问题。有关帮助信息，请参阅 如何修复失败的 KMS 密钥。

如果自定义密钥存储的 ConnectionState 为 FAILED，则 DescribeCustomKeyStores 响应包含一个说明错误原因的 ConnectionErrorCode 元素。

例如，在以下输出中，INVALID_CREDENTIALS 值指示自定义密钥存储连接因 kmsuser 密码无效而导致失败。有关此连接失败及其他连接错误失败的帮助，请参阅对自定义密钥存储进行故障排除。

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
{
    "CustomKeyStores": [
        {
            "CloudHsmClusterId": "cluster-1a23b4cdefg",
            "ConnectionErrorCode": "INVALID_CREDENTIALS",
            "ConnectionState": "FAILED",
            "CustomKeyStoreId": "cks-1234567890abcdef0",
            "CustomKeyStoreName": "ExampleCloudHSMKeyStore",
            "CustomKeyStoreType": "AWS_CLOUDHSM",
            "CreationDate": "1.499288695918E9",
            "TrustAnchorCertificate": "<certificate appears here>"
        }
    ]
}
```

了解更多：

- 查看外部密钥存储
- 识别密钥库中的 KMS Amazon CloudHSM 密钥
- 使用 Amazon CloudTrail 记录 Amazon KMS API 调用

编辑 Amazon CloudHSM 密钥存储设置

您可以更改现有 Amazon CloudHSM 密钥存储的设置。必须断开自定义密钥存储与其 Amazon CloudHSM 集群的连接。

要编辑 Amazon CloudHSM 密钥存储设置，请执行以下操作：

1. 断开自定义密钥存储与其 Amazon CloudHSM 集群的连接。

   自定义密钥存储断开连接后，您将无法在该自定义密钥存储中创建 Amazon KMS keys（KMS 密钥），也无法使用其中包含的 KMS 密钥来执行密码操作。

2. 编辑一个或多个 Amazon CloudHSM 密钥存储设置。

   您可以在自定义密钥存储中编辑以下设置：

   自定义密钥存储的友好名称。

   输入新的友好名称。新名称在您 Amazon Web Services 账户 的所有自定义密钥存储中必须具备唯一性。

   > **Important**
   >
   > 不要在此字段中包含机密或敏感信息。此字段可能会以纯文本形式显示在 CloudTrail 日志和其他输出中。

   关联的 Amazon CloudHSM 集群的集群 ID。

   编辑此值以使用相关 Amazon CloudHSM 集群替换原始集群。如果自定义密钥存储的 Amazon CloudHSM 集群损坏或被删除，则可使用此功能来修复自定义密钥存储。

   指定一个 Amazon CloudHSM 集群，该集群与原始集群共享备份历史记录并满足要求以与自定义密钥存储关联，包括不同可用区中的两个活动 HSM。共享备份历史记录的集群具有相同的集群证书。要查看集群的集群证书，请使用 DescribeClusters 操作。您无法使用编辑功能来将自定义密钥存储与不相关的 Amazon CloudHSM 集群相关联。

   `kmsuser` 加密用户 (CU) 的当前密码。

   向 Amazon KMS 告知 `kmsuser` 集群中 Amazon CloudHSM CU 的当前密码。此操作不会更改 `kmsuser` 集群中 Amazon CloudHSM CU 的密码。

如果更改 kmsuser 集群中 Amazon CloudHSM CU 的密码，请使用此功能来向 Amazon KMS 告知新的 kmsuser 密码。否则，Amazon KMS 将无法登录集群并且所有将自定义密钥存储连接到集群的尝试都将失败。

3. 重新连接自定义密钥存储至其 Amazon CloudHSM 集群。

编辑您的密钥存储设置

您可以在 Amazon KMS 控制台中或使用 UpdateCustomKeyStore 操作来编辑 Amazon CloudHSM 密钥存储设置。

使用 Amazon KMS 控制台

在编辑 Amazon CloudHSM 密钥存储时，您可以更改任何可配置的值。

1. 登录到 Amazon Web Services 管理控制台，然后通过以下网址打开 Amazon Key Management Service（Amazon KMS）控制台：https://console.aws.amazon.com/kms。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择自定义密钥存储、Amazon CloudHSM 密钥存储。
4. 选择要编辑的 Amazon CloudHSM 密钥存储的行。

   如果连接状态列中的值不是已断开连接，则必须先断开自定义密钥存储，然后才能对其进行编辑。[从 Key store actions（密钥存储操作）菜单中选择 Disconnect（断开连接）。]

   当某个 Amazon CloudHSM 密钥存储断开时，您可以管理 Amazon CloudHSM 密钥存储及其 KMS 密钥，但无法在 Amazon CloudHSM 密钥存储中创建或使用 KMS 密钥。
5. 从 Key store actions（密钥存储操作）菜单中选择 Edit（编辑）。
6. 执行以下一项或多项操作。

   - 为自定义密钥存储键入新的友好名称。
   - 键入相关 Amazon CloudHSM 集群的集群 ID。
   - 键入关联的 kmsuser 集群中 Amazon CloudHSM 加密用户的当前密码。
7. 选择保存。

   在此过程成功后，将显示一条消息，描述您编辑的设置。如果此过程失败，则会显示一条错误消息，描述问题并提供有关如何解决问题的帮助。如果您需要更多帮助，请参阅对自定义密钥存储进行故障排除。

8. 重新连接自定义密钥存储。

   要使用 Amazon CloudHSM 密钥存储，您必须在编辑后重新连接它。您可以让 Amazon CloudHSM 密钥存储保留断开状态。不过，在其断开连接后，您无法在 Amazon CloudHSM 密钥存储中创建 KMS 密钥或在加密操作中使用 Amazon CloudHSM 密钥存储中的 KMS 密钥。

使用 Amazon KMS API

要更改 Amazon CloudHSM 密钥存储的属性，请使用 UpdateCustomKeyStore 操作。您可以在同一命令中更改自定义密钥存储的多个属性。如果此操作成功，则 Amazon KMS 返回 HTTP 200 响应和无属性的 JSON 对象。要验证变更是否生效，请使用 DescribeCustomKeyStores 操作。

本部分中的示例使用 Amazon Command Line Interface (Amazon CLI)，但您可以使用任何受支持的编程语言。

首先使用 DisconnectCustomKeyStore 来断开自定义密钥存储与 Amazon CloudHSM 集群的连接。将示例自定义密钥存储 ID cks-1234567890abcdef0 替换为实际 ID。

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

第一个示例使用 UpdateCustomKeyStore 将 Amazon CloudHSM 密钥存储的易记名称更改为 DevelopmentKeys。该命令使用 CustomKeyStoreId 参数标识 Amazon CloudHSM 密钥存储，使用 CustomKeyStoreName 指定自定义密钥存储的新名称。

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --new-custom-key-store-name DevelopmentKeys
```

以下示例将与 Amazon CloudHSM 密钥存储关联的集群更改为同一集群的其他备份。该命令使用 CustomKeyStoreId 参数标识 Amazon CloudHSM 密钥存储，使用 CloudHsmClusterId 参数指定新集群 ID。

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --cloud-hsm-cluster-id cluster-1a23b4cdefg
```

以下示例向 Amazon KMS 告知当前 kmsuser 密码为 ExamplePassword。该命令使用 CustomKeyStoreId 参数标识 Amazon CloudHSM 密钥存储，使用 KeyStorePassword 参数指定当前密码。

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --key-store-password ExamplePassword
```

最后一个命令将 Amazon CloudHSM 密钥存储重新连接到 Amazon CloudHSM 集群。您可以将自定义密钥存储保留断开状态，但必须先连接它，然后才能创建新的 KMS 密钥或使用现有 KMS 密钥来进行加密操作。将示例自定义密钥存储 ID 替换为实际 ID。

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

连接 Amazon CloudHSM 密钥存储

新的 Amazon CloudHSM 密钥存储未连接。您需要先将 Amazon KMS keys 连接到其关联的 Amazon CloudHSM 集群，然后才能在 Amazon CloudHSM 密钥存储中创建和使用这些密钥。您可以随时连接和断开 Amazon CloudHSM 密钥存储，并查看其连接状态。

您无需连接 Amazon CloudHSM 密钥存储。您可以将 Amazon CloudHSM 密钥存储保持在无限期断开的状态并且仅在您需要使用它时进行连接。但是，您可能希望定期测试连接以验证设置是否正确以及自定义密钥存储是否可以连接。

> **Note**
>
> 仅当密钥存储从未连接或您显式断开密钥存储连接时，Amazon CloudHSM 密钥存储才会具有 `DISCONNECTED` 状态。如果您的 Amazon CloudHSM 密钥存储状态为 `CONNECTED`，但您在使用它时遇到问题，请确保其关联的 Amazon CloudHSM 集群处于活动状态，并且包含至少一个活动 HSM。如需帮助解决连接失败问题，请参阅 the section called “对自定义密钥存储进行故障排除”。

当您连接 Amazon CloudHSM 密钥存储时，Amazon KMS 会查找关联的 Amazon CloudHSM 集群，连接到该集群，并以 `kmsuser` 加密用户（CU）身份登录 Amazon CloudHSM 客户端，然后轮换 `kmsuser` 密码。只要 Amazon CloudHSM 密钥存储处于连接状态，Amazon KMS 仍然保持登录到 Amazon CloudHSM 客户端的状态。

为了建立连接，Amazon KMS 在集群的 Virtual Private Cloud (VPC) 中创建一个名为 `kms-<custom key store ID>` 的安全组。该安全组具有一个允许来自集群安全组的入站流量的规则。Amazon KMS 还会在集群的私有子网的每个可用区中创建一个弹性网络接口 (ENI)。Amazon KMS 会将 ENI 添加到 `kms-<cluster ID>` 安全组和集群的安全组。每个 ENI 的描述都是 `KMS managed ENI for cluster <cluster-ID>`。

连接过程可能需要较长时间才能完成；最多 20 分钟。

在连接 Amazon CloudHSM 密钥存储之前，请验证它是否符合要求。

- 其关联的 Amazon CloudHSM 集群必须至少包含一个活动 HSM。要查找集群中的 HSM 数，请在 Amazon CloudHSM 控制台中查看集群或使用 DescribeClusters 操作。如有必要，您可以添加 HSM。
- 集群必须具有 `kmsuser` 加密用户（CU）账户，但当您连接 Amazon CloudHSM 密钥存储时，该 CU 无法登录到集群。要获取有关注销的帮助，请参阅 如何注销并重新连接。
- Amazon CloudHSM 密钥存储的连接状态不能是 `DISCONNECTING` 或 `FAILED`。要查看连接状态，请使用 Amazon KMS 控制台或 DescribeCustomKeyStores 响应。如果连接状态为 `FAILED`，请断开自定义密钥存储，修复问题，然后连接它。

如需帮助解决连接失败问题，请参阅 如何修复连接故障。

连接 Amazon CloudHSM 密钥存储后，您可以在其中创建 KMS 密钥，然后在加密操作中使用现有 KMS 密钥。

连接和重新连接到您的 Amazon CloudHSM 密钥存储

您可以在 Amazon KMS 控制台中或使用 ConnectCustomKeyStore 操作连接或重新连接您的 Amazon CloudHSM 密钥存储。

使用 Amazon KMS 控制台

要在 Amazon Web Services 管理控制台 中连接 Amazon CloudHSM 密钥存储，请首先从 Custom key stores（自定义密钥存储）页面中选择 Amazon CloudHSM 密钥存储。连接过程可能最多需要 20 分钟才能完成。

1. 登录到 Amazon Web Services 管理控制台，然后通过以下网址打开 Amazon Key Management Service（Amazon KMS）控制台：https://console.aws.amazon.com/kms。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择自定义密钥存储、Amazon CloudHSM 密钥存储。
4. 选择要连接的 Amazon CloudHSM 密钥存储的行。

   如果 Amazon CloudHSM 密钥存储的连接状态为失败，则必须在连接之前断开自定义密钥存储。
5. 从 Key store actions（密钥存储操作）菜单中选择 Connect（连接）。

Amazon KMS 将开始连接自定义密钥存储的过程。它将查找关联的 Amazon CloudHSM 集群，构建所需的网络基础设施，连接到网络基础设施，以 Amazon CloudHSM CU 身份登录到 kmsuser 集群，然后轮换 kmsuser 密码。当操作完成时，连接状态将变为已连接。

如果操作失败，则会出现一条描述失败原因的错误消息。在尝试再次连接之前，请查看 Amazon CloudHSM 密钥存储的连接状态。如果状态为失败，则必须先断开自定义密钥存储，然后再次连接。如果您需要帮助，请参阅对自定义密钥存储进行故障排除。

下一步：。the section called “在密钥库中创建 KMS Amazon CloudHSM 密钥”

使用 Amazon KMS API

要连接已断开的 Amazon CloudHSM 密钥存储，请使用 ConnectCustomKeyStore 操作。关联的 Amazon CloudHSM 集群必须包含至少一个活动 HSM，且连接状态不能为 FAILED。

连接过程需要较长时间才能完成；最多 20 分钟。除非该过程迅速失败，否则 操作将返回 HTTP 200 响应和无属性的 JSON 对象。但是，此初始响应不指示连接是否成功。要确定自定义密钥存储的连接状态，请使用 DescribeCustomKeyStores 响应。

本部分中的示例使用 Amazon Command Line Interface (Amazon CLI)，但您可以使用任何受支持的编程语言。

要确定 Amazon CloudHSM 密钥存储，请使用自定义密钥存储 ID。您可以在控制台中的 Custom key stores（自定义密钥存储）页面上或通过使用无参数的 DescribeCustomKeyStores 操作找到该 ID。在运行此示例之前，请将示例 ID 替换为有效的 ID。

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

要验证 Amazon CloudHSM 密钥存储是否已连接，请使用 DescribeCustomKeyStores 操作。默认情况下，此操作将返回您的账户和区域中的所有自定义密钥存储。但您可以使用 CustomKeyStoreId 或 CustomKeyStoreName 参数（但不能同时使用两者）将响应限制到特定自定义密钥存储。ConnectionState 值 CONNECTED 表示自定义密钥存储已连接到其 Amazon CloudHSM 集群。

> **Note**
>
> 在 DescribeCustomKeyStores 响应中添加了 CustomKeyStoreType 字段，以区分 Amazon CloudHSM 密钥存储和外部密钥存储。

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
```

```
{
   "CustomKeyStores": [
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleCloudHSMKeyStore",
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "CustomKeyStoreType": "AWS_CLOUDHSM",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "CONNECTED"
   ],
}
```

如果 ConnectionState 值为 FAILED，ConnectionErrorCode 元素将指示失败的原因。在此情况下，Amazon KMS 在您的账户中找不到集群 ID 为 Amazon CloudHSM 的 cluster-1a23b4cdefg 集群。如果您删除了该集群，则可以从原始集群的备份还原它，然后编辑自定义密钥存储的集群 ID。有关响应连接错误代码的帮助信息，请参阅 如何修复连接故障。

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
{
   "CustomKeyStores": [
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleKeyStore",
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "CustomKeyStoreType": "AWS_CLOUDHSM",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "FAILED"
      "ConnectionErrorCode": "CLUSTER_NOT_FOUND"
   ],
}
```

断开 Amazon CloudHSM 密钥存储

当您断开 Amazon CloudHSM 密钥存储时，Amazon KMS 将从 Amazon CloudHSM 客户端注销，从关联的 Amazon CloudHSM 集群断开，然后移除它创建用于支持连接的网络基础设施。

当某个 Amazon CloudHSM 密钥存储断开时，您可以管理 Amazon CloudHSM 密钥存储及其 KMS 密钥，但无法在 Amazon CloudHSM 密钥存储中创建或使用 KMS 密钥。密钥存储的连接状态为 DISCONNECTED，自定义密钥存储中的 KMS 密钥的密钥状态为 Unavailable，除非它们是 PendingDeletion。您可以随时重新连接 Amazon CloudHSM 密钥存储。

> **Note**
>
> 仅当密钥存储从未连接或您显式断开密钥存储连接时，Amazon CloudHSM 密钥存储才会具有 `DISCONNECTED` 状态。如果您的 Amazon CloudHSM 密钥存储状态为 `CONNECTED`，但您在使用它时遇到问题，请确保其关联的 Amazon CloudHSM 集群处于活动状态，并且包含至少一个活动 HSM。如需帮助解决连接失败问题，请参阅 the section called “对自定义密钥存储进行故障排除”。

当您断开自定义密钥存储时，密钥存储中的 KMS 密钥立即变得不可用（视最终一致性而定）。不过，在再次使用 KMS 密钥（例如解密数据密钥）之前，使用受 KMS 密钥保护的数据密钥加密的资源不会受到影响。此问题会影响 Amazon Web Services 服务，因为许多服务使用数据密钥来保护您的资源。有关更多信息，请参阅 不可用的 KMS 密钥如何影响数据密钥。

> **Note**
>
> 虽然自定义密钥存储已断开连接，但在自定义密钥存储中创建 KMS 密钥或在加密操作中使用现有 KMS 密钥的所有尝试都将失败。此操作可以阻止用户存储和访问敏感数据。

为了更好地估计断开自定义密钥存储的影响，请在自定义密钥存储中标识 KMS 密钥，并确定其过去的使用情况。

您可能出于以下原因断开 Amazon CloudHSM 密钥存储：

- 轮换 **`kmsuser`** 密码。每当 Amazon KMS 连接到 Amazon CloudHSM 集群时，它就会更改 `kmsuser` 密码。要强制轮换密码，只需断开并重新连接。
- 审核 Amazon CloudHSM 集群中的 KMS 密钥的密钥材料。当您断开自定义密钥存储时，Amazon KMS 会退出 Amazon CloudHSM 客户端中的 `kmsuser` 加密用户账户。这样，您便能以 `kmsuser` CU 身份登录到集群并审核和管理 KMS 密钥的密钥材料。
- 在 Amazon CloudHSM 密钥存储中立即禁用所有 KMS 密钥。您可以通过使用 Amazon Web Services 管理控制台 或 DisableKey 操作在 Amazon CloudHSM 密钥存储中禁用和重新启用 KMS 密钥。这些操作会快速完成，但它们一次只针对一个 KMS 密钥。断开 Amazon CloudHSM 密钥存储的连接会立即将 Amazon CloudHSM 密钥中的所有 KMS 密钥的密钥状态更改为 `Unavailable`，这将阻止在任何加密操作中使用这些 KMS 密钥。
- 修复失败的连接尝试。如果连接 Amazon CloudHSM 密钥存储的尝试失败（自定义密钥存储的连接状态为 `FAILED`），则必须在尝试再次连接 Amazon CloudHSM 密钥存储之前将其断开。

断开您的 Amazon CloudHSM 密钥存储

您可以在 Amazon KMS 控制台中或使用 DisconnectCustomKeyStore 操作断开 Amazon CloudHSM 密钥存储的连接。

使用 Amazon KMS 控制台断开连接

要在 Amazon KMS 控制台中断开已连接的 Amazon CloudHSM 密钥存储，请首先从自定义密钥存储页面选择该 Amazon CloudHSM 密钥存储。

1. 登录到 Amazon Web Services 管理控制台，然后通过以下网址打开 Amazon Key Management Service（Amazon KMS）控制台：https://console.aws.amazon.com/kms。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择自定义密钥存储、Amazon CloudHSM 密钥存储。
4. 选择要断开连接的外部密钥存储的行。
5. 从 Key store actions（密钥存储操作）菜单中选择 Disconnect（断开连接）。

当操作完成时，连接状态将从正在断开变为已断开连接。如果操作失败，则会出现一条错误消息，描述问题并提供有关如何修复它的帮助。如果您需要更多帮助，请参阅对自定义密钥存储进行故障排除。

使用 Amazon KMS API 断开连接

要断开已连接的 Amazon CloudHSM 密钥存储，请使用 DisconnectCustomKeyStore 操作。如果此操作成功，则 Amazon KMS 返回 HTTP 200 响应和无属性的 JSON 对象。

本部分中的示例使用 Amazon Command Line Interface (Amazon CLI)，但您可以使用任何受支持的编程语言。

本示例将断开 Amazon CloudHSM 密钥存储。在运行此示例之前，请将示例 ID 替换为有效的 ID。

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

要验证 Amazon CloudHSM 密钥存储是否已断开，请使用 DescribeCustomKeyStores 操作。默认情况下，此操作将返回您的账户和区域中的所有自定义密钥存储。但您可以使用 `CustomKeyStoreId` 和 `CustomKeyStoreName` 参数（但不能同时使用两者）将响应限制到特定自定义密钥存储。`DISCONNECTED` 的 `ConnectionState` 值表示 Amazon CloudHSM 密钥存储示例未连接到其 Amazon CloudHSM 集群。

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0
{
   "CustomKeyStores": [
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "ConnectionState": "DISCONNECTED",
      "CreationDate": "1.499288695918E9",
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleKeyStore",
      "CustomKeyStoreType": "AWS_CLOUDHSM",
      "TrustAnchorCertificate": "<certificate string appears here>"
   ],
}
```

删除 Amazon CloudHSM 密钥存储

当您删除 Amazon CloudHSM 密钥存储时，Amazon KMS 会从 KMS 中删除有关 Amazon CloudHSM 密钥存储的所有元数据，包括有关其与 Amazon CloudHSM 集群的关联的信息。此操作不会影响 Amazon CloudHSM 集群、其 HSM 或其用户。您可以创建与同一 Amazon CloudHSM 集群关联的新 Amazon CloudHSM 密钥存储，但无法撤消删除操作。

您只能删除已与其 Amazon CloudHSM 集群断开连接且不包含任何 Amazon KMS keys 的 Amazon CloudHSM 密钥存储。在删除自定义密钥存储之前，请执行以下操作。

- 验证您是否永远不需要将密钥存储中的任何 KMS 密钥用于任何加密操作。然后从密钥存储中执行所有 KMS 密钥的计划删除。有关在 Amazon CloudHSM 密钥存储中查找 KMS 密钥的帮助信息，请参阅 在 Amazon CloudHSM 密钥存储中查找 KMS 密钥。
- 确认已删除所有 KMS 密钥。要在 Amazon CloudHSM 密钥存储中查看 KMS 密钥，请参阅 the section called “识别密钥库中的 KMS Amazon CloudHSM 密钥”。
- 从其 Amazon CloudHSM 集群断开 Amazon CloudHSM 密钥存储。

请考虑断开其与关联的 Amazon CloudHSM 集群的连接，而不是删除 Amazon CloudHSM 密钥存储。当某个 Amazon CloudHSM 密钥存储断开连接时，您可以管理 Amazon CloudHSM 密钥存储及其 Amazon KMS keys。不过，您无法在 Amazon CloudHSM 密钥存储中创建或使用 KMS 密钥。您可以随时重新连接 Amazon CloudHSM 密钥存储。

删除您的 Amazon CloudHSM 密钥存储

您可以在 Amazon CloudHSM 控制台中或使用 DeleteCustomKeyStore 操作删除 Amazon KMS 密钥存储。

使用 Amazon KMS 控制台

要在 Amazon Web Services 管理控制台 中删除 Amazon CloudHSM 密钥存储，请首先从 Custom key stores（自定义密钥存储）页面中选择 Amazon CloudHSM 密钥存储。

1. 登录到 Amazon Web Services 管理控制台，然后通过以下网址打开 Amazon Key Management Service（Amazon KMS）控制台：https://console.aws.amazon.com/kms。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择自定义密钥存储、Amazon CloudHSM 密钥存储。
4. 找到表示要删除的 Amazon CloudHSM 密钥存储的行。如果 Amazon CloudHSM 密钥存储的连接状态不是已断开连接，则必须先断开 Amazon CloudHSM 密钥存储，然后再删除密钥存储。
5. 从 Key store actions（密钥存储操作）菜单中选择 Delete（删除）。

在操作完成后，会显示一条成功消息，并且 Amazon CloudHSM 密钥存储不再显示在密钥存储列表中。如果操作失败，则会显示一条错误消息，描述问题并提供有关如何解决该问题的帮助。如果您需要更多帮助，请参阅对自定义密钥存储进行故障排除。

使用 Amazon KMS API

要删除 Amazon CloudHSM 密钥存储，请使用 DeleteCustomKeyStore 操作。如果此操作成功，则 Amazon KMS 返回 HTTP 200 响应和无属性的 JSON 对象。

要开始操作，请确认 Amazon CloudHSM 密钥存储不包含任何 Amazon KMS keys。您无法删除包含 KMS 密钥的自定义密钥存储。第一个示例命令使用 ListKeys 和 DescribeKey 来搜索 Amazon CloudHSM 密钥存储中具有示例 *cks-1234567890abcdef0* 自定义密钥存储 ID 的 Amazon KMS keys。在此情况下，该命令不会返回任何 KMS 密钥。如果返回了，请使用 ScheduleKeyDeletion 操作计划每个 KMS 密钥的删除。

Bash

```
for key in $(aws kms list-keys --query 'Keys[*].KeyId' --output text) ;
do aws kms describe-key --key-id $key |
grep '"CustomKeyStoreId": "cks-1234567890abcdef0"' --context 100; done
```

PowerShell

```
PS C:\> Get-KMSKeyList | Get-KMSKey | where CustomKeyStoreId -eq
 'cks-1234567890abcdef0'
```

接下来，断开 Amazon CloudHSM 密钥存储。此示例命令使用 DisconnectCustomKeyStore 操作来断开 Amazon CloudHSM 密钥存储与其 Amazon CloudHSM 集群的连接。在运行此命令之前，请将示例自定义密钥存储 ID 替换为有效 ID。

Bash

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

PowerShell

```
PS C:\> Disconnect-KMSCustomKeyStore -CustomKeyStoreId cks-1234567890abcdef0
```

在断开自定义密钥存储后，您可以使用 DeleteCustomKeyStore 操作将其删除。

Bash

```
$ aws kms delete-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

PowerShell

```
PS C:\> Remove-KMSCustomKeyStore -CustomKeyStoreId cks-1234567890abcdef0
```

对自定义密钥存储进行故障排除

Amazon CloudHSM 密钥存储被设计为可用且有弹性。不过，为了让 Amazon CloudHSM 密钥存储正常运行，您必须修复存在的一些错误情况。

主题

- 如何修复不可用的 KMS 密钥
- 如何修复失败的 KMS 密钥
- 如何修复连接故障
- 如何响应加密操作失败
- 如何修复无效的 kmsuser 凭证
- 如何删除孤立密钥材料
- 如何恢复 KMS 密钥的已删除密钥材料
- 如何以 kmsuser 身份登录

如何修复不可用的 KMS 密钥

Amazon CloudHSM 密钥存储中 Amazon KMS keys 的密钥状态通常为 Enabled。与所有 KMS 密钥相似，当您禁用 Amazon CloudHSM 密钥存储中的 KMS 密钥或者计划删除这些密钥时，密钥状态会发生变化。但是，与其他 KMS 密钥不同，自定义密钥存储中的 KMS 密钥还可具有密钥状态 Unavailable。

密钥状态 Unavailable 表示 KMS 密钥位于被故意断开连接的自定义密钥存储中，并且尝试重新连接该集群（如果有）失败。当某个 KMS 密钥不可用时，您可以查看和管理该 KMS 密钥，但不能将其用于加密操作。

要查找 KMS 密钥的密钥状态，请在 Customer managed keys（客户托管密钥）页面上，查看 KMS 密钥的 Status（状态）字段。或者，使用 DescribeKey 操作并查看响应中的 KeyState 元素。有关更多信息，请参阅 识别和查看密钥。

已断开的自定义密钥存储中的 KMS 密钥的密钥状态将为 Unavailable 或 PendingDeletion。计划从自定义密钥存储中删除的 KMS 密钥的密钥状态为 Pending Deletion，即使自定义密钥存储已断开连接也是如此。这使您可以取消计划的密钥删除而无需重新连接自定义密钥存储。

要修复不可用的 KMS 密钥，请重新连接自定义密钥存储。重新连接自定义密钥存储后，自定义密钥存储中的 KMS 密钥的密钥状态将自动还原到之前的状态，例如 Enabled 或 Disabled。待删除的 KMS 密钥将保持 PendingDeletion 状态。但是，当问题仍然存在时，启用和禁用不可用的 KMS 密钥不会更改其密钥状态。仅当密钥变得可用时，启用或禁用操作才会生效。

如需帮助解决失败的连接，请参阅如何修复连接故障。

如何修复失败的 KMS 密钥

在 Amazon CloudHSM 密钥存储中创建和使用 KMS 密钥的问题可能由 Amazon CloudHSM 密钥存储、其关联的 Amazon CloudHSM 集群、KMS 密钥或其密钥材料导致。

当某个 Amazon CloudHSM 密钥存储与其 Amazon CloudHSM 集群断开连接时，该自定义密钥存储中 KMS 密钥的密钥状态为 Unavailable。在断开的 Amazon CloudHSM 密钥存储中创建 KMS 密钥的所有请求都将返回 CustomKeyStoreInvalidStateException 异常。所有加密、解密、重新加密或生成数据密钥的请求都将返回 KMSInvalidStateException 异常。要修复该问题，请重新连接 Amazon CloudHSM 密钥存储。

但是，您使用 Amazon CloudHSM 密钥存储 KMS 密钥进行加密操作的尝试可能会失败，即使其密钥状态为 Enabled 并且 Amazon CloudHSM 密钥存储的连接状态为 Connected 也是如此。这可能由以下任一情况导致。

- KMS 密钥的密钥材料可能已从关联的 Amazon CloudHSM 集群中删除。要进行调查，请查找 KMS 密钥的密钥材料的密钥 ID，并在必要时尝试恢复密钥材料。
- 所有 HSM 都已从与 Amazon CloudHSM 密钥存储关联的 Amazon CloudHSM 集群中删除。要在加密操作中使用 Amazon CloudHSM 密钥存储中的某个 KMS 密钥，其 Amazon CloudHSM 集群必须至少包含一个活动 HSM。要验证 Amazon CloudHSM 集群中的 HSM 的数量和状态，请使用 Amazon CloudHSM 控制台或 DescribeClusters 操作。要向集群添加 HSM，请使用 Amazon CloudHSM 控制台或 CreateHsm 操作。
- 与 Amazon CloudHSM 密钥存储关联的 Amazon CloudHSM 集群已删除。要修复该问题，请从与原始集群相关的备份创建一个集群，例如原始集群的备份或用于创建原始集群的备份。然后，在自定义密钥存储设置中编辑集群 ID。有关说明，请参阅如何恢复 KMS 密钥的已删除密钥材料。
- 与自定义密钥存储关联的 Amazon CloudHSM 集群没有任何可用的 PKCS #11 会话。这种情况通常发生在高突发流量期间，此时需要额外的会话来服务流量。要响应带有关 PKCS #11 会话的错误消息的 `KMSInternalException`，请退后并重试该请求。

如何修复连接故障

如果您尝试连接 Amazon CloudHSM 密钥存储到其 Amazon CloudHSM 集群，但操作失败，则 Amazon CloudHSM 密钥存储的连接状态将更改为 `FAILED`。要查找 Amazon CloudHSM 密钥存储的连接状态，请使用 Amazon KMS 控制台或 DescribeCustomKeyStores 操作。

另外，由于容易检测到集群配置错误，一些连接尝试会很快失败。在这种情况下，连接状态仍为 `DISCONNECTED`。这些失败将返回错误消息或例外来说明尝试失败的原因。查看例外描述和集群要求，纠正问题，更新 Amazon CloudHSM 密钥存储（如有必要）并尝试重新连接。

当连接状态为 `FAILED` 时，运行 DescribeCustomKeyStores 操作并查看响应中的 `ConnectionErrorCode` 元素。

> **Note**
>
> 当 Amazon CloudHSM 密钥存储的连接状态为 `FAILED` 时，您必须先断开 Amazon CloudHSM 密钥存储，然后再尝试重新连接它。您无法连接具有 `FAILED` 连接状态的 Amazon CloudHSM 密钥存储。

- `CLUSTER_NOT_FOUND` 表示 Amazon KMS 找不到具有指定集群 ID 的 Amazon CloudHSM 集群。这可能是因为向 API 操作提供了错误的集群 ID，或者集群被删除而不是被替换。要修复此错误，请验证集群 ID，例如通过使用 Amazon CloudHSM 控制台或 DescribeClusters 操作。如果集群已被删

除，请从源的最新备份创建一个集群。然后断开 Amazon CloudHSM 密钥存储连接，编辑 Amazon CloudHSM 密钥存储集群 ID 设置，然后将 Amazon CloudHSM 密钥存储重新连接到集群。

- INSUFFICIENT_CLOUDHSM_HSMS 表示关联的 Amazon CloudHSM 集群不包含任何 HSM。要连接，集群必须至少具有一个 HSM。要查找集群中的 HSM 数，请使用 DescribeClusters 操作。要解决此错误，请添加至少一个 HSM 到集群。如果您添加了多个 HSM，最好在不同的可用区中创建它们。
- INSUFFICIENT_FREE_ADDRESSES_IN_SUBNET 表明 Amazon KMS 无法将 Amazon CloudHSM 密钥存储连接至其 Amazon CloudHSM 集群，因为至少有一个与集群关联的私有子网没有任何可用的 IP 地址。Amazon CloudHSM 密钥存储连接需要在每个关联的私有子网中有一个空闲的 IP 地址，但最好有两个空闲的 IP 地址。

  您无法将 IP 地址（CIDR 块）添加到现有子网。如有可能，请移动或删除在子网中使用 IP 地址的其他资源，例如未使用的 EC2 实例或弹性网络接口。除此之外，您可以通过 Amazon CloudHSM 集群的近期备份创建一个集群，集群中包含具有更多可用地址空间的新的或现有私有子网。然后，要将新集群与您的 Amazon CloudHSM 密钥存储关联，请断开自定义密钥存储，将 Amazon CloudHSM 密钥存储的集群 ID 更改为新集群的 ID，然后尝试再次连接。

  > **Tip**
  >
  > 要避免重置 kmsuser 密码，请使用 Amazon CloudHSM 集群的最新备份。

- INTERNAL_ERROR 指示 Amazon KMS 因内部错误而无法完成请求。重试请求。对于 ConnectCustomKeyStore 请求，先断开 Amazon CloudHSM 密钥存储，然后再尝试重新连接它。
- INVALID_CREDENTIALS 表示 Amazon KMS 无法登录到关联的 Amazon CloudHSM 集群，因为它没有正确的 kmsuser 账户密码。如需帮助解决此错误，请参阅如何修复无效的 kmsuser 凭证。
- NETWORK_ERRORS 通常表示暂时性网络问题。断开 Amazon CloudHSM 密钥存储，等待几分钟，然后重试连接。
- SUBNET_NOT_FOUND 表示 Amazon CloudHSM 集群配置中至少有一个子网已被删除。如果 Amazon KMS 无法找到集群配置中的所有子网，则尝试将 Amazon CloudHSM 密钥存储连接到 Amazon CloudHSM 集群将失败。

  要修复此错误，请从同一 Amazon CloudHSM 集群的最近备份创建集群。（此过程使用 VPC 和私有子网创建新的集群配置。）验证新集群是否满足自定义密钥存储的要求，并记下新集群 ID。然后，要将新集群与您的 Amazon CloudHSM 密钥存储关联，请断开自定义密钥存储，将 Amazon CloudHSM 密钥存储的集群 ID 更改为新集群的 ID，然后尝试再次连接。

> **Tip**
>
> 要避免重置 kmsuser 密码，请使用 Amazon CloudHSM 集群的最新备份。

- USER_LOCKED_OUT 表示 kmsuser 加密用户 (CU) 账户无法访问关联的 Amazon CloudHSM 集群，因为失败的密码尝试过多。如需帮助解决此错误，请参阅如何修复无效的 kmsuser 凭证。

  要修复此错误，请断开 Amazon CloudHSM 密钥存储并使用 CloudHSM CLI 中的 user change-password 命令以更改 kmsuser 账户密码。然后，编辑自定义密钥存储的 kmsuser 密码设置并重试连接。如需帮助，请使用如何修复无效的 kmsuser 凭证主题中所述的过程。
- USER_LOGGED_IN 表示 kmsuser CU 帐户已登录到关联的 Amazon CloudHSM 集群。这会阻止 Amazon KMS 轮换 kmsuser 账户密码和登录到群集。要修复此错误，请从集群中注销 kmsuser CU。如果您更改了用于登录到集群的 kmsuser 密码，则还必须更新 Amazon CloudHSM 密钥存储的密钥存储密码值。有关帮助信息，请参阅 如何注销并重新连接。
- USER_NOT_FOUND 表示 Amazon KMS 无法在关联的 kmsuser 集群中找到 Amazon CloudHSM CU 账户。要修复此错误，请在集群中创建 kmsuser CU 账户，然后更新 Amazon CloudHSM 密钥存储的密钥存储密码值。有关帮助信息，请参阅 如何修复无效的 kmsuser 凭证。

如何响应加密操作失败

在自定义密钥存储中使用 KMS 密钥的加密操作可能会失败，并显示 KMSInvalidStateException。以下错误消息可能附带 KMSInvalidStateException。

KMS 无法与您的 CloudHSM 集群进行通信。这可能是暂时的网络问题。如果您反复看到此错误，请验证 Amazon CloudHSM 集群 VPC 的网络 ACL 和安全组规则是否正确。

- 虽然它是 HTTPS 400 错误，但它可能是由于暂时的网络问题引起的。要进行响应，首先重试请求。但是，如果继续失败，请检查网络组件的配置。此错误很可能是由于网络组件（例如阻止传出流量的防火墙规则或 VPC 安全组规则）的错误配置引起的。例如，KMS 无法通过 IPv6 与 Amazon CloudHSM 集群通信。有关先决条件的详细信息，请参阅创建 Amazon CloudHSM 密钥存储。

KMS 无法与您的 Amazon CloudHSM 集群通信，因为 kmsuser 已被锁定。如果您反复看到此错误，请断开 Amazon CloudHSM 密钥存储的连接并重置 kmsuser 账户的密码。更新自定义密钥存储的 kmsuser 密码，然后重试请求。

- 此错误消息表示 kmsuser 加密用户（CU）账户无法访问关联的 Amazon CloudHSM 集群，因为失败的密码尝试过多。如需帮助解决此错误，请参阅如何断开和登录。

如何修复无效的 **kmsuser** 凭证

连接 Amazon CloudHSM 密钥存储时，Amazon KMS 会以 kmsuser 加密用户（CU）身份登录到关联的 Amazon CloudHSM 集群。在 Amazon CloudHSM 密钥存储断开之前，它将保持登录状态。DescribeCustomKeyStores 响应将显示 ConnectionState FAILED 和 ConnectionErrorCode 值 INVALID_CREDENTIALS，如以下示例所示。

如果您断开 Amazon CloudHSM 密钥存储并更改 kmsuser 密码，Amazon KMS 将无法使用 kmsuser CU 账户的凭证登录到 Amazon CloudHSM 集群。因此，所有连接 Amazon CloudHSM 密钥存储的尝试都将失败。DescribeCustomKeyStores 响应显示 ConnectionState 值 FAILED 以及 ConnectionErrorCode 值 INVALID_CREDENTIALS，如以下示例中所示。

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleKeyStore
{
   "CustomKeyStores": [
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "ConnectionErrorCode": "INVALID_CREDENTIALS"
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleKeyStore",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "FAILED"
   ],
}
```

此外，在使用不正确的密码登录到集群的尝试失败五次后，Amazon CloudHSM 将锁定用户账户。要登录到集群，您必须更改账户密码。

如果 Amazon KMS 在尝试以 kmsuser CU 身份登录到集群时获得锁定响应，连接 Amazon CloudHSM 密钥存储的请求将失败。DescribeCustomKeyStores 响应包含 ConnectionState FAILED 和 ConnectionErrorCode 值 USER_LOCKED_OUT，如以下示例所示。

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleKeyStore
{
   "CustomKeyStores": [
      "CloudHsmClusterId": "cluster-1a23b4cdefg",
      "ConnectionErrorCode": "USER_LOCKED_OUT"
```

```
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleKeyStore",
      "TrustAnchorCertificate": "<certificate string appears here>",
      "CreationDate": "1.499288695918E9",
      "ConnectionState": "FAILED"
   ],
}
```

要修复上述任一状况，请使用以下过程。

1. 断开 Amazon CloudHSM 密钥存储。
2. 运行 DescribeCustomKeyStores 操作并查看响应中的 ConnectionErrorCode 元素的值。
   - 如果 ConnectionErrorCode 值为 INVALID_CREDENTIALS，请确定 kmsuser 账户的当前密码。必要时，请使用 CloudHSM CLI 中的 user change-password 命令将密码设置为已知值。
   - 如果 ConnectionErrorCode 值是 USER_LOCKED_OUT，您必须使用 CloudHSM CLI 中的 user change-password 命令更改 kmsuser 密码。
3. 编辑 kmsuser 密码设置，使其与当前集群中的 kmsuser 密码匹配。此操作将告知 Amazon KMS 要用来登录到集群的密码。它不会更改集群中的 kmsuser 密码。
4. 连接自定义密钥存储。

如何删除孤立密钥材料

在计划从 Amazon CloudHSM 密钥存储中删除 KMS 密钥后，您可能需要从关联的 Amazon CloudHSM 集群中手动删除对应的密钥材料。

当您在 Amazon CloudHSM 密钥存储中创建 KMS 密钥时，Amazon KMS 将在 Amazon KMS 中创建 KMS 密钥元数据并在关联的 Amazon CloudHSM 集群中生成密钥材料。当您计划在 Amazon CloudHSM 密钥存储中删除 KMS 密钥时，在等待期过后，Amazon KMS 将删除 KMS 密钥元数据。然后，Amazon KMS 将尽可能从 Amazon CloudHSM 集群中删除对应的密钥材料。如果 Amazon KMS 无法访问集群（例如与 Amazon CloudHSM 密钥存储断开连接或 kmsuser 密码更改），尝试可能会失败。Amazon KMS 不会尝试从集群备份中删除密钥材料。

Amazon KMS 会报告其尝试从 Amazon CloudTrail 日志 DeleteKey 事件条目中的集群中删除密钥材料的结果。它会在 additionalEventData 元素的 backingKeysDeletionStatus 元素中显示，如以下示例条目所示。该条目还包含 KMS 密钥 ARN、Amazon CloudHSM 集群 ID 和密钥材料的 ID（backing-key-id）。

```
{
```

```
    "eventVersion": "1.08",
    "userIdentity": {
        "accountId": "111122223333",
        "invokedBy": "Amazon Internal"
    },
    "eventTime": "2021-12-10T14:23:51Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "DeleteKey",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "Amazon Internal",
    "userAgent": "AWS Internal",
    "requestParameters": null,
    "responseElements":  {
        "keyId":"arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "additionalEventData": {
        "customKeyStoreId": "cks-1234567890abcdef0",
        "clusterId": "cluster-1a23b4cdefg",
        "backingKeys": "[{\"backingKeyId\":\"backing-key-id\"}]",
        "backingKeysDeletionStatus": "[{\"backingKeyId\":\"backing-key-id\",
\"deletionStatus\":\"FAILURE\"}]"
    },
    "eventID": "c21f1f47-f52b-4ffe-bff0-6d994403cf40",
    "readOnly": false,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:eu-
west-1:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsServiceEvent",
    "recipientAccountId": "111122223333",
    "managementEvent": true,
    "eventCategory": "Management"
}
```

备注

以下过程使用 Amazon CloudHSM 客户端 SDK 5 命令行工具 CloudHSM CLI。CloudHSM CLI 将 `key-handle` 替换为 `key-reference`。

2025 年 1 月 1 日，Amazon CloudHSM 将终止对客户端 SDK 3 命令行工具、CloudHSM 管理实用程序（CMU）和密钥管理实用程序（KMU）的支持。有关客户端 SDK 3 命令行工具和客户端 SDK 5 命令行工具之间区别的更多信息，请参阅《Amazon CloudHSM 用户指南》中的从客户端 SDK 3 CMU 和 KMU 迁移到客户端 SDK 5 CloudHSM CLI。

以下过程演示如何从关联的 Amazon CloudHSM 集群中删除孤立密钥材料。

1. 断开 Amazon CloudHSM 密钥存储（如果尚未断开），然后登录，如 如何断开和登录 中所述。

   > **Note**
   >
   > 虽然自定义密钥存储已断开连接，但在自定义密钥存储中创建 KMS 密钥或在加密操作中使用现有 KMS 密钥的所有尝试都将失败。此操作可以阻止用户存储和访问敏感数据。

2. 使用 CloudHSM CLI 中的 key delete 命令从集群内的 HSM 中删除密钥。

   使用 Amazon CloudHSM 密钥存储中的 KMS 密钥进行加密操作的所有 CloudTrail 日志条目都包含带有 `customKeyStoreId` 和 `backingKey` 的 `additionalEventData` 字段。`backingKeyId` 字段中返回的值是 CloudHSM 密钥 `id` 属性。我们建议按 `id` 筛选 key delete 操作，从而删除您在 CloudTrail 日志中识别出的孤立密钥材料。

   Amazon CloudHSM 将 `backingKeyId` 值识别为十六进制值。要按 `id` 筛选，必须在 `backingKeyId` 前面加上 `0x`。例如，如果 CloudTrail 日志中的 `backingKeyId` 是 `1a2b3c45678abcdef`，则可以按 **`0x`**`1a2b3c45678abcdef` 进行筛选。

   以下示例从集群中的 HSM 中删除密钥。`backing-key-id` 列在 CloudTrail 日志条目中。在运行此命令之前，请将示例 `backing-key-id` 替换为您账户中的有效密钥 ID。

   ```
   aws-cloudhsm key delete --filter attr.id="0x<backing-key-id>"
   {
     "error_code": 0,
     "data": {
       "message": "Key deleted successfully"
     }
   }
   ```

3. 注销并重新连接 Amazon CloudHSM 密钥存储，如 如何注销并重新连接 中所述。

如何恢复 KMS 密钥的已删除密钥材料

如果删除了 Amazon KMS key 的密钥材料，则 KMS 密钥不可用，并且在 KMS 密钥下加密的所有密文都无法解密。如果从关联的 Amazon CloudHSM 集群中删除了 Amazon CloudHSM 密钥存储中的 KMS 密钥的密钥材料，则会出现这种情况。但是，可以恢复该密钥材料。

当您在 Amazon CloudHSM 密钥存储中创建 Amazon KMS key（KMS 密钥）时，Amazon KMS 将登录到关联的 Amazon CloudHSM 集群并创建 KMS 密钥的密钥材料。它还会将密码更改为只有它知道的值，并且只要连接了 Amazon CloudHSM 密钥存储，它就会保持登录状态。由于只有密钥所有者（即创建密钥的 CU）才能删除密钥，因此不太可能意外地从 HSM 中删除密钥。

但是，如果从集群内的 HSM 中删除 KMS 密钥的密钥材料，KMS 密钥的密钥状态最终将变为 `UNAVAILABLE`。如果您尝试使用 KMS 密钥进行加密操作，该操作将失败并出现 `KMSInvalidStateException` 异常。最重要的是，在 KMS 密钥下加密的任何数据都无法解密。

在某些情况下，您可以通过从包含密钥材料的备份创建集群来恢复已删除的密钥材料。仅当在密钥存在且未被删除的情况下创建了一个备份时，此策略才有效。

使用以下过程恢复密钥材料。

1. 查找包含密钥材料的集群备份。备份还必须包含支持集群及其加密数据所需的所有用户和密钥。

   使用 DescribeBackups 操作列出集群的备份。然后，使用备份时间戳帮助您选择一个备份。要将输出限制为与 Amazon CloudHSM 密钥存储关联的集群，请使用 `Filters` 参数，如以下示例所示。

   ```
   $ aws cloudhsmv2 describe-backups --filters clusterIds=<cluster ID>
   {
       "Backups": [
           {
               "ClusterId": "cluster-1a23b4cdefg",
               "BackupId": "backup-9g87f6edcba",
               "CreateTimestamp": 1536667238.328,
               "BackupState": "READY"
           },
                ...
       ]
   }
   ```

2. 从所选备份创建集群。验证备份是否包含已删除的密钥以及集群所需的其他用户和密钥。
3. 断开 Amazon CloudHSM 密钥存储，以便您可以编辑其属性。

4. 编辑 Amazon CloudHSM 密钥存储的集群 ID。输入您从备份创建的集群的集群 ID。由于该集群与原始集群共享备份历史记录，新集群 ID 应该是有效的。
5. 重新连接 Amazon CloudHSM 密钥存储。

如何以 **kmsuser** 身份登录

为了创建和管理 Amazon CloudHSM 密钥存储的 Amazon CloudHSM 集群中的密钥材料，Amazon KMS 将使用 kmsuser 加密用户（CU）账户。您在集群中创建 kmsuser CU 账户并在创建 Amazon CloudHSM 密钥存储时将该账户的密码提供给 Amazon KMS。

一般而言，Amazon KMS 将管理 kmsuser 账户。但是，对于某些任务，您需要断开 Amazon CloudHSM 密钥存储，以 kmsuser CU 身份登录到集群，并使用 CloudHSM 命令行界面（CLI）。

> **Note**
>
> 虽然自定义密钥存储已断开连接，但在自定义密钥存储中创建 KMS 密钥或在加密操作中使用现有 KMS 密钥的所有尝试都将失败。此操作可以阻止用户存储和访问敏感数据。

本主题介绍了如何断开 Amazon CloudHSM 密钥存储并以 kmsuser 身份登录、运行 Amazon CloudHSM 命令行工具以及注销并重新连接 Amazon CloudHSM 密钥存储。

主题

- 如何断开和登录
- 如何注销并重新连接

如何断开和登录

每当需要以 kmsuser 加密用户身份登录到关联的集群时，请使用以下过程。

> **备注**
>
> 以下过程使用 Amazon CloudHSM 客户端 SDK 5 命令行工具 CloudHSM CLI。CloudHSM CLI 将 key-handle 替换为 key-reference。
> 2025 年 1 月 1 日，Amazon CloudHSM 将终止对客户端 SDK 3 命令行工具、CloudHSM 管理实用程序（CMU）和密钥管理实用程序（KMU）的支持。有关客户端 SDK 3 命令行工具和客

户端 SDK 5 命令行工具之间区别的更多信息，请参阅《Amazon CloudHSM 用户指南》中的从客户端 SDK 3 CMU 和 KMU 迁移到客户端 SDK 5 CloudHSM CLI。

1. 断开 Amazon CloudHSM 密钥存储（如果尚未断开）。可以使用 Amazon KMS 控制台或 Amazon KMS API。

   当您的 Amazon CloudHSM 密钥已连接时，您便已经以 kmsuser 身份登录 Amazon KMS。这将防止您以 kmsuser 身份登录或更改 kmsuser 密码。

   例如，此命令使用 DisconnectCustomKeyStore 断开示例密钥存储。将示例 Amazon CloudHSM 密钥存储 ID 替换为有效的 ID。

   ```
   $ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
   ```

2. 使用 login 命令并以管理员身份登录。使用《Amazon CloudHSM 用户指南》的使用 CloudHSM CLI 部分中描述的步骤。

   ```
   aws-cloudhsm > login --username admin --role admin
             Enter password:
   {
     "error_code": 0,
     "data": {
       "username": "admin",
       "role": "admin"
     }
   }
   ```

3. 使用 CloudHSM CLI 中的 user change-password 命令将 kmsuser 账户的密码更改为您知道的值。（Amazon KMS 将在您连接 Amazon CloudHSM 密钥存储时轮换密码。）密码必须由 7 到 32 个字母数字字符组成。它区分大小写，并且不能包含任何特殊字符。

4. 使用您设置的密码以 kmsuser 身份登录。有关详细说明，请参阅《Amazon CloudHSM 用户指南》中的使用 CloudHSM CLI 部分。

   ```
   aws-cloudhsm > login --username kmsuser --role crypto-user
             Enter password:
   {
     "error_code": 0,
     "data": {
       "username": "kmsuser",
   ```

```
    "role": "crypto-user"
  }
}
```

如何注销并重新连接

每次需要以 kmsuser 加密用户身份注销并重新连接密钥存储时，请按以下步骤操作。

> **备注**
>
> 以下过程使用 Amazon CloudHSM 客户端 SDK 5 命令行工具 CloudHSM CLI。CloudHSM CLI 将 key-handle 替换为 key-reference。
> 2025 年 1 月 1 日，Amazon CloudHSM 将终止对客户端 SDK 3 命令行工具、CloudHSM 管理实用程序（CMU）和密钥管理实用程序（KMU）的支持。有关客户端 SDK 3 命令行工具和客户端 SDK 5 命令行工具之间区别的更多信息，请参阅《Amazon CloudHSM 用户指南》中的从客户端 SDK 3 CMU 和 KMU 迁移到客户端 SDK 5 CloudHSM CLI。

1. 执行任务，然后使用 CloudHSM CLI 中的 logout 命令注销。如果您不注销，重新连接 Amazon CloudHSM 密钥存储的尝试将失败。

```
aws-cloudhsm  logout
{
  "error_code": 0,
  "data": "Logout successful"
}
```

2. 为自定义密钥存储编辑 kmsuser 密码设置。

   这将告知 Amazon KMS 集群中的 kmsuser 的当前密码。如果忽略此步骤，Amazon KMS 将无法以 kmsuser 身份登录到集群，并且重新连接自定义密钥存储的所有尝试都将失败。您可以使用 Amazon KMS 控制台或 UpdateCustomKeyStore 操作的 KeyStorePassword 参数。

   例如，以下命令将告知 Amazon KMS 当前密码为 tempPassword。将示例密码替换为实际密码。

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --
key-store-password tempPassword
```

3. 重新连接 Amazon KMS 密钥存储至其 Amazon CloudHSM 集群。将示例 Amazon CloudHSM 密钥存储 ID 替换为有效的 ID。在连接过程中，Amazon KMS 会将 `kmsuser` 密码更改为只有它知道的值。

   ConnectCustomKeyStore 操作将快速返回，但连接过程可能需要更长时间。初始响应不指示连接过程是否成功。

   ```
   $ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
   ```

4. 使用 DescribeCustomKeyStores 操作验证 Amazon CloudHSM 密钥存储是否已连接。将示例 Amazon CloudHSM 密钥存储 ID 替换为有效的 ID。

   在本示例中，连接状态字段表明 Amazon CloudHSM 密钥存储现已连接。

   ```
   $ aws kms describe-custom-key-stores --custom-key-store-
   id cks-1234567890abcdef0
   {
      "CustomKeyStores": [
         "CustomKeyStoreId": "cks-1234567890abcdef0",
         "CustomKeyStoreName": "ExampleKeyStore",
         "CloudHsmClusterId": "cluster-1a23b4cdefg",
         "TrustAnchorCertificate": "<certificate string appears here>",
         "CreationDate": "1.499288695918E9",
         "ConnectionState": "CONNECTED"
      ],
   }
   ```

外部密钥存储

外部密钥存储允许您使用外部的加密密钥来保护您的 Amazon 资源。 Amazon此高级功能专为受监管的工作负载设计，这些工作负载必须使用存储在您控制的外部密钥管理系统中的加密密钥加以保护。外部密钥存储支持Amazon 数字主权承诺，为您提供对数据的主权控制权 Amazon，包括能够使用您拥有并在外部控制的密钥材料进行加密 Amazon。

外部密钥存储库是由您拥有并在外部管理的外部密钥管理器支持的自定义密钥存储库 Amazon。您的外部密钥管理器可以是物理或虚拟硬件安全模块 (HSMs)，也可以是任何能够生成和使用加密密钥的基于硬件或软件的系统。使用外部密钥存储库中的 KMS 密钥的加密和解密操作由您的外部密钥管理器使用您的加密密钥材料执行，该功能被称为 “保留自己的密钥” ()。HYOKs

Amazon KMS 切勿直接与外部密钥管理器交互，也无法创建、查看、管理或删除您的密钥。相反，仅与您提供的外部密钥存储代理（XKS 代理）软件进行 Amazon KMS 交互。您的外部密钥存储代理负责调解与您的外部密钥管理器 Amazon KMS 之间的所有通信。它会将来自 Amazon KMS 您的外部密钥管理器的所有请求传送回您的外部密钥管理器，并将来自外部密钥管理器的响应传回到。 Amazon KMS外部密钥存储代理还将来自 Amazon KMS 的通用请求转换为外部密钥管理器可以理解的供应商特定格式，允许您将外部密钥存储与来自不同供应商的密钥管理器一起使用。

您可以在外部密钥存储中使用 KMS 密钥进行客户端加密，包括使用 Amazon Encryption SDK。但是外部密钥存储是服务器端加密的重要资源，它允许您使用外部的加密密钥对 Amazon 资源 Amazon Web Services 服务 进行多重保护。 Amazon Amazon Web Services 服务 支持用于对称加密的客户托管密钥还支持外部密钥存储中的 KMS 密钥。有关服务支持的详细信息，请参阅 Amazon 服务集成。

外部密钥存储允许您 Amazon KMS 用于受监管的工作负载，在这些工作负载中，加密密钥必须在外部存储和使用 Amazon。不过，这样的工作负载与标准责任共担模型相去甚远，会造成额外的运营负担。对大多数客户而言，可用性和延迟的更大风险将超过外部密钥存储的预期安全优势。

外部密钥存储允许您控制信任根密钥。在外部密钥存储中以 KMS 密钥加密的数据，只能使用您控制的外部密钥管理器进行解密。如果您暂时撤消对外部密钥管理器的访问权限，例如断开外部密钥存储库的连接或断开外部密钥管理器与外部密钥存储代理的连接，则在恢复加密密钥之前，将 Amazon 失去对加密密钥的所有访问权限。在此期间，无法解密以 KMS 密钥加密的加密文字。如果您永久撤消对外部密钥管理器的访问权限，在外部密钥存储中以 KMS 密钥加密的所有加密文字都不可恢复。唯一的例外是那些会短暂缓存受您的 KMS 密钥保护的数据密钥的 Amazon 服务。这些数据密钥将继续有效，直到您停用资源或缓存过期。有关更多信息，请参阅 不可用的 KMS 密钥如何影响数据密钥。

外部密钥存储可以解锁受监管工作负载的少数用例，在这些用例中，加密密钥必须完全由您控制且无法访问 Amazon。不过，这是您运营基于云的基础设施方式的重大变化，也是责任共担模型的重大转变。对大多数工作负载而言，额外的运营负担以及可用性和性能的更大风险将超过外部密钥存储所带来的预期安全优势。

我需要外部密钥存储吗？

对于大多数用户来说，默认 Amazon KMS 密钥存储区受到 FIPS 140-3 安全等级 3 验证的硬件安全模块的保护，可以满足他们的安全、控制和监管要求。外部密钥存储用户会承担大量成本、维护和故障排除负担，以及与延迟、可用性和可靠性有关的风险。

在考虑外部密钥存储时，请花点时间了解替代方案，包括由您拥有和管理的 Amazon CloudHSM 集群支持的密Amazon CloudHSM 钥存储库，以及包含您自己生成 HSMs 并可以按需从 KMS 密钥中删除

的导入密钥材料的 KMS 密钥。特别要注意的是，导入过期间隔非常短的密钥材料可以提供类似级别的控制，却不会带来性能或可用性风险。

如果您有以下要求，则外部密钥存储可能是适合您组织的解决方案：

- 您需要在本地密钥管理器或您控制之外的密钥管理器中使用加密密钥。 Amazon
- 您必须证明，在云端之外，您加密密钥的保留完全由自己控制。
- 您必须使用具有独立授权的加密密钥进行加密和解密。
- 密钥材料必须受辅助、独立的审计路径的约束。

如果您选择外部密钥存储，请将其使用限制在需要使用 Amazon之外的加密密钥进行保护的工作负载。

责任共担模式

标准 KMS 密钥使用在中生成和使用的密钥材料 HSMs ，由其 Amazon KMS 拥有和管理。您可以在 KMS 密钥上建立访问控制策略，并配置 Amazon Web Services 服务 该策略使用 KMS 密钥来保护您的资源。 Amazon KMS 负责您的 KMS 密钥中密钥材料的安全性、可用性、延迟和持久性。

外部密钥存储中的 KMS 密钥依赖外部密钥管理器中的密钥材料和操作。因此，责任的天平朝着您的方向移动。您对外部密钥管理器中加密密钥的安全性、可靠性、耐用性和性能负责。 Amazon KMS 负责迅速响应请求并与您的外部密钥存储代理进行通信，并负责维护我们的安全标准。为确保每个外部密钥存储的密文至少与标准密文一样强，请 Amazon KMS 先使用您的 KMS Amazon KMS 密钥特有的 Amazon KMS 密钥材料对所有明文进行加密，然后将其发送到您的外部密钥管理器以使用您的外部密钥进行加密，这种过程称为双重加密。因此，无论是 Amazon KMS 还是外部密钥材料的所有者，都无法单独解密双重加密的加密文字。

您要对以下事项负责：维护符合您监管和性能标准的外部密钥管理器，提供和维护符合 Amazon KMS 外部密钥存储代理 API 规范的外部密钥存储代理，以及确保密钥材料的可用性和持久性。您还必须创建、配置和维护外部密钥存储。当出现由您维护的组件引起的错误时，您必须做好识别和解决错误的准备，以便 Amazon 服务能够在不造成不当干扰的情况下访问您的资源。 Amazon KMS 提供故障排除指导，帮助您确定问题的原因和最可能的解决方案。

查看 Amazon KMS 记录外部密钥存储的 Amazon CloudWatch 指标和维度。 Amazon KMS 强烈建议您创建 CloudWatch 警报来监控您的外部密钥存储，这样您就可以在性能和操作问题出现之前检测到这些问题的早期迹象。

发生了什么变化？

外部密钥存储仅支持对称加密 KMS 密钥。在内部 Amazon KMS，您使用和管理外部密钥存储库中的 KMS 密钥的方式与管理其他客户托管密钥的方式大致相同，包括设置访问控制策略和监控密钥使用情况。对于用于任何 KMS 密钥的外部密钥存储库中的 KMS 密钥，您可以使用相同的参数来请求加密操作。 APIs 定价也与标准 KMS 密钥相同。有关详细信息，请参阅 外部密钥存储中的 KMS 密钥 和 Amazon Key Management Service 定价。

不过，使用外部密钥存储时，以下原则会发生变化：

- 您对密钥操作的可用性、持久性和延迟负责。
- 您对外部密钥管理器系统的开发、购买、运营和许可的所有费用负责。
- 您可以对来自 Amazon KMS 外部密钥存储代理的所有请求实施独立授权。
- 您可以监控、审核和记录外部密钥存储代理的所有操作以及外部密钥管理器中与 Amazon KMS 请求相关的所有操作。

从何处开始？

要创建和管理外部密钥存储，您需要选择外部密钥存储代理连接选项、汇编先决条件，然后创建和配置外部密钥存储。

配额

Amazon KMS 允许每个 Amazon Web Services 账户 和区域中最多有 10 个自定义密钥存储库，包括Amazon CloudHSM 密钥存储库和外部密钥存储库，无论其连接状态如何。此外，使用外部密钥存储中 KMS 密钥 存在 Amazon KMS 请求限额。

如果您为外部密钥存储代理选择 VPC 代理连接，则所需组件（例如 VPCs子网和网络负载均衡器）也可能有配额。有关这些限额的信息，请使用服务限额控制台。

区域

为了最大限度地减少网络延迟，请在离外部密钥管理器最近的 Amazon Web Services 区域 中创建外部密钥存储组件。如果可行，请选择网络往返时间（RTT）不超过 35 毫秒的区域。

除中国（北京）和中国（宁夏）外， Amazon KMS 所有支持外部密钥存储的 Amazon Web Services 区域 地区均支持外部密钥存储。

不支持的功能

Amazon KMS 不支持自定义密钥存储库中的以下功能。

- 非对称 KMS 密钥
- HMAC KMS 密钥
- 具有导入密钥材料的 KMS 密钥
- 自动密钥轮换
- 多区域密钥

了解更多：

- Amazon 新闻博客中的宣布推出 Amazon KMS 外部密钥存储。

外部密钥存储概念

了解外部密钥存储中使用的基本术语和概念。

外部密钥存储

外部密钥存储库是由您拥有和管理的外部密钥管理器支持的 Amazon KMS 自定义密钥存储库。Amazon 外部密钥存储中的每个 KMS 密钥都与您外部密钥管理器中的外部密钥相关联。在外部密钥存储中使用 KMS 密钥进行加密或解密时，该操作将在您的外部密钥管理器中使用您的外部密钥执行，这种安排称为持有自己的密钥（HYOK）。此功能专为需要在自己的外部密钥管理器中维护加密密钥的组织设计。

外部密钥存储可确保保护您的 Amazon 资源的加密密钥和操作保留在您的外部密钥管理器中，由您控制。 Amazon KMS 向您的外部密钥管理器发送请求以加密和解密数据，但 Amazon KMS 无法创建、删除或管理任何外部密钥。来自 Amazon KMS 外部密钥管理器的所有请求均由您提供、拥有和管理的外部密钥存储代理软件组件进行中介。

Amazon 支持 Amazon KMS 客户托管密钥的服务可以使用外部密钥存储区中的 KMS 密钥来保护您的数据。因此，您的数据最终由使用您外部密钥管理器中的加密操作的密钥进行保护。

与标准 KMS 密钥相比，外部密钥存储中的 KMS 密钥具有根本上不同的信任模型、责任共担安排和性能预期。使用外部密钥存储时，您要对密钥材料和加密操作的安全性和完整性负责。外部密钥存储中 KMS 密钥的可用性和延迟情况，受硬件、软件、网络组件以及 Amazon KMS 与外部密钥管理器之间的距离的影响。您还可能会为外部密钥管理器以及外部密钥管理器与之通信所需的网络和负载平衡基础设施支付额外费用 Amazon KMS

您可以将外部密钥存储作为更广泛的数据保护策略的一部分加以使用。对于您保护的每项 Amazon 资源，您可以决定哪些资源需要在外部密钥存储中使用 KMS 密钥，哪些可以由标准 KMS 密钥保护。这可以让您灵活地为特定的数据分类、应用程序或项目选择 KMS 密钥。

外部密钥管理器

外部密钥管理器是 Amazon 之外的组件，可以生成 256 位 AES 对称密钥并执行对称加密和解密。外部密钥存储的外部密钥管理器，可以是实体硬件安全模块（HSM）、虚拟 HSM 或带/不带 HSM 组件的软件密钥管理器。它可以位于外部的任何地方 Amazon，包括您的本地、本地或远程数据中心或任何云中。您的外部密钥存储可以由单个外部密钥管理器提供支持，也可以由共享加密密钥的多个相关密钥管理器实例（例如 HSM 集群）提供支持。外部密钥存储旨在支持来自不同供应商的各种外部管理器。有关连接外部密钥管理器的详细信息，请参阅 选择外部密钥存储代理连接选项。

外部密钥

外部密钥存储中的每个 KMS 密钥都与外部密钥管理器中的加密密钥（称为外部密钥）相关联。在使用外部密钥存储中的 KMS 密钥加密或解密时，将使用外部密钥在外部密钥管理器中执行加密操作。

> **Warning**
>
> 外部密钥对 KMS 密钥的操作至关重要。如果外部密钥丢失或遭删除，则以相关 KMS 密钥加密的加密文字将无法恢复。

使用外部密钥存储时，外部密钥必须是已启用且可以执行加密和解密的 256 位 AES 密钥。有关详细的外部密钥要求，请参阅 外部密钥存储中 KMS 密钥的要求。

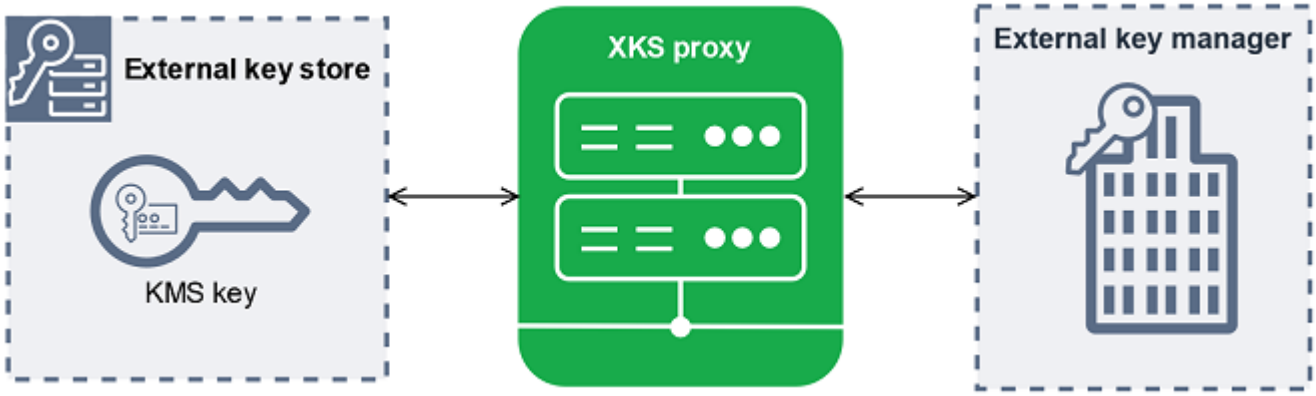
Amazon KMS 无法创建、删除或管理任何外部密钥。您的加密密钥材料永远不会离开外部密钥管理器。在外部密钥存储中创建 KMS 密钥时，您需要提供外部密钥的 ID（`XksKeyId`）。您无法更改与 KMS 密钥关联的外部密钥 ID，即便外部密钥管理器可以轮换与外部密钥 ID 关联的密钥材料。

除了外部密钥，外部密钥存储中的 KMS 密钥还具有 Amazon KMS 密钥材料。受 KMS 密钥保护的数据首先 Amazon KMS 使用密 Amazon KMS 钥材料进行加密，然后由您的外部密钥管理器使用您的外部密钥进行加密。这种双重加密过程可确保受您 KMS 密钥保护的加密文字始终至少与仅受 Amazon KMS保护的加密文字一样强大。

许多加密密钥具有不同类型的标识符。在外部密钥存储中创建 KMS 密钥时，您要提供外部密钥存储代理用来引用外部密钥的外部密钥 ID。如果使用了错误的标识符，尝试在外部密钥存储中创建 KMS 密钥会失败。

外部密钥存储代理

外部密钥存储代理（“XKS 代理”）是客户拥有和客户管理的软件应用程序，用于调解与您的外部密钥管理器 Amazon KMS 之间的所有通信。它还将通用 Amazon KMS 请求转换为供应商特定的外部密钥管理器可以理解的格式。外部密钥存储需要外部密钥存储代理。每个外部密钥存储会关联一个外部密钥存储代理。



Amazon KMS 无法创建、删除或管理任何外部密钥。您的加密密钥材料永远不会离开外部密钥管理器。 Amazon KMS 与您的外部密钥管理器之间的所有通信均由您的外部密钥存储代理进行中介。Amazon KMS 向外部密钥存储代理发送请求并接收来自外部密钥存储代理的响应。外部密钥存储代理负责将请求从您的外部密钥管理器传输 Amazon KMS 到您的外部密钥管理器，并将来自外部密钥管理器的响应传回到 Amazon KMS

您拥有并管理外部密钥存储的外部密钥存储代理，并负责其维护和操作。您可以根据开源外部密钥存储代理 API 规范开发外部密钥存储代理 API 规范，该规范 Amazon KMS 发布或从供应商那里购买代理应用程序。您的外部密钥存储代理可能包含在您的外部密钥管理器中。为了支持代理开发， Amazon KMS 还提供了外部密钥存储代理示例 (aws-kms-xks-proxy) 和测试客户端 (xks-kms-xksproxy-test-client)，用于验证您的外部密钥存储代理是否符合规范。

要进行身份验证 Amazon KMS，代理使用服务器端 TLS 证书。要向您的代理进行身份验证， Amazon KMS 请使用 Sigv4 代理身份验证凭据对外部密钥存储代理的所有请求进行签名。或者，您的代理可以启用双向 TLS (mTLS)，以进一步保证它只接受来自 Amazon KMS的请求。

您的外部密钥存储代理必须支持 HTTP/1.1 或更高版本以及 TLS 1.2 或更高版本，并至少使用以下密码套件之一：

- TLS_AES_256_GCM SHA384 _ (TLS 1.3)
- TLS_ CHACHA2 0_ 0 POLY13 5 SHA256 _ (TLS 1.3)

> **Note**
>
> Amazon GovCloud (US) Region 不支持 TLS_ CHACHA2 0_ 05 POLY13 _。SHA256

- TLS_ECDHE_RSA_WITH SHA384 _AES_256_GCM_ (TLS 1.2)
- TLS_ECDHE_ECDSA_WITH SHA384 _AES_256_GCM_ (TLS 1.2)

要在外部密钥存储中创建和使用 KMS 密钥，必须先将外部密钥存储连接到其外部密钥存储代理。您也可以根据需要断开外部密钥存储与其代理的连接。在这样做时，外部密钥存储中的所有 KMS 密钥都会变得不可用，也不能用于任何加密操作。

外部密钥存储代理连接

外部密钥存储代理连接（“XKS 代理连接”）描述了 Amazon KMS 用于与外部密钥存储代理进行通信的方法。

您在创建外部密钥存储时指定代理连接选项，该选项将成为外部密钥存储的属性。您可以通过更新自定义密钥存储属性来更改代理连接选项，但必须确定外部密钥存储代理仍然可以访问相同的外部密钥。

Amazon KMS 支持以下连接选项。

- 公共终端节点连接-通过 Internet 将外部密钥存储代理的请求 Amazon KMS 发送到您控制的公共终端节点。此选项易于创建和维护，但可能无法满足每次安装的安全要求。
- VPC 终端节点服务连接 — Amazon KMS 向您创建和维护的亚马逊虚拟私有云（亚马逊 VPC）终端节点服务发送请求。您可以在 Amazon VPC 内托管外部密钥存储代理，也可以将外部密钥存储代理托管在 Amazon VPC 之外 Amazon 并仅用于通信。此外还可以将外部密钥存储连接到由其他 Amazon Web Services 账户拥有的 Amazon VPC 端点服务。

有关外部密钥存储代理连接选项的详细信息，请参阅 选择外部密钥存储代理连接选项。

外部密钥存储代理身份验证凭证

要向外部密钥存储代理进行身份验证， Amazon KMS 请使用签名 V4 (Sigv4) 身份验证凭据对外部密钥存储代理的所有请求进行签名。您在代理上建立并维护身份验证凭据，然后在创建外部存储 Amazon KMS 时提供此凭据。

> **Note**
>
> Amazon KMS 用于签署 XKS 代理请求的 Sigv4 凭据与您的中的委托人关联的任何 Sigv4 凭据无关。 Amazon Identity and Access Management Amazon Web Services 账户不要将任何 IAM SigV4 凭证重复用于外部密钥存储代理。

每个代理身份验证凭证有两个部分。在创建外部密钥存储或更新外部密钥存储的身份验证凭证时，必须提供这两部分。

- 访问密钥 ID：标识秘密访问密钥。您能以明文形式提供此 ID。
- 秘密访问密钥：凭证的秘密部分。 Amazon KMS 在存储凭据之前对凭据中的私有访问密钥进行加密。

您随时可以编辑凭证设置，例如输入了错误值时、在代理上更改凭证时或者代理轮换证书时。有关对外部密钥存储代理 Amazon KMS 进行身份验证的技术细节，请参阅《 Amazon KMS 外部密钥存储代理 API 规范》中的身份验证。

为了允许您在不中断外部密钥存储中使用 KMS 密钥的凭证的情况下轮换凭证，我们建议外部密钥存储代理至少支持两个有效的身份验证凭据。 Amazon Web Services 服务 Amazon KMS这样可以确保在您向 Amazon KMS提供新凭证时，以前的凭证继续有效。

为了帮助您跟踪代理身份验证凭证的使用年限， Amazon KMS 定义了 Amazon CloudWatch 指标。XksProxyCredentialAge您可以使用此指标创建 CloudWatch 警报，当您的凭证有效期达到您设定的阈值时通知您。

为了进一步确保您的外部密钥存储代理仅响应 Amazon KMS，一些外部密钥代理支持相互传输层安全性协议（mTLS）。有关更多信息，请参阅 mTLS 身份验证（可选）。

代理 APIs

要支持 Amazon KMS 外部密钥存储，外部密钥存储代理必须实现所需的代理， APIs 如Amazon KMS 外部密钥存储代理 API 规范中所述。这些代理 API 请求是 Amazon KMS 发送到代理的唯一请求。即便您从不直接发送这些请求，了解这些请求也可能有助于您修复外部密钥存储或其代理可能出现的任何问题。例如，在其外部密钥存储的 Amazon CloudWatch 指标中 Amazon KMS 包含有关这些 API 调用的延迟和成功率的信息。有关更多信息，请参阅 监控外部密钥存储。

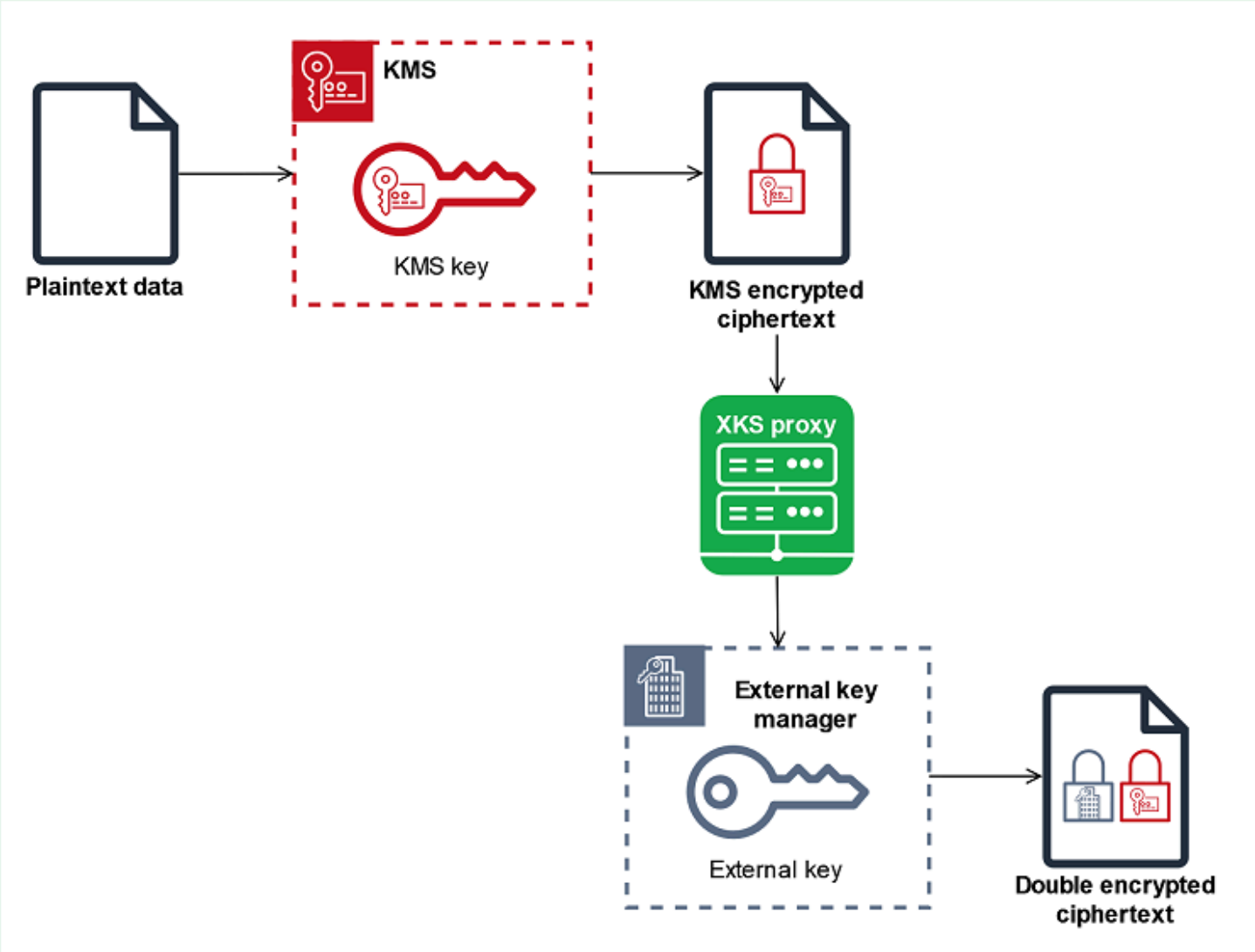
下表列出并描述了每个代理 APIs。它还包括触发对 Amazon KMS 代理 API 的调用的 Amazon KMS 操作以及与代理 API 相关的任何操作异常。

代理 API	描述	相关 Amazon KMS 操作
Decrypt	Amazon KMS 发送要解密的密文以及要使用的外部密钥的 ID。所需的加密算法为 AES_GCM。	解密，ReEncrypt
Encrypt	Amazon KMS 发送要加密的数据以及要使用的外部密钥的 ID。所需的加密算法为 AES_GCM。	加密、GenerateDataKey、GenerateDataKeyWithoutPlaintext、ReEncrypt
GetHealthStatus	Amazon KMS 请求有关代理和您的外部密钥管理器状态的信息。 每个外部密钥管理器的状态可以是以下状态之一。 • `Active`：正常，可以传输流量 • `Degraded`：不正常，但可以传输流量 • `Unavailable`：不正常，不可以传输流量	CreateCustomKeyStore（用于公共终端节点连接）、ConnectCustomKeyStore（用于 VPC 终端节点服务连接） 如果所有外部密钥管理器实例都处于 `Unavailable` 状态，则尝试创建或连接密钥存储将失败并显示 `XksProxyUriUnreachableException`。
GetKeyMetadata	Amazon KMS 请求有关与您的外部密钥存储库中的 KMS 密钥关联的外部密钥的信息。 响应内容包含密钥规范（`AES_256`）、密钥用法（`[ENCRYPT, DECRYPT]`），以及外部密钥处于 `ENABLED` 还是 `DISABLED` 状态。	CreateKey 如果密钥规范不是 `AES_256`，或者密钥用法不是 `[ENCRYPT, DECRYPT]`，或者状态为 `DISABLED`，则 `CreateKey` 操作将失败并显示 `XksKeyInvalidConfigurationException`。

双重加密

由外部密钥存储中的 KMS 密钥加密的数据经过两次加密。首先，Amazon KMS 使用特定于 KMS Amazon KMS 密钥的密钥材料对数据进行加密。然后，外部密钥管理器使用外部密钥加密经过 Amazon KMS加密的加密文字。此过程称为双重加密。

双重加密可确保外部密钥存储中经 KMS 密钥加密的数据至少与经标准 KMS 密钥加密的加密文字一样强大。它还可以保护您从外部密钥存储代理传输 Amazon KMS 的纯文本。您可以借助双重加密保留对加密文字的完全控制。如果您通过外部代理永久撤消 Amazon 对外部密钥的访问权限，Amazon 中剩余的任何加密文字都会受到有效的加密粉碎处理。



要启用双重加密，外部密钥存储中的每个 KMS 密钥都有两个加密备用密钥：

- KMS Amazon KMS 密钥独有的密钥材料。此密钥材料是生成的，仅用于 Amazon KMS FIPS 140-3 安全等级 3 认证的硬件安全模块 ()。HSMs
- 外部密钥管理器中的外部密钥。

双重加密具有以下效果：

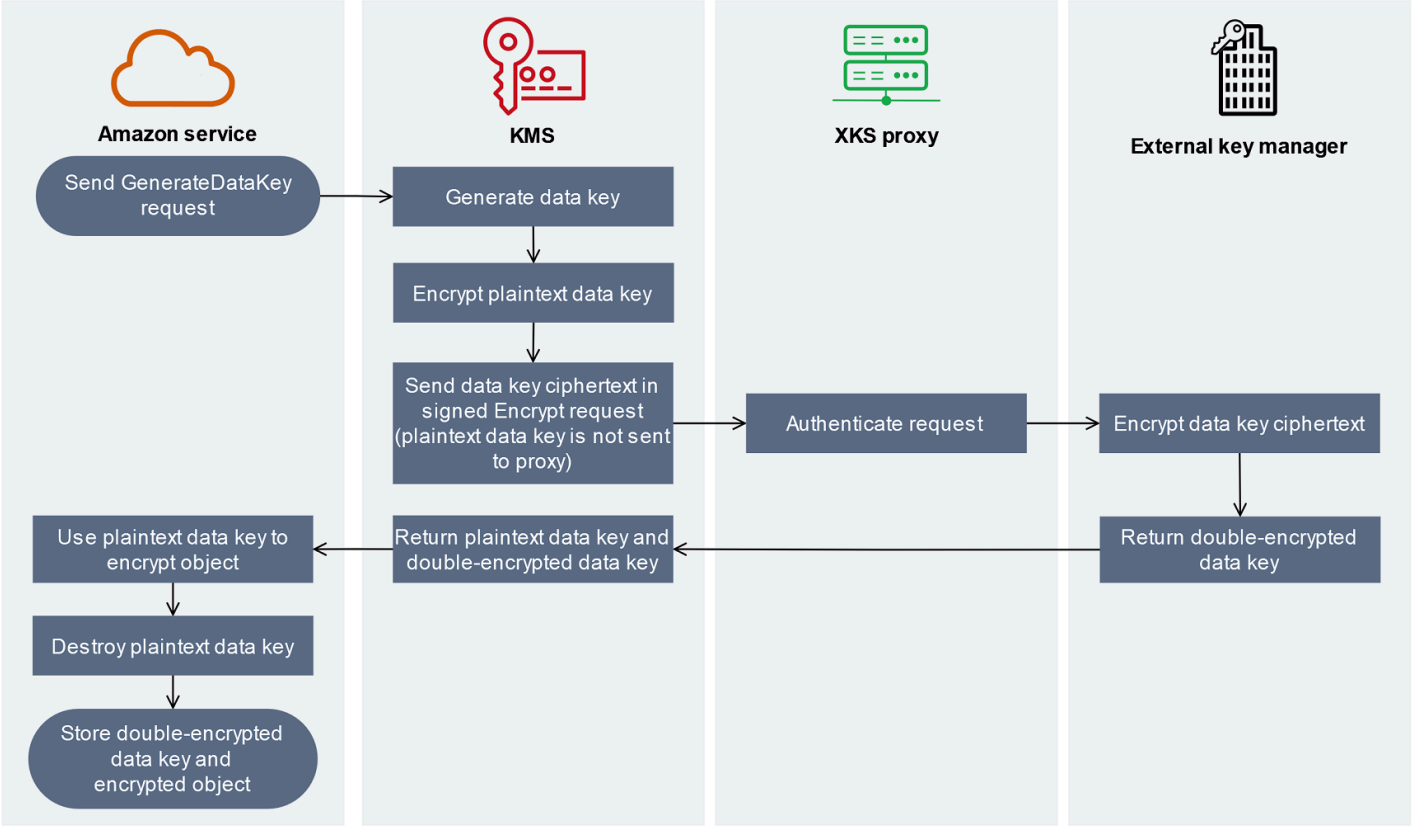
- Amazon KMS 如果不通过外部密钥存储代理访问您的外部密钥，则无法解密外部密钥存储库中由 KMS 密钥加密的任何密文。
- 即使您有外部密钥材料，也无法解密外部密钥存储库中由 KMS 密钥加密的任何密文。 Amazon
- 即使您拥有外部密钥材料，也无法重新创建已从外部密钥存储中删除的 KMS 密钥。每个 KMS 密钥都有包含在对称加密文字中的唯一元数据。新的 KMS 密钥将无法解密由原始密钥加密的加密文字，即使其使用相同的外部密钥材料也是如此。

有关实际应用中的双重加密示例，请参见 外部密钥存储的工作原理。

外部密钥存储的工作原理

外部密钥存储、外部密钥存储代理和外部密钥管理器会协同保护您的 Amazon 资源。以下过程描述了典型 Amazon Web Services 服务 的加密工作流程，即使用由 KMS 密钥保护的唯一数据密钥对每个对象进行加密。在本例中，您选择了外部密钥存储中的 KMS 密钥来保护对象。该示例说明了如何 Amazon KMS 使用双重加密来保护传输中的数据密钥，并确保外部密钥存储库中的 KMS 密钥生成的密文始终至少与由标准对称 KMS 密钥加密并包含密钥材料的密文一样强大。 Amazon KMS

与之集成的每个实际 Amazon Web Services 服务 使用的加密方法各 Amazon KMS 不相同。有关详细信息，请参阅 Amazon Web Services 服务 文档“安全”章节中的“数据保护”主题。



1. 您向 Amazon Web Services 服务 资源中添加了一个新对象。要加密对象，请 Amazon KMS 使用外部密钥存储库中的 KMS 密钥 Amazon Web Services 服务 向发送GenerateDataKey请求。
2. Amazon KMS 生成 256 位对称数据密钥，并准备通过外部密钥存储代理将纯文本数据密钥的副本发送到外部密钥管理器。 Amazon KMS 使用与外部密钥存储库中的 KMS 密钥关联的密钥材料对纯文本数据Amazon KMS 密钥进行加密，从而开始双重加密过程。
3. Amazon KMS 向与外部密钥存储关联的外部密钥存储代理发送加密请求。该请求包括要加密的数据密钥密文以及与 KMS 密钥关联的外部密钥的 ID。 Amazon KMS 使用外部密钥存储代理的代理身份验证凭据对请求进行签名。

   数据密钥的明文副本不会发送到外部密钥存储代理。
4. 外部密钥存储代理对请求进行身份验证，然后将加密请求传递给您的外部密钥管理器。

   一些外部密钥存储代理还实现了可选的授权策略，该策略仅允许选定的主体在特定条件下执行操作。
5. 您的外部密钥管理器使用指定的外部密钥对数据密钥加密文字进行加密。外部密钥管理器将经过双重加密的数据密钥返回给外部密钥存储代理，后者再将其返回给 Amazon KMS。

6. Amazon KMS 将纯文本数据密钥和该数据密钥的双重加密副本返回到。 Amazon Web Services 服务
7. Amazon Web Services 服务 使用纯文本数据密钥对资源对象进行加密，销毁纯文本数据密钥，并将加密的数据密钥与加密对象一起存储。

   有些人 Amazon Web Services 服务 可能会缓存纯文本数据密钥以用于多个对象，或者在使用资源时重复使用。有关更多信息，请参阅 不可用的 KMS 密钥如何影响数据密钥。

要解密加密对象， Amazon Web Services 服务 必须在 Decrypt 请求 Amazon KMS 中将加密的数据密钥发回给。要解密加密的数据密钥， Amazon KMS 必须使用外部密钥的 ID 将加密的数据密钥发送回您的外部密钥存储代理服务器。如果对外部密钥存储代理的解密请求因任何原因失败，则 Amazon KMS 无法解密加密的数据密钥， Amazon Web Services 服务 也无法解密加密对象。

控制对外部密钥存储的访问

与标准 KMS 密钥配合使用的所有 Amazon KMS 访问控制功能（密钥策略、IAM 策略和授权）对外部密钥存储中的 KMS 密钥的作用相同。您可以使用 IAM policy 来控制对创建和管理外部密钥存储的 API 操作的访问。您可以使用 IAM 策略和密钥策略来控制对外部密钥存储 Amazon KMS keys 中的的访问权限。您还可以使用适用于您的 Amazon 组织的服务控制策略和 VPC 终端节点策略来控制对外部密钥存储中的 KMS 密钥的访问。

我们建议您仅向用户和角色提供他们可能执行的任务所需的权限。

主题

- 授权外部密钥存储管理器
- 授权外部密钥存储中 KMS 密钥的用户
- 授权 Amazon KMS 与您的外部密钥存储代理进行通信
- 外部密钥存储代理授权（可选）
- mTLS 身份验证（可选）

授权外部密钥存储管理器

创建和管理外部密钥存储的主体需要自定义密钥存储操作的权限。以下列表描述了外部密钥存储管理器所需的最低权限。由于自定义密钥存储不是 Amazon 资源，因此您无法为其他 Amazon Web Services 账户密钥库中的委托人提供对外部密钥存储的权限。

- `kms:CreateCustomKeyStore`

- kms:DescribeCustomKeyStores
- kms:ConnectCustomKeyStore
- kms:DisconnectCustomKeyStore
- kms:UpdateCustomKeyStore
- kms:DeleteCustomKeyStore

要创建具有 Amazon VPC 终端节点服务连接的外部密钥存储库，并且 VPC 终端节点服务归另一方所有 Amazon Web Services 账户，您还需要以下权限：

- ec2:DescribeVPCEndpointServices

创建外部密钥存储的主体需要权限来创建和配置外部密钥存储组件。主体只能在自己的账户中创建外部密钥存储。要创建具有 VPC 端点服务连接的外部密钥存储，主体必须具有创建以下组件的权限：

- Amazon VPC
- 公有子网和私有子网
- 网络负载均衡器和目标组
- Amazon VPC 端点服务

有关详细信息，请参阅 Amazon VPC 的身份和访问管理、VPC 端点和 VPC 端点服务的身份和访问管理以及 Elastic Load Balancing API 权限。

授权外部密钥存储中 KMS 密钥的用户

Amazon KMS keys 在您的外部密钥存储中创建和管理的委托人需要与在中创建和管理任何 KMS 密钥的委托人相同的权限。 Amazon KMS外部密钥存储中的 KMS 密钥的默认密钥策略与 Amazon KMS中 KMS 密钥的默认密钥策略相同。基于属性的访问权限控制（ABAC）使用标签和别名来控制对 KMS 密钥的访问，对外部密钥存储中的 KMS 密钥也有效。

使用自定义密钥存储中的 KMS 密钥进行加密操作的委托人需要使用 KMS 密钥执行加密操作的权限，如 kms:Decrypt。您可以在 IAM 或密钥策略中提供这些权限。但是，他们无需任何额外权限即可在自定义密钥存储中使用 KMS 密钥。

要设置仅适用于外部密钥存储中 KMS 密钥的权限，请使用值为 EXTERNAL_KEY_STORE 的 kms:KeyOrigin 策略条件。您可以使用此条件来限制 kms: CreateKey 权限或任何特定于 KMS 密钥资源的权限。例如，以下 IAM policy 允许其所连接的身份对账户中的所有 KMS 密钥调用指定操作，只

要这些 KMS 密钥位于外部密钥存储中。请注意，您可以将权限限制为外部密钥存储中的 KMS 密钥和账户中的 KMS 密钥 Amazon Web Services 账户，但不能限制账户中任何特定的外部密钥存储区。

```
{
  "Sid": "AllowKeysInExternalKeyStores",
  "Effect": "Allow",
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "arn:aws:kms:us-west-2:111122223333:key/*",
  "Condition": {
    "StringEquals": {
      "kms:KeyOrigin": "EXTERNAL_KEY_STORE"
    }
  }
}
```

授权 Amazon KMS 与您的外部密钥存储代理进行通信

Amazon KMS 仅通过您提供的外部密钥存储代理与您的外部密钥管理器通信。 Amazon KMS 使用签名版本 4 (Sigv4) 进程使用您指定的外部密钥存储代理身份验证凭据对代理的请求进行签名，从而对代理进行身份验证。如果您为外部密钥存储代理使用公共端点连接， Amazon KMS 则不需要任何其他权限。

但是，如果您使用的是 VPC 终端节点服务连接，则必须 Amazon KMS 授予权限才能为您的 Amazon VPC 终端节点服务创建接口终端节点。无论外部密钥存储代理位于您的 VPC 中，还是外部密钥存储代理位于其他地方，但使用 VPC 终端节点服务与之通信，都需要此权限 Amazon KMS。

Amazon KMS 要允许创建接口终端节点，请使用 Amazon VPC 控制台或ModifyVpcEndpointServicePermissions操作。允许以下主体的权限：`cks.kms.<region>.amazonaws.com`。

如果您的 Amazon VPC 终端节点服务由拥有外部密钥存储库 (XKS) 以外 Amazon Web Services 账户的其他人所有，则还需要允许 XKS 访问该 VPC 终端节点服务。 Amazon Web Services 账户 为此，请将 XKS Amazon Web Services 账户 ID 列为亚马逊 VPC 终端节点服务的委托人。

Same Amazon Web Services 账户

如果您的 VPC 终端节点服务归您的外部密钥存储库所有，则必须将其添加 Amazon KMS 到 VPC 终端节点服务的允许委托人列表中。 Amazon Web Services 账户

以下示例使用modify-vpc-endpoint-service-permissions Amazon CLI 命令允许 Amazon KMS 连接到美国西部（俄勒冈）(us-west-2) 区域中的指定 VPC 终端节点服务。在使用此命令之前，请将 Amazon VPC 服务 ID 和 Amazon Web Services 区域 替换为适用于您的配置的有效值。

```
modify-vpc-endpoint-service-permissions
--service-id vpce-svc-12abc34567def0987
--add-allowed-principals '["cks.kms.us-west-2.amazonaws.com"]'
```

要移除此权限，请使用 Amazon VPC 控制台或ModifyVpcEndpointServicePermissions带RemoveAllowedPrincipals参数的。

Cross Amazon Web Services 账户

如果您的 VPC 终端节点服务归另一方所有 Amazon Web Services 账户，则必须将两者 Amazon KMS 以及您的外部密钥存储库添加到 VPC 终端节点服务的允许委托人列表中。

以下示例使用modify-vpc-endpoint-service-permissions Amazon CLI 命令允许两者 Amazon KMS 以及您的外部密钥存储 (XKS) 连接到美国西部（俄勒冈）(us-west-2) 区域中的指定 VPC 终端节点服务。请首先将 Amazon VPC 服务 ID、 Amazon Web Services 区域和 IAM 主体 ARN 替换为适用于您的配置的有效值，然后再使用此命令。应将 IAM 委托人替换为 XKS 所有者 Amazon Web Services 账户中的委托人。

在此示例中，arn:aws:iam::*123456789012*:role/*cks_role* 是 XKS 所有者账户中的一个 IAM 主体，将用于创建、更新 XKS 或将其连接到您的 VPC 端点服务。如果要允许 XKS 所有者账户中的所有主体访问您的 VPC 端点服务，则可以指定 arn:aws:iam::*123456789012*:root。

```
modify-vpc-endpoint-service-permissions
--service-id vpce-svc-12abc34567def0987
--add-allowed-principals '["cks.kms.us-west-2.amazonaws.com",
 "arn:aws:iam::123456789012:role/cks_role"]'
```

要移除此权限，请使用 Amazon VPC 控制台或ModifyVpcEndpointServicePermissions带RemoveAllowedPrincipals参数的。

外部密钥存储代理授权（可选）

一些外部密钥存储代理针对其外部密钥的使用执行授权要求。允许但不要求使用外部密钥存储代理来设计和实现授权方案，该方案允许特定用户仅在特定条件下请求特定操作。例如，可以将代理配置为允许用户 A 使用特定的外部密钥进行加密，但不允许使用该外部密钥进行解密。

代理授权独立于基于 SIGV4 的代理身份验证，后者 Amazon KMS 需要所有外部密钥存储代理。其还独立于授权访问影响外部密钥存储或其 KMS 密钥的操作的密钥策略、IAM policy 和授权。

要启用外部密钥存储代理的授权，请在每个代理 API 请求中 Amazon KMS 包含元数据，包括调用者、KMS 密钥、 Amazon KMS 操作、 Amazon Web Services 服务 （如果有）。外部密钥代理 API 版本 1（v1）的请求元数据如下所示。

```
"requestMetadata": {
    "awsPrincipalArn": string,
    "awsSourceVpc": string, // optional
    "awsSourceVpce": string, // optional
    "kmsKeyArn": string,
    "kmsOperation": string,
    "kmsRequestId": string,
    "kmsViaService": string // optional
}
```

例如，您可以将代理配置为允许来自特定委托人 (`awsPrincipalArn`) 的请求，但前提是请求由特定 Amazon Web Services 服务 (`kmsViaService`) 代表委托人提出。

如果代理授权失败，则相关 Amazon KMS 操作将失败，并显示一条解释错误的消息。有关详细信息，请参阅 代理授权问题。

mTLS 身份验证（可选）

要使您的外部密钥存储代理能够对来自的请求进行身份验证 Amazon KMS，请使用外部密钥存储的 S Amazon KMS ignature V4 (Sigv4) 代理身份验证凭据对外部密钥存储代理的所有请求进行签名。

为了进一步保证您的外部密钥存储代理仅响应 Amazon KMS 请求，一些外部密钥代理支持相互传输层安全 (mTLS)，即交易双方使用证书进行身份验证。mTLS 在标准 TLS 提供的服务器端身份验证中添加了客户端身份验证（即外部密钥存储代理服务器对 Amazon KMS 客户端进行身份验证）。在极少数情况下，代理身份验证凭证会遭到泄露，mTLS 会阻止第三方成功向外部密钥存储代理发出 API 请求。

要实现 mTLS，请将您的外部密钥存储代理配置为仅接受具有以下属性的客户端 TLS 证书：

- TLS 证书上的主题通用名称必须是 `cks.kms.<Region>.amazonaws.com`，例如 `cks.kms.eu-west-3.amazonaws.com`。
- 证书必须链接到与 Amazon Trust Services 关联的证书颁发机构。

选择外部密钥存储代理连接选项

在创建外部密钥库之前，请选择决定与外部密钥库组件 Amazon KMS 通信方式的连接选项。您选择的连接选项决定了规划过程的剩余步骤。

如果要创建外部密钥存储库，则需要确定 Amazon KMS 与外部密钥存储代理的通信方式。此选项将决定您需要哪些组件以及如何配置它们。 Amazon KMS 支持以下连接选项。

- 公有端点连接
- VPC 端点服务连接

选择能满足性能和安全目标的选项。

在开始之前，请确认您需要外部密钥存储。大多数客户都可以使用由密钥材料支持的 KMS Amazon KMS 密钥。

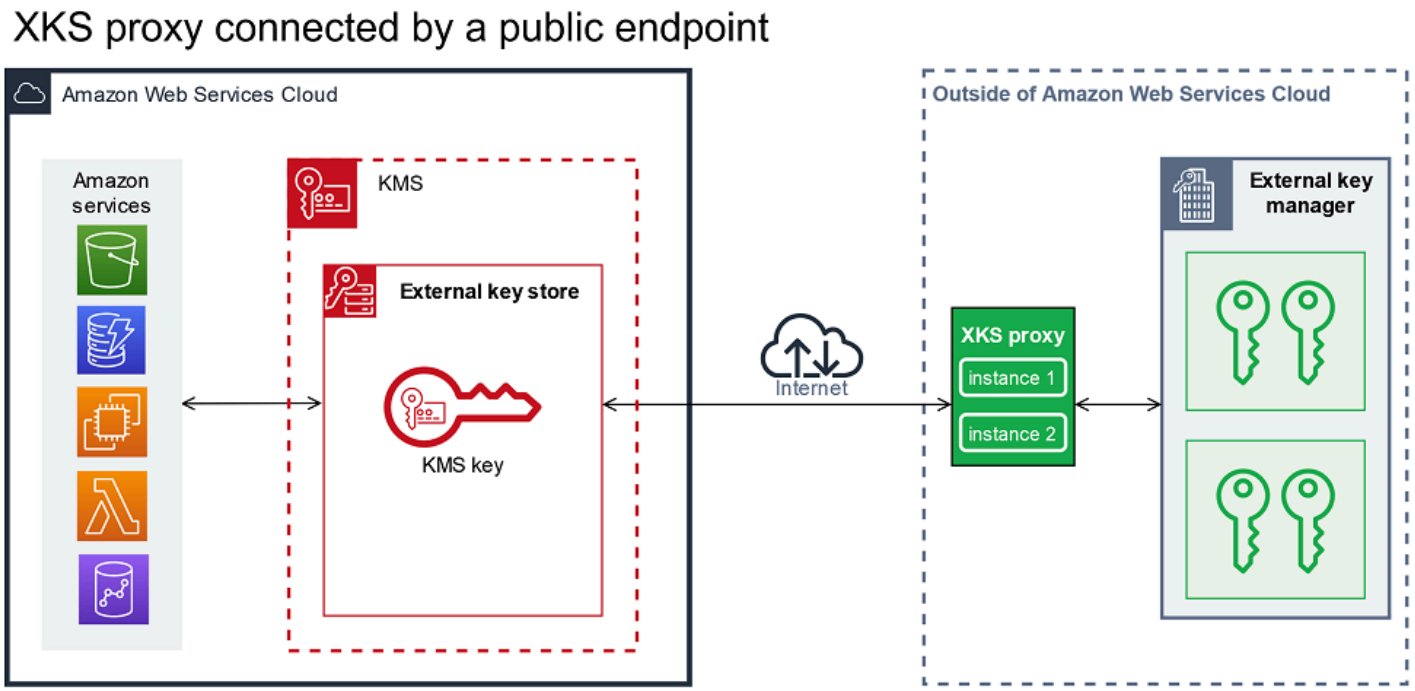
注意事项

- 如果您的外部密钥存储代理构建到外部密钥管理器中，则您的连接可能是预先确定的。有关详细信息，请参阅外部密钥管理器或外部密钥存储代理的文档。
- 即使在正在运行的外部密钥存储上，您也可以更改外部密钥存储代理的连接选项。但是，必须仔细规划和执行该过程，以尽可能减少中断和避免错误，并确保继续访问加密数据的加密密钥。

公有端点连接

Amazon KMS 使用公共端点通过互联网连接到外部密钥存储代理（XKS 代理）。

此连接选项更易于设置和维护，并且可以与某些密钥管理模型很好地匹配。但是，此选项可能无法满足某些组织的安全要求。



要求

如果您选择公有端点连接，则需要满足以下条件。

- 您的外部密钥存储代理必须可以在可公开路由的端点上访问。
- 您可以将同一个公有端点用于多个外部密钥存储，前提是这些外部密钥存储使用不同的代理 URI 路径值。
- 对于具有公共终端节点连接的外部密钥存储库，以及任何具有 VPC 终端节点服务连接的外部密钥存储库 Amazon Web Services 区域，即使密钥存储库位于不同的位置，您也不能使用相同的终端节点 Amazon Web Services 账户。
- 您必须获得由外部密钥存储支持的公有证书颁发机构颁发的 TLS 证书。有关列表，请参阅 Trusted Certificate Authorities（受信任的证书颁发机构）。

  TLS 证书上的主题公用名（CN）必须与外部密钥存储代理的代理 URI 端点中的域名相匹配。例如，如果公有端点是 `https://myproxy.xks.example.com`，即 TLS，则 TLS 证书上的 CN 必须为 `myproxy.xks.example.com` 或 `*.xks.example.com`。
- 确保与外部密钥存储代理 Amazon KMS 之间的任何防火墙都允许进出代理上端口 443 的流量。Amazon KMS 通过端口 443 进行通信。 IPv4此值不可配置。

有关外部密钥存储的所有要求，请参阅 Assemble the prerequisites（汇编先决条件）。

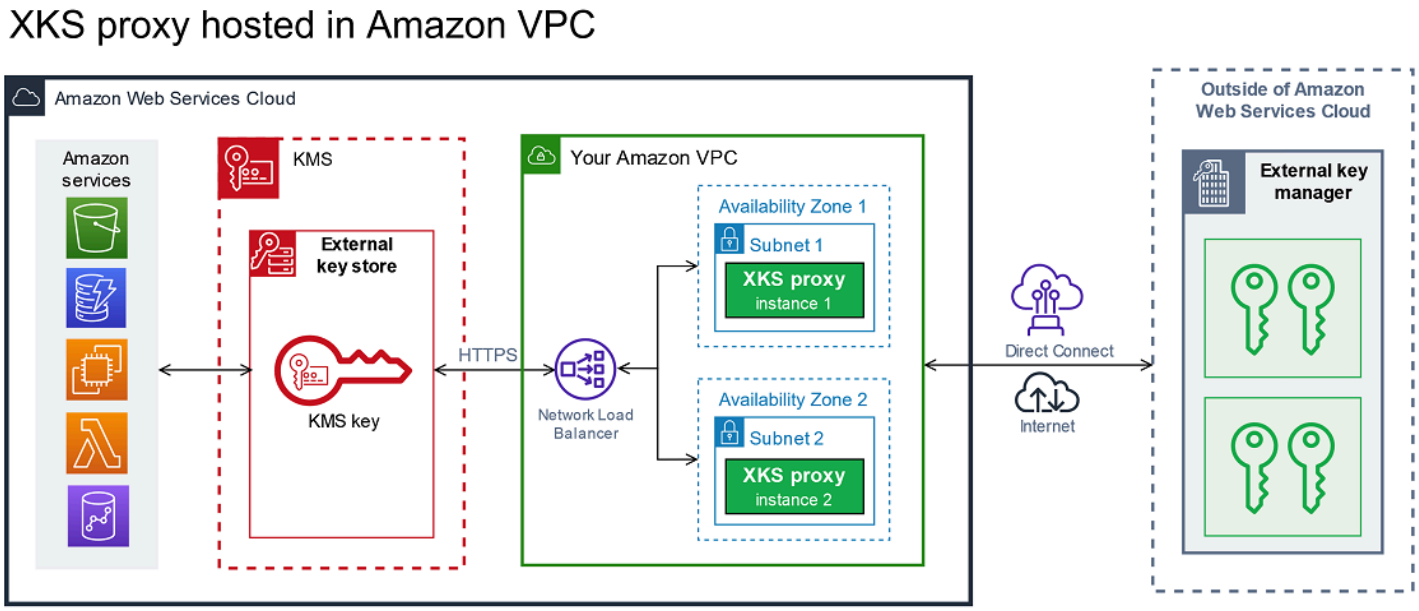
VPC 端点服务连接

Amazon KMS 通过创建与您创建和配置的 Amazon VPC 终端节点服务的接口终端节点来连接到外部密钥存储代理（XKS 代理）。您负责创建 VPC 端点服务并将您的 VPC 连接到外部密钥管理器。

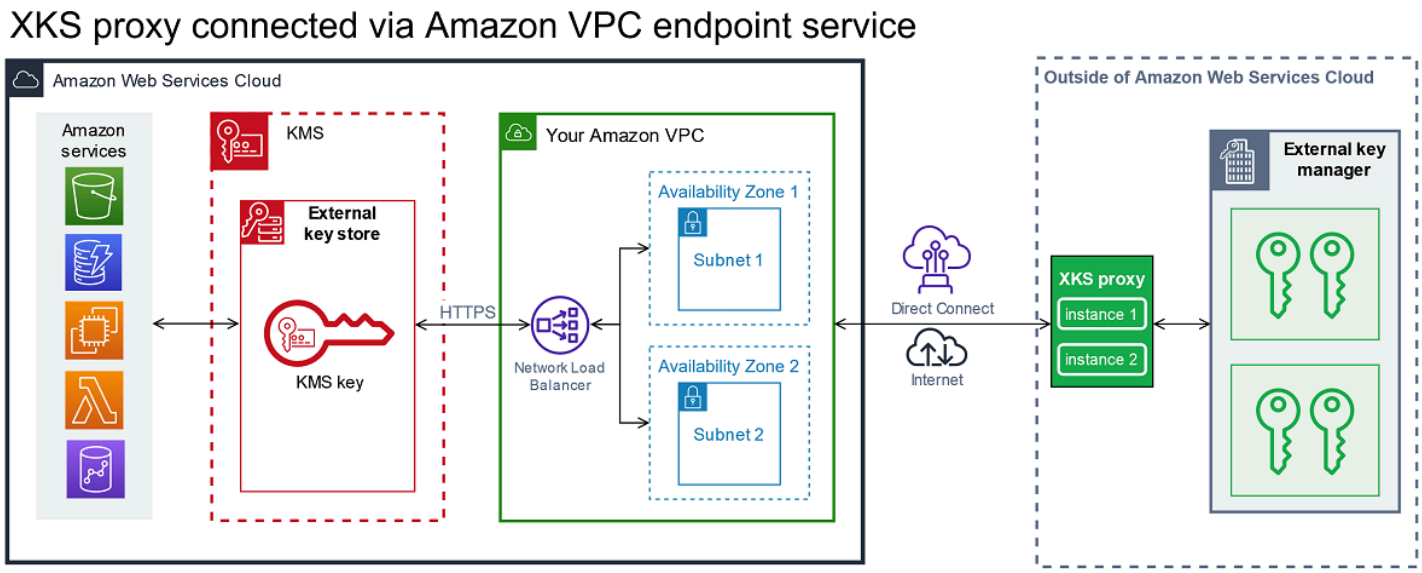
您的终端节点服务可以使用任何支持的 network-to-Amazon VPC 选项进行通信，包括Amazon Direct Connect。

此连接选项的设置和维护更为复杂。但是它使用了 Amazon PrivateLink，这样就可以在不使用公共互联网的情况下私下 Amazon KMS 连接到您的 Amazon VPC 和外部密钥存储代理。

您可以在您的 Amazon VPC 中找到外部密钥存储代理。



或者，您可以将外部密钥存储代理放在外部， Amazon Web Services 云 并仅使用您的 Amazon VPC 终端节点服务进行安全通信 Amazon KMS。



您也可以将外部密钥存储连接到另一个密钥存储库拥有的 Amazon VPC 终端节点服务 Amazon Web Services 账户。两者都 Amazon Web Services 账户 需要必要的权限才能允许与 VPC 终端节点服务 Amazon KMS 之间的通信

了解更多：

- 查看创建外部密钥存储的过程，包括汇编先决条件。这将帮助您确保在创建外部密钥存储时拥有所需的所有组件。
- 了解如何控制对外部密钥存储的访问，包括外部密钥存储管理员和用户所需的权限。
- 了解 Amazon KMS 记录外部密钥存储的 Amazon CloudWatch 指标和维度。我们强烈建议您创建警报来监控外部密钥存储，以便您就可以检测出性能和操作问题的早期迹象。

配置 VPC 端点服务连接

使用本节中的指南创建和配置使用 VPC 终端节点服务连接的外部密钥存储所需的 Amazon 资源和相关组件。此连接选项列出的资源是对所有外部密钥存储所需资源的补充。创建和配置所需资源后，您可以创建外部密钥存储。

您可以在 Amazon VPC 中找到您的外部密钥存储代理，也可以在外部找到代理 Amazon 并使用您的 VPC 终端节点服务进行通信。

在开始之前，请确认您需要外部密钥存储。大多数客户都可以使用由密钥材料支持的 KMS Amazon KMS 密钥。

> **Note**
>
> VPC 端点服务连接所需的某些元素可能包含在您的外部密钥管理器中。此外，您的软件可能有其他配置要求。在创建和配置本节中的 Amazon 资源之前，请查阅代理和密钥管理器文档。

主题

VPC 端点服务连接的要求

如果您为外部密钥存储选择 VPC 端点服务连接，则需要以下资源。

- 连接到外部密钥管理器的 Amazon VPC。它必须在两个不同的可用区中至少拥有两个私有子网。

  您可以将现有 Amazon VPC 用于外部密钥存储，前提是此 VPC 符合与外部密钥存储一起使用的要求。多个外部密钥存储可以共享一个 Amazon VPC，但每个外部密钥存储必须有自己的 VPC 端点服务和私有 DNS 名称。
- 由 Amazon PrivateLink提供支持的 Amazon VPC 端点服务具有网络负载均衡器和目标组。

  端点服务不能要求接受。此外，您必须将 Amazon KMS 添加为允许的主体。这 Amazon KMS 允许创建接口端点，以便它可以与您的外部密钥存储代理进行通信。
- VPC 端点服务在其 Amazon Web Services 区域中的唯一私有 DNS 名称。

  私有 DNS 名称必须是更高级别的公有域的子域。例如，如果私有 DNS 名称为 `myproxy-private.xks.example.com`，则其必须是公有域（例如 `xks.example.com` 或 `example.com`）的子域。

  您必须针对私有 DNS 名称的 DNS 域验证所有权。
- 由受支持的公有证书颁发机构为您的外部密钥存储代理颁发的 TLS 证书。

TLS 证书上的主题通用名称（CN）必须与私有 DNS 名称相匹配。例如，若私有 DNS 名称为 `myproxy-private.xks.example.com`，则 TLS 证书上的 CN 必须为 `myproxy-private.xks.example.com` 或 `*.xks.example.com`。

- 为了最大限度地减少网络延迟，请在最接近外部密钥管理器的支持 Amazon Web Services 区域环境中创建 Amazon 组件。如果可行，请选择网络往返时间（RTT）不超过 35 毫秒的区域。

有关外部密钥存储的所有要求，请参阅 Assemble the prerequisites（汇编先决条件）。

步骤 1：创建 Amazon VPC 和子网

VPC 端点服务连接需要连接到外部密钥管理器的 Amazon VPC，该管理器至少拥有两个私有子网。您可以创建 Amazon VPC 或使用满足外部密钥存储要求的现有 Amazon VPC。有关创建新 VPC 的帮助，请参阅《Amazon Virtual Private Cloud 用户指南》中的创建 VPC。

Amazon VPC 的要求

Amazon VPC 端点服务必须具有以下属性才能使用外部密钥存储。

- 必须与您的外部密钥存储位于受支持的区域中。
- 需要至少两个私有子网，每个子网均位于不同的可用区内。
- Amazon VPC 的私有 IP 地址范围不得与托管外部密钥管理器的数据中心的私有 IP 地址范围重叠。
- 所有组件都必须使用 IPv4。

您可以通过多种方式将 Amazon VPC 连接到外部密钥存储代理。选择能满足性能和安全需求的选项。有关列表，请参阅将您的 VPC 连接到其他网络和 Network-to-Amazon VPC 连接选项。有关详细信息，请参阅 Amazon Direct Connect 和《Amazon Site-to-Site VPN 用户指南》https://docs.amazonaws.cn/vpn/latest/s2svpn/。

为外部密钥存储创建 Amazon VPC

使用下面的说明为外部密钥存储创建 Amazon VPC。仅当您选择 VPC 端点服务连接选项时，才需要 Amazon VPC。您可以使用满足外部密钥存储要求的现有 Amazon VPC。

按照创建 VPC、子网和其他 VPC 资源主题中的说明使用以下必需值。对于其他字段，请接受默认值并按要求提供名称。

Field	Value
IPv4 CIDR 块	输入 VPC 的 IP 地址。Amazon VPC 的私有 IP 地址范围不得与托管外部密钥管理器的数据中心的私有 IP 地址范围重叠。
可用区数量 (AZs)	2 或更多
公有子网的数量	不需要任何内容（0）
私有子网的数量	每个 AZ 一个
NAT 网关	不需要任何内容。
VPC 端点	不需要任何内容。
启用 DNS 主机名	是
启用 DNS 解析	是

请务必测试您的 VPC 通信。例如，如果您的外部密钥存储代理不在您的亚马逊 VPC 中，请在您的亚马逊 VPC 中创建一个亚马逊 EC2 实例，验证亚马逊 VPC 是否可以与您的外部密钥存储代理进行通信。

将 VPC 连接到外部密钥管理器

使用 Amazon VPC 支持的任何网络连接选项将 VPC 连接到托管外部密钥管理器的数据中心。确保 VPC 中的 Amazon EC2 实例（或外部密钥存储代理，如果在 VPC 中）可以与数据中心和外部密钥管理器通信。

步骤 2：创建目标组

在创建所需的 VPC 端点服务之前，创建其必需的组件、网络负载均衡器（NLB）和目标组。网络负载均衡器（NLB）在多个运行状况良好的目标之间分配请求，其中任何一个目标都可以为请求提供服务。在此步骤中，您将为外部密钥存储代理创建至少具有两台主机的目标组，并将您的 IP 地址注册到目标组。

使用以下必填值，按照配置目标组主题中的说明使用以下必需值。对于其他字段，请接受默认值并按要求提供名称。

Field	Value
Target type	IP 地址
协议	TCP
端口	443
IP 地址类型	IPv4
VPC	选择 VPC，您将在其中为外部密钥存储创建 VPC 端点服务。
运行状况检查协议和路径	运行状况检查协议和路径将与外部密钥存储代理配置不同。请参阅外部密钥管理器或外部密钥存储代理的文档。 若要为您的目标组配置运行状况检查，请参阅《网络负载均衡器的弹性负载均衡用户指南》中的目标组的运行状况检查。
Network	其他私有 IP 地址
IPv4 地址	外部密钥存储代理的私有地址
端口	443

步骤 3：创建网络负载均衡器

网络负载均衡器将网络流量（包括从 Amazon KMS 到外部密钥存储代理的请求）分配到配置的目标。

按照配置负载均衡器和侦听器主题中的说明配置和添加侦听器，并使用以下必需值创建负载均衡器。对于其他字段，请接受默认值并按要求提供名称。

Field	Value
Scheme	Internal
IP 地址类型	IPv4
网络映射	选择 VPC，您将在其中为外部密钥存储创建 VPC 端点服务。
Mapping	选择您为 VPC 子网配置的两个可用区（至少两个）。验证子网名称和私有 IP 地址。

Field	Value
协议	TCP
端口	443
默认操作：转发至	选择网络负载均衡器的目标组。

步骤 4：创建 VPC 端点服务

通常，您创建到服务的端点。但是，当您创建 VPC 终端节点服务时，您就是提供者，并为您的服务 Amazon KMS 创建终端节点。对于外部密钥存储，使用您在上一个步骤中创建的网络负载均衡器创建 VPC 端点服务。VPC 终端节点服务可以与您的外部密钥存储库 Amazon Web Services 账户 相同，也可以不同 Amazon Web Services 账户。

多个外部密钥存储可以共享一个 Amazon VPC，但每个外部密钥存储必须有自己的 VPC 端点服务和私有 DNS 名称。

按照创建端点服务主题中的说明创建具有以下必需值的 VPC 端点服务。对于其他字段，请接受默认值并按要求提供名称。

Field	Value
负载均衡器类型	Network
可用的负载均衡器	选择您在之前的步骤中创建的网络负载均衡器。 如果新的负载均衡器未出现在列表中，请验证其状态是否为活动。负载均衡器状态可能需要几分钟才能从预置中变为活动。
需要接受	False。取消选中复选框。 不需要接受。 Amazon KMS 未经手动接受，无法连接到 VPC 终端节点服务。如果需要接受，则创建外部密钥存储的尝试会失败，并出现 `XksProxyInvalidConfigurationException` 异常。
启用私有 DNS 名称	将私有 DNS 名称与服务关联

Field	Value
私有 DNS 名称	输入在其 Amazon Web Services 区域中唯一的私有 DNS 名称。 私有 DNS 名称必须是更高级别的公有域的子域。例如，如果私有 DNS 名称为 myproxy-private.xks.example.com ，则其必须是公有域（例如 xks.example.com 或 example.com ）的子域。 此私有 DNS 名称必须与在外部密钥存储代理上配置的 TLS 证书中的主题公用名（CN）相匹配。例如，若私有 DNS 名称为 myproxy-private.xks.example.com ，则 TLS 证书上的 CN 必须为 myproxy-private.xks.example.com 或 *.xks.example.com 。 如果证书和私有 DNS 名称不匹配，则将外部密钥存储连接到其外部密钥存储代理的尝试会失败，连接错误代码为 XKS_PROXY_INVALID_TLS_CONFIGURATION 。有关更多信息，请参阅 常规配置错误。
支持的 IP 地址类型	IPv4

步骤 5：验证私有 DNS 名称域

创建 VPC 端点服务时，其域验证状态为 pendingVerification。使用 VPC 端点服务创建外部密钥存储之前，此状态必须为 verified。若要验证您是否拥有与私有 DNS 名称关联的域，您必须在公有 DNS 服务器中创建 TXT 记录。

例如，如果您的 VPC 终端节点服务的私有 DNS 名称为myproxy-private.xks.example.com，则必须在公共域（例如xks.example.com或）中创建 TXT 记录example.com，以公有域为准。Amazon PrivateLink 先打开 TXT 记录，xks.example.com然后再打开example.com。

> **Tip**
>
> 添加 TXT 记录后，Domain verification status（域验证状态）值可能需要几分钟才能从 pendingVerification 变为 verify。

首先，使用以下任一方法找到域的验证状态。有效值包括 verified、pendingVerification 和 failed。

- 在 Amazon VPC 控制台中，选择 Endpoint services（端点服务），然后选择您的端点服务。在详细信息窗格中，查看 Domain verification status（域验证状态）。
- 使用 DescribeVpcEndpointServiceConfigurations 操作。`State` 值在 `ServiceConfigurations.PrivateDnsNameConfiguration.State` 字段中。

如果验证状态不是 `verified`，请按照 Domain ownership verification（域所有权验证）主题中的说明将 TXT 记录添加到域的 DNS 服务器并验证 TXT 记录是否已发布。然后再次检查您的验证状态。

您无需为私有 DNS 域名创建 A 记录。在 Amazon KMS 为您的 VPC 终端节点服务创建接口终端节点时， Amazon PrivateLink 会自动创建一个托管区域，其中包含所需的 Amazon KMS VPC 中私有域名的 A 记录。对于具有 VPC 端点服务连接的外部密钥存储，将外部密钥存储连接到其外部密钥存储代理时，就会发生这种情况。

步骤 6：授权 Amazon KMS 连接 VPC 终端节点服务

请参阅以下过程来管理您的 Amazon VPC 端点服务权限。其中的每个步骤都取决于您的外部密钥存储、VPC 端点服务以及 Amazon Web Services 账户之间的连接和配置。

Same Amazon Web Services 账户

如果您的 VPC 终端节点服务归您的外部密钥存储库所有，则必须将其添加 Amazon KMS 到 VPC 终端节点服务的允许委托人列表中。 Amazon Web Services 账户 这 Amazon KMS 允许为您的 VPC 终端节点服务创建接口终端节点。如果 Amazon KMS 不是允许的主体，则尝试创建外部密钥存储将失败，但会出现异`XksProxyVpcEndpointServiceNotFoundException`常。

按照《Amazon PrivateLink 指南》中管理权限主题中的说明进行操作。使用以下必需值。

Field	Value
Amazon KMS ARN	`cks.kms.<region>.amazonaws.com` 例如，`cks.kms.us-east-1.amazonaws.com`

Cross Amazon Web Services 账户

如果您的 VPC 终端节点服务归其他人所有， Amazon Web Services 账户 则必须将两者 Amazon KMS 以及您的账户添加到 “允许委托人” 列表中。这允许 Amazon KMS 您的外部密钥存储库为您的 VPC 终端节点服务创建接口终端节点。如果 Amazon KMS 不是允许的主体，则创建外部密钥存储

的尝试将失败，并出现 XksProxyVpcEndpointServiceNotFoundException 异常。您需要提供外部密钥存储库所在的 Amazon Web Services 账户 ARN。

按照《Amazon PrivateLink 指南》中管理权限主题中的说明进行操作。使用以下必需值。

Field	Value
Amazon KMS ARN	`cks.kms.<region>.amazonaws.com` 例如，`cks.kms.us-east-1.amazonaws.com`
Amazon Web Services 账户 ARN	`arn:aws:iam:: 111122223333 :role/role_name` 例如，`arn:aws:iam:: 123456789012 :role/cks_role`

下一步：创建外部密钥存储

创建外部密钥存储

您可以在每个 Amazon Web Services 账户 和区域中创建一个或多个外部密钥存储库。每个外部密钥存储都必须与外部密钥管理器相关联 Amazon，以及用于调解外部密钥管理器之间 Amazon KMS 通信的外部密钥存储代理（XKS 代理）。有关更多信息，请参阅 选择外部密钥存储代理连接选项。在开始之前，请确认您需要外部密钥存储。大多数客户都可以使用由密钥材料支持的 KMS Amazon KMS 密钥。

> **Tip**
>
> 一些外部密钥管理器为创建外部密钥存储提供了更简单的方法。有关详细信息，请参阅外部密钥管理器的文档。

在创建外部密钥存储之前，您需要汇编先决条件。在创建过程中，您要指定外部密钥存储的属性。最重要的是，您需要指明您的外部密钥存储是 Amazon KMS 使用公共终端节点还是 VPC 终端节点服务来连接到其外部密钥存储代理。您还可以指定连接详细信息，包括代理的 URI 端点以及该代理端点中向代理 Amazon KMS 发送 API 请求的路径。

注意事项

- KMS 无法 IPv6 与外部密钥存储库进行通信。

- 如果您使用公共终端节点连接，请确保它 Amazon KMS 可以使用 HTTPS 连接通过互联网与您的代理进行通信。这包括在外部密钥存储代理上配置 TLS，并确保与代理之间的 Amazon KMS 任何防火墙都允许代理上进出端口 443 的 IPv4 流量。在创建具有公共端点连接的外部密钥存储时，通过向外部密钥存储代理发送状态请求来 Amazon KMS 测试连接。此测试会验证端点是否可访问，以及外部密钥存储代理是否会接受使用外部密钥存储代理身份验证凭证签名的请求。如果此测试请求失败，则创建外部密钥存储的操作将失败。
- 如果使用 VPC 端点服务连接，请确保网络负载均衡器、私有 DNS 名称和 VPC 端点服务配置正确且可运行。如果外部密钥存储代理不在 VPC 中，则需要确保 VPC 终端节点服务可以与外部密钥存储代理通信。（在将外部密钥存储库连接到其外部密钥存储代理时Amazon KMS 测试 VPC 终端节点服务的连接。）
- Amazon KMS 记录 Amazon CloudWatch 指标和维度，尤其是外部密钥存储的指标和维度。基于其中一些指标的监控图表显示在每个外部密钥存储的 Amazon KMS 控制台中。我们强烈建议您使用这些指标创建监控外部密钥存储的警报。这些警报会在发生性能和操作问题之前提醒您注意相关问题的早期迹象。有关说明，请参阅监控外部密钥存储。
- 外部密钥存储受资源限额的限制。在外部密钥存储中使用 KMS 密钥受请求限额的限制。在设计外部密钥存储实现之前，请查看这些限额。

> **Note**
>
> 检查您的配置中是否存在可能使其无法运行的循环依赖关系。
> 例如，如果您使用 Amazon 资源创建外部密钥存储代理，请确保操作代理不需要通过该代理访问的外部密钥存储中有 KMS 密钥。

所有新的外部密钥存储都在断开连接的状态下创建。在外部密钥存储中创建 KMS 密钥之前，您必须将其连接到外部密钥存储代理。要更改外部密钥存储的属性，请编辑外部密钥存储设置。

主题

汇编先决条件

在创建外部密钥存储库之前，您需要组装所需的组件，包括用于支持外部密钥存储的外部密钥管理器和将 Amazon KMS 请求转换为外部密钥管理器可以理解的格式的外部密钥存储代理。

所有外部密钥存储都需要以下组件。除了这些组件，您还需要提供组件来支持自己选择的外部密钥存储代理连接选项。

> **Tip**
>
> 您的外部密钥管理器可能包含其中一些组件，或者可以为您配置这些组件。有关详细信息，请参阅外部密钥管理器的文档。
>
> 如果您在 Amazon KMS 控制台中创建外部密钥存储，则可以选择上传基于 JSON 的代理配置文件，该文件指定代理 URI 路径和代理身份验证凭据。一些外部密钥存储代理会为您生成此文件。有关详细信息，请参阅外部密钥存储代理或外部密钥管理器的文档。

外部密钥管理器

每个外部密钥存储都需要至少一个外部密钥管理器实例。这可以是实体或虚拟硬件安全模块（HSM），也可以是密钥管理软件。

您可以使用单个密钥管理器，但我们建议准备至少两个共享加密密钥的相关密钥管理器实例，以便实现冗余配置。外部密钥存储不需要独占使用外部密钥管理器。但是，外部密钥管理器必须有能力处理来自使用外部密钥存储中的 KMS 密钥来保护您的资源的 Amazon 服务的预期频率的加密和解密请求。您的外部密钥管理器应配置为每秒最多处理 1800 个请求，并在 250 毫秒的超时限制内响应每个请求。我们建议您将外部密钥管理器放在附近， Amazon Web Services 区域 以便网络往返时间 (RTT) 为 35 毫秒或更短。

如果外部密钥存储代理允许，则可以更改与外部密钥存储代理关联的外部密钥管理器，但新的外部密钥管理器必须是具有相同密钥材料的备份或快照。如果您与 KMS 密钥关联的外部密钥不再可供外部密钥存储代理使用，则 Amazon KMS 无法解密使用 KMS 密钥加密的密文。

外部密钥存储代理必须可以访问外部密钥管理器。如果代理的GetHealthStatus响应报告所有外部密钥管理器实例都是`Unavailable`，则所有创建外部密钥存储的尝试都将失败，并显示为`XksProxyUriUnreachableException`。

外部密钥存储代理

您必须指定符合 Amazon KMS 外部密钥存储代理 API 规范中设计要求的外部密钥存储代理（XKS 代理）。您可以开发或购买外部密钥存储代理，也可以使用外部密钥管理器提供或内置的外部密钥存储代理。 Amazon KMS 建议将您的外部密钥存储代理配置为每秒最多处理 1800 个请求，并在每个请求的 250 毫秒超时内做出响应。我们建议您将外部密钥管理器放在附近， Amazon Web Services 区域 以便网络往返时间 (RTT) 为 35 毫秒或更短。

您可以将外部密钥存储代理用于多个外部密钥存储，但每个外部密钥存储必须在其请求的外部密钥存储代理中具有唯一的 URI 端点和路径。

如果您使用的是 VPC 端点服务连接，则可以在 Amazon VPC 中找到您的外部密钥存储代理，但这不是必需的。您可以将代理定位在外部 Amazon（例如私有数据中心），并且只能使用 VPC 终端节点服务与代理通信。

代理身份验证凭证

要创建外部密钥存储，必须指定外部密钥存储代理身份验证凭证（`XksProxyAuthenticationCredential`）。

您必须在外部密钥存储代理 Amazon KMS 上为建立身份验证凭据(`XksProxyAuthenticationCredential`)。 Amazon KMS 使用签名版本 4 (Sigv4) 流程使用外部密钥存储代理身份验证凭据对代理的请求进行签名，从而对代理进行身份验证。您可以在创建外部密钥存储时指定身份验证凭证，并且可以随时对其进行更改。如果代理要轮换凭证，则务必更新外部密钥存储的凭证值。

代理身份验证凭证有两个部分。您必须为外部密钥存储提供这两部分。

- 访问密钥 ID：标识秘密访问密钥。您能以明文形式提供此 ID。
- 秘密访问密钥：凭证的秘密部分。 Amazon KMS 在存储凭据之前对凭据中的私有访问密钥进行加密。

Amazon KMS 用于签署外部密钥存储代理请求的 Sigv4 凭据与您账户中任何委托人关联的任何 Sigv4 凭据无关。 Amazon Identity and Access Management Amazon 不要将任何 IAM SigV4 凭证重复用于外部密钥存储代理。

代理连接

要创建外部密钥存储，您必须指定外部密钥存储代理连接选项（`XksProxyConnectivity`）。

Amazon KMS 可以使用公共终端节点或 Amazon Virtu al Private Cloud (Amazon VPC) 终端节点服务与您的外部密钥存储代理进行通信。公有端点虽然更易于配置和维护，但可能无法满足每次安装的安全要求。如果您选择 Amazon VPC 端点服务连接选项，则必须创建和维护所需的组件，包括在两个不同可用区内至少有两个子网的 Amazon VPC、具有网络负载均衡器和目标组的 VPC 端点服务，以及 VPC 端点服务的私有 DNS 名称。

您可以为外部密钥存储更改代理连接选项。不过，您必须确保外部密钥存储中与 KMS 密钥相关的密钥材料持续可用。否则， Amazon KMS 无法解密使用这些 KMS 密钥加密的任何密文。

有关确定哪种代理连接选项最适合外部密钥存储的帮助信息，请参阅 选择外部密钥存储代理连接选项。有关创建和配置 VPC 端点服务连接的帮助信息，请参阅 配置 VPC 端点服务连接。

代理 URI 端点

要创建外部密钥存储库，必须指定用于向外部密钥存储代理发送请求的端点 (XksProxyUriEndpoint)。 Amazon KMS

协议必须是 HTTPS。 Amazon KMS 通过端口 44 IPv4 3 进行通信。不要在代理 URI 端点值中指定端口。

- 公有端点连接 – 为外部密钥存储代理指定公开可用的端点。在创建外部密钥存储之前，此端点必须可供访问。
- VPC 端点服务连接 – 指定 `https://`，后接 VPC 端点服务的私有 DNS 名称。

在外部密钥存储代理上配置的 TLS 服务器证书必须与外部密钥存储代理 URI 端点中的域名相匹配，并且由支持外部密钥存储的证书颁发机构颁发。有关列表，请参阅 Trusted Certificate Authorities（受信任的证书颁发机构）。在颁发 TLS 证书之前，证书颁发机构会要求您提供域所有权证明。

TLS 证书上的主题通用名称（CN）必须与私有 DNS 名称相匹配。例如，若私有 DNS 名称为 `myproxy-private.xks.example.com`，则 TLS 证书上的 CN 必须为 `myproxy-private.xks.example.com` 或 `*.xks.example.com`。

您可以更改代理 URI 端点，但请确保外部密钥存储代理有权访问与外部密钥存储中的 KMS 密钥关联的密钥材料。否则， Amazon KMS 无法解密使用这些 KMS 密钥加密的任何密文。

唯一性要求

- 合并的代理 URI 端点（`XksProxyUriEndpoint`）和代理 URI 路径（`XksProxyUriPath`）值在 Amazon Web Services 账户 和区域中必须具备唯一性。
- 使用公有端点连接的外部密钥存储可以共享相同的代理 URI 端点，前提是这些外部密钥存储具有不同的代理 URI 路径值。
- 具有公共终端节点连接的外部密钥存储不能使用与任何具有相同 VPC 终端节点服务连接的外部密钥存储相同的代理 URI 终端节点值 Amazon Web Services 区域，即使密钥存储库位于不同的位置 Amazon Web Services 账户。
- 每个具有 VPC 端点连接的外部密钥存储都必须有自己的私有 DNS 名称。代理 URI 终端节点（私有 DNS 名称）在 Amazon Web Services 账户 和区域中必须是唯一的。

代理 URI 路径

要创建外部密钥存储库，您必须在外部密钥存储代理中指定所需代理的基本路径 APIs。该值必须以开头/且必须以/kms/xks/v1 结尾，其中v1表示外部密钥存储代理的 Amazon KMS API 版本。此路径可以在必要元素之间包含可选前缀，例如 `/example-prefix/kms/xks/v1`。要找到此值，请参阅外部密钥存储代理的文档。

Amazon KMS 将代理请求发送到由代理 URI 端点和代理 URI 路径串联所指定的地址。例如，如果代理 URI 端点是，`https://myproxy.xks.example.com`而代理 URI 路径是`/kms/xks/v1`，则将其代理 API 请求 Amazon KMS 发送到`https://myproxy.xks.example.com/kms/xks/v1`。

您可以更改代理 URI 路径，但要确保外部密钥存储代理有权访问与外部密钥存储中的 KMS 密钥关联的密钥材料。否则， Amazon KMS 无法解密使用这些 KMS 密钥加密的任何密文。

唯一性要求

- 合并的代理 URI 端点（`XksProxyUriEndpoint`）和代理 URI 路径（`XksProxyUriPath`）值在 Amazon Web Services 账户 和区域中必须具备唯一性。

VPC 终端节点服务

指定用于与外部密钥存储代理通信的 Amazon VPC 端点服务的名称。只有使用 VPC 端点服务连接的外部密钥存储才需要此组件。有关为外部密钥存储设置和配置 VPC 端点服务的帮助信息，请参阅 配置 VPC 端点服务连接。

VPC 端点服务必须具有以下属性：

- VPC 端点服务可与外部密钥存储位于同一或不同 Amazon Web Services 账户 中。
  - VPC 终端节点服务必须与外部密钥存储库位于同一 Amazon Web Services 区域 位置。
  - 如果 VPC 终端节点服务位于其他 VPC 终端节点服务中，则需要提供该服务的 Amazon Web Services 账户 ID Amazon Web Services 账户。
- 其必须有至少连接到两个子网的网络负载均衡器（NLB），每个子网位于不同的可用区中。
- VPC 终端节点服务的允许委托人列表必须包括该区域的 Amazon KMS 服务委托人：`cks.kms.<region>.amazonaws.com`，例如`cks.kms.us-east-1.amazonaws.com`。
  - 如果您的 Amazon VPC 终端节点服务归拥有外部密钥存储库 (XKS) 以外 Amazon Web Services 账户 的其他人所有，则还需要允许 XKS 访问该 VPC 终端节点服务。 Amazon Web Services 账户 为此，请将 XKS Amazon Web Services 账户 ID 列为亚马逊 VPC 终端节点服务的委托人。
- 其不得要求接受连接请求。

- 其必须在更高级别的公有域内具有私有 DNS 名称。例如，在 xks.example.com 公有域中，私有 DNS 名称可能为 myproxy-private.xks.example.com。

  使用 VPC 端点服务连接的外部密钥存储的私有 DNS 名称，在其 Amazon Web Services 区域中必须具备唯一性。

- 私有 DNS 名称域的域验证状态必须为 verified。
- 在外部密钥存储代理上配置的 TLS 服务器证书，必须指定可访问端点的私有 DNS 主机名。

唯一性要求

- 具有 VPC 端点连接的外部密钥存储可以共享 Amazon VPC，但每个外部密钥存储必须有自己的 VPC 端点服务和私有 DNS 名称。

代理配置文件

代理配置文件是基于 JSON 的可选文件，其中包含外部密钥存储的代理 URI 路径和代理身份验证凭证属性的值。在 Amazon KMS 控制台中创建或编辑外部密钥存储时，您可以上传代理配置文件来为外部密钥存储提供配置值。使用此文件可以避免键入和粘贴错误，并确保外部密钥存储中的值与外部密钥存储代理中的值相匹配。

代理配置文件由外部密钥存储代理生成。要了解外部密钥存储代理是否提供代理配置文件，请参阅您的外部密钥存储代理文档。

下面是具有虚拟值的格式正确的代理配置文件示例。

```
{
  "XksProxyUriPath": "/example-prefix/kms/xks/v1",
  "XksProxyAuthenticationCredential": {
    "AccessKeyId": "ABCDE12345670EXAMPLE",
    "RawSecretAccessKey": "0000EXAMPLEFA5FT0mCc3DrGUe2sti527BitkQ0Zr9MO9+vE="
  }
}
```

只有在 Amazon KMS 控制台中创建或编辑外部密钥存储时，才能上传代理配置文件。您不能将其与CreateCustomKeyStore或UpdateCustomKeyStore操作一起使用，但可以使用代理配置文件中的值来确保参数值正确无误。

创建新的外部密钥存储

准备好必要的先决条件后，可以在 Amazon KMS 控制台中或使用CreateCustomKeyStore操作创建新的外部密钥存储。

使用控制 Amazon KMS 台

在创建外部密钥存储之前，请选择代理连接类型，并确保已创建和配置了所有必需组件。如果需要查找任何所需值的帮助信息，请查阅外部密钥存储代理或密钥管理软件的文档。

> **Note**
>
> 在中创建外部密钥存储时 Amazon Web Services 管理控制台，可以上传基于 JSON 的代理配置文件，其中包含代理 URI 路径和代理身份验证凭据的值。一些代理会为您生成此文件，但其并非必要项目。

1. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/km s 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择 Custom key stores（自定义密钥存储）、External key stores（外部密钥存储）。
4. 选择 Create external key store（创建外部密钥存储）。
5. 为外部密钥存储输入易记名称。该名称在您账户的所有外部密钥存储中必须具备唯一性。

   > **Important**
   >
   > 不要在此字段中包含机密或敏感信息。此字段可能会以纯文本形式显示在 CloudTrail 日志和其他输出中。

6. 选择代理连接类型。

   您的代理连接选择决定了外部密钥存储代理所需的组件。有关做出此选择的帮助信息，请参阅 选择外部密钥存储代理连接选项。

   a. 如果您的 VPC 端点服务位于不同的 Amazon Web Services 账户中，请选择跨账户 VPC 端点服务。然后在 VPC 终端节点服务所有者账户 Amazon Web Services 账户 ID 字段中输入 VPC 终端节点所有者的 ID。

b. 选择或输入此外部密钥存储的 VPC 端点服务的名称。此步骤仅在外部密钥存储代理连接类型为 VPC 端点服务时出现。

VPC 终端节点服务及其 VPCs 必须满足外部密钥存储的要求。有关更多信息，请参阅 the section called “汇编先决条件”。

7. 选择或输入此外部密钥存储的 VPC 端点服务的名称。此步骤仅在外部密钥存储代理连接类型为 VPC 端点服务时出现。

VPC 终端节点服务及其 VPCs 必须满足外部密钥存储的要求。有关更多信息，请参阅 the section called “汇编先决条件”。

8. 输入代理 URI 端点。协议必须是 HTTPS。 Amazon KMS 通过端口 44 IPv4 3 进行通信。不要在代理 URI 端点值中指定端口。

如果 Amazon KMS 识别出您在上一步中指定的 VPC 终端节点服务，它将为您填写此字段。

若使用公有端点连接，请输入公开可用的端点 URI。若使用 VPC 端点连接，输入 `https://`，后接 VPC 端点服务的私有 DNS 名称。

9. 要输入代理 URI 路径前缀和代理身份验证凭证的值，请上传代理配置文件或手动输入值。

- 如果您有包含代理 URI 路径和代理身份验证凭证值的可选代理配置文件，请选择 Upload configuration file（上传配置文件）。然后按照步骤上传该文件。

  上传文件后，控制台会在可编辑字段中显示文件中的值。您可以立即更改值，也可以在创建外部密钥存储后编辑这些值。

  要显示秘密访问密钥的值，请选择 Show secret access key（显示私有访问密钥）。

- 如果您没有代理配置文件，则可以手动输入代理 URI 路径和代理身份验证凭证值。

  a. 如果没有代理配置文件，您可以手动输入代理 URI。控制台提供所需的/kms/xks/v1 值。

  如果代理 URI 路径包含可选前缀，例如 `/example-prefix/kms/xks/v1` 中的 `example-prefix`，请在 Proxy URI path prefix（代理 URI 路径前缀）字段中输入该前缀。如若没有，则将字段留空。

  b. 如果没有代理配置文件，您可以手动输入代理身份验证凭证。访问密钥 ID 和秘密访问密钥均为必填项。

  - 在 Proxy credential: Access key ID（代理凭证：访问密钥 ID）中，输入代理身份验证凭证的访问密钥 ID。访问密钥 ID 标识秘密访问密钥。

- 在 Proxy credential: Secret access key（代理凭证：秘密访问密钥）中，输入代理身份验证凭证的秘密访问密钥。

  要显示秘密访问密钥的值，请选择 Show secret access key（显示私有访问密钥）。

  此过程不会设置或更改您在外部密钥存储代理上建立的身份验证凭证。其只是将这些值关联到外部密钥存储。有关设置、更改和轮换代理身份验证凭证的信息，请参阅外部密钥存储代理或密钥管理软件的文档。

  如果您的代理身份验证凭证发生变化，请编辑外部密钥存储的凭证设置。

10. 选择 Create external key store（创建外部密钥存储）。

当该过程成功时，新的外部密钥存储将显示在账户和区域的外部密钥存储列表中。如果该过程失败，则会显示一条错误消息，描述问题并提供有关如何解决该问题的帮助。如果您需要更多帮助，请参阅外部密钥的 CreateKey 错误。

下一步：不会自动连接新的外部密钥存储。必须先将外部密钥存储库连接到其外部密钥存储代理服务器，然后才能在外部密钥存储库中进行创建 Amazon KMS keys 。

使用 Amazon KMS API

您可以使用该CreateCustomKeyStore操作来创建新的外部密钥存储库。如果需要查找所需参数值的帮助信息，请参阅外部密钥存储代理或密钥管理软件的文档。

> **Tip**
>
> 使用 `CreateCustomKeyStore` 操作时，您无法上传代理配置文件。不过，您可以使用代理配置文件中的值来确保参数值正确无误。

要创建外部密钥存储，`CreateCustomKeyStore` 操作需要以下参数值。

- `CustomKeyStoreName` - 外部密钥存储的易记名称，该名称在账户中具备唯一性。

> **Important**
>
> 不要在此字段中包含机密或敏感信息。此字段可能会以纯文本形式显示在 CloudTrail 日志和其他输出中。

- CustomKeyStoreType – 指定 EXTERNAL_KEY_STORE。
- XksProxyConnectivity – 指定 PUBLIC_ENDPOINT 或 VPC_ENDPOINT_SERVICE。
- XksProxyAuthenticationCredential – 指定访问密钥 ID 和秘密访问密钥。
- XksProxyUriEndpoint – Amazon KMS 用于与外部密钥存储代理通信的端点。
- XksProxyUriPath— 代理内到代理的路径 APIs。
- XksProxyVpcEndpointServiceName – 仅当 XksProxyConnectivity 值为 VPC_ENDPOINT_SERVICE 时才需要。

> **Note**
>
> 如果您使用 Amazon CLI 版本 1.0，请在指定带有 HTTP 或 HTTPS 值的参数（例如XksProxyUriEndpoint参数）之前运行以下命令。
>
> ```
> aws configure set cli_follow_urlparam false
> ```
>
> 否则， Amazon CLI 版本 1.0 会将参数值替换为在该 URI 地址中找到的内容，从而导致以下错误：
>
> ```
> Error parsing parameter '--xks-proxy-uri-endpoint': Unable to retrieve
> https:// : received non 200 status code of 404
> ```

以下示例使用虚拟值。在运行命令之前，将虚拟值替换为外部密钥存储的有效值。

创建使用公有端点连接的外部密钥存储。

```
$ aws kms create-custom-key-store
        --custom-key-store-name ExampleExternalKeyStorePublic \
        --custom-key-store-type EXTERNAL_KEY_STORE \
        --xks-proxy-connectivity PUBLIC_ENDPOINT \
        --xks-proxy-uri-endpoint https://myproxy.xks.example.com \
        --xks-proxy-uri-path /kms/xks/v1 \
        --xks-proxy-authentication-credential
 AccessKeyId=<value>,RawSecretAccessKey=<value>
```

创建使用 VPC 端点服务连接的外部密钥存储。

```
$ aws kms create-custom-key-store
        --custom-key-store-name ExampleExternalKeyStoreVPC \
        --custom-key-store-type EXTERNAL_KEY_STORE \
        --xks-proxy-connectivity VPC_ENDPOINT_SERVICE \
        --xks-proxy-vpc-endpoint-service-name com.amazonaws.vpce.us-east-1.vpce-svc-
example \
        --xks-proxy-uri-endpoint https://myproxy-private.xks.example.com \
        --xks-proxy-uri-path /kms/xks/v1 \
        --xks-proxy-authentication-credential
 AccessKeyId=<value>,RawSecretAccessKey=<value>
```

当此操作成功时，CreateCustomKeyStore 将返回自定义密钥存储 ID，如以下示例响应中所示。

```
{
    "CustomKeyStoreId": cks-1234567890abcdef0
}
```

如果操作失败，请更正异常指示的错误，然后重试。有关其他帮助，请参阅排查外部密钥存储的问题。

下一步：要使用外部密钥存储，请将其连接到外部密钥存储代理。

编辑外部密钥存储属性

您可以编辑现有外部密钥存储的选定属性。

在连接或断开外部密钥存储后，您可以编辑某些属性。若要编辑其他属性，则必须先断开外部密钥存储与其外部密钥存储代理的连接。外部密钥存储的连接状态必须为 DISCONNECTED。在外部密钥存储断开后，您可以管理密钥存储及其 KMS 密钥，但无法在外部密钥存储中创建或使用 KMS 密钥。要查找外部密钥存储库的连接状态，请使用DescribeCustomKeyStores操作或查看外部密钥存储详细信息页面上的 “常规配置” 部分。

在更新外部密钥存储库的属性之前，使用新值向外部密钥存储代理 Amazon KMS 发送GetHealthStatus请求。如果请求成功，则表明您可以使用更新后的属性值连接到外部密钥存储代理并进行身份验证。如果请求失败，编辑操作也将失败，并且会显示标识错误的异常。

编辑操作完成后，外部密钥存储库的更新属性值将显示在 Amazon KMS 控制台和DescribeCustomKeyStores响应中。不过，要让更改完全生效，可能需要长达五分钟的时间。

如果您在 Amazon KMS 控制台中编辑外部密钥存储，则可以选择上传基于 JSON 的代理配置文件，该文件指定代理 URI 路径和代理身份验证凭据。一些外部密钥存储代理会为您生成此文件。有关详细信息，请参阅外部密钥存储代理或外部密钥管理器的文档。

> **Warning**
>
> 更新后的属性值必须将您的外部密钥存储连接到与先前值相同的外部密钥管理器的代理，或者使用相同的加密密钥对外部密钥管理器进行备份或快照。如果外部密钥存储永久失去对与其 KMS 密钥关联的外部密钥的访问权限，则在这些外部密钥下加密的加密文字将无法恢复。特别是，更改外部密钥存储库的代理连接可能会 Amazon KMS 阻止访问您的外部密钥。

> **Tip**
>
> 一些外部密钥管理器为编辑外部密钥存储属性提供了更简单的方法。有关详细信息，请参阅外部密钥管理器的文档。

您可以更改外部密钥存储的以下属性。

可编辑的外部密钥存储属性	任何连接状态	需要“已断开连接”状态
自定义密钥存储名称 自定义密钥存储必需的易记名称。 **Important** 不要在此字段中包含机密或敏感信息。此字段可能会以纯文本形式显示在 CloudTrail 日志和其他输出中。	✓	
代理身份验证凭证 () XksProxyAuthenticationCredential （即使只更改一个元素，您也必须同时指定访问密钥 ID 和秘密访问密钥。）	✓	
代理 URI 路径 (XksProxyUriPath)	✓	

可编辑的外部密钥存储属性	任何连接状态	需要“已断开连接”状态
代理连接 (XksProxyConnectivity) （您还必须更新代理 URI 端点。如果要更改为 VPC 端点服务连接，则必须指定代理 VPC 端点服务名称。）		✓
代理 URI 端点 (XksProxyUriEndpoint) 如果更改代理端点 URI，您可能还需要更改关联的 TLS 证书。		✓
代理 VPC 终端节点服务名称 (XksProxyVpcEndpointServiceName) （此字段是 VPC 端点服务连接的必填字段）		✓
VPC 终端节点服务所有者 (XksProxyVpcEndpointServiceOwner) （此字段是 VPC 端点服务连接的必填字段）		✓

编辑外部密钥存储的属性

您可以在 Amazon KMS 控制台中或使用UpdateCustomKeyStore操作来编辑外部密钥存储库的属性。

使用控制 Amazon KMS 台

编辑密钥库时，可以更改任何可编辑的值。某些更改要求将外部密钥存储与其外部密钥存储代理断开连接。

如果您正在编辑代理 URI 路径或代理身份验证凭证，则可以输入新值或上传包含新值的外部密钥存储代理配置文件。

1. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/km s 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。

3. 在导航窗格中，选择 Custom key stores（自定义密钥存储）、External key stores（外部密钥存储）。
4. 选择要编辑的密钥存储。
   - 如有必要，请断开外部密钥存储与其外部密钥存储代理的连接。从 Key store actions（密钥存储操作）菜单中选择 Disconnect（断开连接）。
5. 从 Key store actions（密钥存储操作）菜单中选择 Edit（编辑）。
6. 更改一个或多个可编辑的外部密钥存储属性。您还可以上传包含代理 URI 路径和代理身份验证凭证值的外部密钥存储代理配置文件。即使文件中指定的某些值没有更改，也可以使用代理配置文件。
7. 选择 Update external key store（更新外部密钥存储）。
8. 查看警告，如果您决定继续，请确认警告，然后选择 Update external key store（更新外部密钥存储）。

   在此过程成功后，将显示一条消息，描述您编辑的属性。如果此过程失败，则会显示一条错误消息，描述问题并提供有关如何解决问题的帮助。
9. 如有必要，重新连接外部密钥存储。从 Key store actions（密钥存储操作）菜单中选择 Connect（连接）。

   您可以让外部密钥存储保持断开状态。不过，在其断开连接后，您无法在外部密钥存储中创建 KMS 密钥或在加密操作中使用外部密钥存储中的 KMS 密钥。

使用 Amazon KMS API

要更改外部密钥存储库的属性，请使用UpdateCustomKeyStore操作。您可以在同一操作中更改外部密钥存储的多个属性。如果操作成功，则 Amazon KMS 返回一个 HTTP 200 响应和一个没有属性的 JSON 对象。

使用 `CustomKeyStoreId` 参数标识外部密钥存储。使用其他参数更改属性。您不能在 `UpdateCustomKeyStore` 操作中使用代理配置文件。只有 Amazon KMS 控制台支持代理配置文件。不过，您可以使用代理配置文件来帮助自己确定外部密钥存储代理的正确参数值。

本部分中的示例使用 Amazon Command Line Interface (Amazon CLI)，但您可以使用任何受支持的编程语言。

在开始之前，如有必要，请断开外部密钥存储与其外部密钥存储代理的连接。更新后，如有必要，您可以将外部密钥存储重新连接到其外部密钥存储代理。您可以让外部密钥存储保持断开状态，但之后必须

先重新连接，才能在密钥存储中创建新的 KMS 密钥或在密钥存储中使用现有 KMS 密钥来进行加密操作。

> **Note**
>
> 如果您使用 Amazon CLI 版本 1.0，请在指定带有 HTTP 或 HTTPS 值的参数（例如`XksProxyUriEndpoint`参数）之前运行以下命令。
>
> ```
> aws configure set cli_follow_urlparam false
> ```
>
> 否则， Amazon CLI 版本 1.0 会将参数值替换为在该 URI 地址中找到的内容，从而导致以下错误：
>
> ```
> Error parsing parameter '--xks-proxy-uri-endpoint': Unable to retrieve
> https:// : received non 200 status code of 404
> ```

更改外部密钥存储的名称

第一个示例使用UpdateCustomKeyStore操作将外部密钥存储库的友好名称更改为`XksKeyStore`。该命令使用 `CustomKeyStoreId` 参数标识自定义密钥存储，使用 `CustomKeyStoreName` 指定自定义密钥存储的新名称。将所有示例值替换为外部密钥存储的实际值。

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 --new-
custom-key-store-name XksKeyStore
```

更改代理身份验证凭证

以下示例更新了 Amazon KMS 用于向外部密钥存储代理进行身份验证的代理身份验证凭证。如果凭证在代理上轮换，您可以使用这样的命令来更新凭证。

首先更新外部密钥存储代理上的凭证。然后使用此功能将更改报告给 Amazon KMS。（您的代理将暂时支持新旧凭证，因此您有时间在中 Amazon KMS更新凭证。）

您必须始终在凭证中同时指定访问密钥 ID 和秘密访问密钥，即使只更改了一个值也是如此。

前两个命令设置变量来保存凭证值。`UpdateCustomKeyStore` 操作使用 `CustomKeyStoreId` 参数来标识外部密钥存储。该操作使用 `XksProxyAuthenticationCredential` 参数及其 `AccessKeyId` 和 `RawSecretAccessKey` 字段来指定新凭证。将所有示例值替换为外部密钥存储的实际值。

```
$ accessKeyID=access key id
$ secretAccessKey=secret access key

$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 \
       --xks-proxy-authentication-credential \
           AccessKeyId=$accessKeyId,RawSecretAccessKey=$secretAccessKey
```

更改代理 URI 路径

以下示例更新了代理 URI 路径（`XksProxyUriPath`）。在和区域中，代理 URI 端点和代理 URI 路径的 Amazon Web Services 账户 组合必须是唯一的。将所有示例值替换为外部密钥存储的实际值。

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 \
           --xks-proxy-uri-path /kms/xks/v1
```

更改 VPC 端点服务连接

以下示例使用UpdateCustomKeyStore操作将外部密钥存储代理连接类型更改为`VPC_ENDPOINT_SERVICE`。要进行此更改，您必须指定 VPC 端点服务连接所需的值，包括 VPC 端点服务名称（`XksProxyVpcEndpointServiceName`）和包含 VPC 端点服务的私有 DNS 名称的代理 URI 端点（`XksProxyUriEndpoint`）值。将所有示例值替换为外部密钥存储的实际值。

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 \
           --xks-proxy-connectivity "VPC_ENDPOINT_SERVICE" \
           --xks-proxy-uri-endpoint https://myproxy-private.xks.example.com \
           --xks-proxy-vpc-endpoint-service-name com.amazonaws.vpce.us-east-1.vpce-
svc-example
```

更改为公有端点连接

以下示例将外部密钥存储代理连接类型更改为 `PUBLIC_ENDPOINT`。进行此更改时，必须更新代理 URI 端点（`XksProxyUriEndpoint`）值。将所有示例值替换为外部密钥存储的实际值。

> **Note**
>
> VPC 端点连接提供的安全性高于公有端点连接。在更改为公有端点连接之前，请考虑其他选项，包括在本地找到外部密钥存储代理以及仅使用 VPC 进行通信。

```
$ aws kms update-custom-key-store --custom-key-store-id cks-1234567890abcdef0 \
```

```
        --xks-proxy-connectivity "PUBLIC_ENDPOINT" \
        --xks-proxy-uri-endpoint https://myproxy.xks.example.com
```

查看外部密钥存储

您可以使用 Amazon KMS 控制台或使用DescribeCustomKeyStores操作来查看每个账户和区域中的外部密钥存储库。

查看外部密钥存储时，可以看到以下内容：

- 有关密钥存储的基本信息，包括其易记名称、ID、密钥存储类型和创建日期。
- 外部密钥存储代理的配置信息，包括连接类型、代理 URI 端点和路径，以及当前代理身份验证凭证的访问密钥 ID。
- 如果外部密钥存储代理使用 VPC 端点服务连接，则控制台将显示 VPC 端点服务的名称。
- 当前连接状态。

> **Note**
>
> 如果连接状态值为 Disconnected（已断开连接），则表示外部密钥存储从未连接过，或者已有意与其外部密钥存储代理断开连接。但是，如果您尝试使用已连接的外部密钥存储中的 KMS 密钥时失败，则可能表示该外部密钥存储或其代理存在问题。有关帮助信息，请参阅外部密钥存储连接错误。

- 带有 Amazon CloudWatch 指标图表的 “监控” 部分，旨在帮助您检测和解决外部密钥存储的问题。有关解释图表、在规划和故障排除中使用图表以及根据图表中的指标创建 CloudWatch 警报的帮助，请参阅监控外部密钥存储。

外部密钥存储属性

外部密钥库的以下属性在 Amazon KMS 控制台和DescribeCustomKeyStores响应中可见。

自定义密钥存储属性

以下值显示在每个自定义密钥库详情页面的 “常规配置” 部分。这些属性适用于所有自定义密钥存储库，包括密钥存储库和外部 Amazon CloudHSM 密钥存储库。

自定义密钥存储 ID

Amazon KMS 分配给自定义密钥库的唯一 ID。

自定义密钥存储名称

在创建自定义密钥存储时为其分配的易记名称。您可以随时更改此值。

自定义密钥存储类型

自定义密钥存储的类型。有效值为 Amazon CloudHSM （`AWS_CLOUDHSM`）或外部密钥存储（`EXTERNAL_KEY_STORE`）。创建自定义密钥存储后无法更改其类型。

Creation date (创建日期)

创建自定义密钥存储的日期。此日期显示为 Amazon Web Services 区域的本地时间。

连接状态

表示自定义密钥存储已连接到其备用密钥存储。仅当自定义密钥存储从未连接到其备用密钥存储或故意断开连接时，连接状态才会为 `DISCONNECTED`。有关更多信息，请参阅 the section called “连接状态”。

外部密钥存储配置属性

以下值显示在每个外部密钥存储详细信息页面的外部密钥存储代理配置部分和DescribeCustomKeyStores响应`XksProxyConfiguration`元素中。有关每个字段的详细描述，包括唯一性要求，以及有关帮助确定每个字段的正确值的详细描述，请参阅 Creating an external key store（创建外部密钥存储）主题中的 the section called “汇编先决条件”。

代理连接

指示外部密钥存储是使用公有端点连接还是 VPC 端点服务连接。

代理 URI 端点

Amazon KMS 用于连接到您的外部密钥存储代理的端点。

代理 URI 路径

从 Amazon KMS 发送代理 API 请求的代理 URI 端点的路径。

代理凭证：访问密钥 ID

您在外部密钥存储代理上建立的代理身份验证凭证的一部分。访问密钥 ID 标识出凭证中的秘密访问密钥。

Amazon KMS 使用 Sigv4 签名过程和代理身份验证凭据来签署其对外部密钥存储代理的请求。签名中的凭据允许外部密钥存储代理代表您对来自 Amazon KMS的请求进行身份验证。

VPC 端点服务名称

支持您的外部密钥存储的 Amazon VPC 端点服务的名称。此值仅在外部密钥存储使用 VPC 端点服务连接时显示。您可以在 VPC 中找到您的外部密钥存储代理，也可以使用 VPC 端点服务与您的外部密钥存储代理安全地进行通信。

VPC 端点服务所有者 ID

支持您的外部密钥存储的 Amazon VPC 端点服务 ID。此值仅在外部密钥存储使用 VPC 端点服务连接时显示。您可以在 VPC 中找到您的外部密钥存储代理，也可以使用 VPC 端点服务与您的外部密钥存储代理安全地进行通信。

查看您的外部密钥存储属性

您可以在 Amazon KMS 控制台中或使用DescribeCustomKeyStores操作来查看您的外部密钥存储库及其关联属性。

使用控制 Amazon KMS 台

要查看给定账户和区域中的外部密钥存储，请按照以下流程操作。

1. 登录 Amazon Web Services 管理控制台 并在 https://console.aws.amazon.com/km s 处打开 Amazon Key Management Service (Amazon KMS) 控制台。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择 Custom key stores（自定义密钥存储）、External key stores（外部密钥存储）。
4. 要查看有关外部密钥存储的详细信息，请选择密钥存储名称。

使用 Amazon KMS API

要查看您的外部密钥存储库，请使用DescribeCustomKeyStores操作。默认情况下，此操作将返回账户和区域中的所有自定义密钥存储。不过，您可以使用 `CustomKeyStoreId` 或 `CustomKeyStoreName` 参数（但不能同时使用两者）将输出限制到特定的自定义密钥存储。

对于自定义密钥存储，输出包含自定义密钥存储 ID、名称和类型以及密钥存储的连接状态。如果连接状态为 `FAILED`，则输出还包含描述错误原因的 `ConnectionErrorCode`。有关解释外部密钥存储 `ConnectionErrorCode` 的帮助，请参阅 外部密钥存储的连接错误代码。

对于外部密钥存储，输出还包括 `XksProxyConfiguration` 元素。此元素包括连接类型、代理 URI 端点、代理 URI 路径和代理身份验证凭证的访问密钥 ID。

本部分中的示例使用 Amazon Command Line Interface (Amazon CLI)，但您可以使用任何受支持的编程语言。

例如，以下命令返回账户和区域中的所有自定义密钥存储。您可以使用 Limit 和 Marker 参数来浏览输出中的自定义密钥存储。

```
$ aws kms describe-custom-key-stores
```

以下示例命令使用 CustomKeyStoreName 参数以仅获取具有 ExampleXksPublic 易记名称的示例外部密钥存储。此示例密钥存储使用公有端点连接。该密钥存储连接到其外部密钥存储代理。

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksPublic
{
    "CustomKeyStores": [
    {
      "CustomKeyStoreId": "cks-1234567890abcdef0",
      "CustomKeyStoreName": "ExampleXksPublic",
      "ConnectionState": "CONNECTED",
      "CreationDate": "2022-12-14T20:17:36.419000+00:00",
      "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE12345670EXAMPLE",
        "Connectivity": "PUBLIC_ENDPOINT",
        "UriEndpoint": "https://xks.example.com:6443",
        "UriPath": "/example/prefix/kms/xks/v1"
      }
    }
  ]
}
```

以下命令获取具有 VPC 端点服务连接的示例外部密钥存储。在此示例中，外部密钥存储连接到其外部密钥存储代理。

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksVpc
{
    "CustomKeyStores": [
    {
      "CustomKeyStoreId": "cks-9876543210fedcba9",
      "CustomKeyStoreName": "ExampleXksVpc",
      "ConnectionState": "CONNECTED",
      "CreationDate": "2022-12-13T18:34:10.675000+00:00",
      "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
```

```
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE98765432EXAMPLE",
        "Connectivity": "VPC_ENDPOINT_SERVICE",
        "UriEndpoint": "https://example-proxy-uri-endpoint-vpc",
        "UriPath": "/example/prefix/kms/xks/v1",
        "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-example"
      }
    }
  ]
}
```

ConnectionState 为 Disconnected 则表示外部密钥存储从未连接过，或者已有意与其与外部密钥存储代理断开连接。但是，如果您尝试使用已连接的外部密钥存储中的 KMS 密钥时失败，则可能表示该外部密钥存储代理或其他外部组件存在问题。

如果外部密钥存储的 ConnectionState 为 FAILED，则 DescribeCustomKeyStores 响应包含说明错误原因的 ConnectionErrorCode 元素。

例如，在以下输出中，该XKS_PROXY_TIMED_OUT值表示 Amazon KMS 可以连接到外部密钥存储代理，但是由于外部密钥存储代理 Amazon KMS 在分配的时间内没有响应，连接失败了。如果您反复看到此连接错误代码，请通知您的外部密钥存储代理供应商。有关此连接失败及其他连接错误失败的帮助，请参阅排查外部密钥存储的问题。

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksVpc
{
    "CustomKeyStores": [
    {
      "CustomKeyStoreId": "cks-9876543210fedcba9",
      "CustomKeyStoreName": "ExampleXksVpc",
      "ConnectionState": "FAILED",
      "ConnectionErrorCode": "XKS_PROXY_TIMED_OUT",
      "CreationDate": "2022-12-13T18:34:10.675000+00:00",
      "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE98765432EXAMPLE",
        "Connectivity": "VPC_ENDPOINT_SERVICE",
        "UriEndpoint": "https://example-proxy-uri-endpoint-vpc",
        "UriPath": "/example/prefix/kms/xks/v1",
        "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-example"
      }
    }
  ]
```

```
}
```

监控外部密钥存储

Amazon KMS 收集与外部密钥存储的每个交互的指标，并在 CloudWatch 账户中发布。这些指标用于在每个外部密钥存储的详细信息页面的监控部分生成图表。以下主题详细介绍了如何使用图表来识别和排查影响外部密钥存储的操作和配置问题。我们建议使用 CloudWatch 指标设置警报，以便在外部密钥存储未按预期执行时通知您。有关更多信息，请参阅使用 Amazon CloudWatch 进行监控。

主题

- 查看图表
- 解释图表

查看图表

您可以在不同的详细程度下查看图表。默认情况下，每个图表使用三小时的时间范围和五分钟的汇总期。您可以在控制台中调整图表视图，但是当关闭外部密钥存储的详细信息页面或刷新浏览器时，您的更改将恢复为默认设置。有关 Amazon CloudWatch 术语的帮助，请参阅 Amazon CloudWatch 的概念。

查看数据点详细信息

每个图表中的数据通过 Amazon KMS 指标收集。若要查看有关特定数据点的更多信息，请将鼠标悬停在折线图上的数据点上方。这将显示一个弹出窗口，其中包含有关派生该图表的指标的更多信息。每个列表项都显示记录在该数据点的维度值。如果该数据点的维度值没有可用的指标数据，则弹出窗口将显示空值（–）。有些图表记录了单个数据点的多个维度和值。其他图表，例如可靠性图表，使用指标收集的数据来计算唯一值。每个列表项均与不同的折线图颜色关联。

修改时间范围

若要修改时间范围，请在监控部分的右上角选择一个预定义的时间范围。预定义的时间范围从 1 小时到 1 周（1 小时、3 小时、12 小时、1 天、3 天或者 1 周）。这将调整所有图表的时间范围。如果您想在不同的时间范围内查看一个特定的图表，或者想要设置自定义时间范围，请放大图表或在 Amazon CloudWatch 控制台中查看。

放大图表

您可以使用缩微贴图缩放功能来重点查看折线图和图表的堆叠部分，而无需在放大和缩小视图之间进行切换。例如，您可以使用缩微贴图缩放功能来重点查看折线图中的峰值，以便将该峰值与同一时间线的监控部分中的其他图表进行比较。

1. 选择并拖动要突出的图表区域，然后释放拖动对象。
2. 要重置缩放，选择 Reset zoom（重置缩放）图标，该图标看起来像放大镜里面包含减号 (-) 符号。

放大图表

若要放大图表，请选择单个图表右上角的菜单图标，然后选择 Enlarge（放大）。将鼠标悬停在图表上方时，也可以选择菜单图标旁边显示的放大图标。

放大图表可让您通过指定不同的时间段、自定义时间范围或刷新间隔来进一步修改图表的视图。关闭放大视图时，这些更改将恢复为默认设置。

修改时间段

1. 选择 Period options（时间段选项）菜单。默认情况下，此菜单显示的值为：5 minutes（5 分钟）。
2. 选择一个时间段，预定义的时间段从 1 秒到 30 天不等。

   例如，您可以选择一分钟视图，这在您排查问题时会非常有用。或者，选择不太详细的一小时视图。这在您查看更大的时间范围（例如 3 天）来了解一段时间内的趋势时会很有用。有关更多信息，请参阅《Amazon CloudWatch 用户指南》中的时间段。

修改时间范围或时区

1. 选择一个预定义的时间范围，此范围的跨度从 1 小时到 1 周（1 小时、3 小时、12 小时、1 天、3 天或 1 周）。或者，您也可以选择 Custom（自定义）来设置自己的时间范围。
2. 选择 Custom（自定义）。

   a. 时间范围：选择方框左上角的 Absolute（绝对）选项卡。使用日历选取器或文本字段框指定时间范围。

   b. 时区：选择方框右上角的下拉菜单。您可以将时区更改为 UTC（协调世界时）或 Local time zone（本地时区）。
3. 在指定时间范围后，选择 Apply（应用）。

修改图表中数据的刷新频率

1. 选择右上角的 Refresh options（刷新选项）菜单。
2. 选择刷新间隔（关闭、10 秒、1 分钟、2 分钟、5 分钟或 15 分钟）。

在 Amazon CloudWatch 控制台中查看指标

监控部分中的图表派生自 Amazon KMS 发布到 Amazon CloudWatch 的预定义指标。您可以在 CloudWatch 控制台中打开这些图表并将其保存到 CloudWatch 控制面板。如果您有多个外部密钥存储，则可以在 CloudWatch 中打开它们各自的图表，并将图表保存到单个控制面板以比较外部密钥存储的运行状况和使用情况。

添加到 CloudWatch 控制面板

选择右上角的 Add to dashboard（添加到控制面板），将所有图表添加到 Amazon CloudWatch 控制面板。您可以选择现有的控制面板或创建一个新控制面板。有关使用此控制面板创建图表和警报的自定义视图的信息，请参阅《Amazon CloudWatch 用户指南》中的使用 Amazon CloudWatch 控制面板。

在 CloudWatch 指标中查看

选择单个图表右上角的菜单图标，然后选择 View in metrics（在指标中查看）以在 Amazon CloudWatch 控制台中查看此图表。在 CloudWatch 控制台中，您可以将此单个图表添加到控制面板并修改时间范围、时间段和刷新间隔。有关更多信息，请参阅《Amazon CloudWatch 用户指南》中的绘制指标的图表。

解释图表

Amazon KMS 提供了多个图表来监控 Amazon KMS 控制台中的外部密钥存储的运行状况。这些图表为自动配置，派生自 Amazon KMS 指标。

图形数据作为您对外部密钥存储和外部密钥进行的调用的一部分收集。在您未进行任何调用的时间范围内，您可能会看到数据填充图表，此数据来自 Amazon KMS 代表您进行的定期 `GetHealthStatus` 调用，以检查外部密钥存储代理和外部密钥管理器的状态。如果您的图表显示 No data available（无可用数据）消息，则表示在该时间范围内没有记录任何调用，或者您的外部密钥存储处于 `DISCONNECTED` 状态。您可以通过将视图调整到更大的时间范围来确定外部密钥存储断开连接的时间。

主题

- 请求总数
- 可靠性

- 延迟
- 前 5 个异常
- 证书过期天数

请求总数

在给定时间范围内收到的针对特定外部密钥存储的 Amazon KMS 请求总数。使用此图表来确定您是否面临节流的风险。

Amazon KMS 建议您的外部密钥管理器每秒能够处理多达 1800 个加密操作请求。如果您在五分钟内接通 540000 个电话，则有节流的风险。

您可以监控外部密钥存储中 Amazon KMS 使用 ExternalKeyStoreThrottle 指标进行节流的 KMS 密钥的加密操作请求数量。

如果您经常收到 KMSInvalidStateException 错误，其中包含一条说明请求“due to a very high request rate”（由于请求率很高）而被拒绝的消息，则可能表明您的外部密钥管理器或外部密钥存储代理无法跟上当前的请求速率。如可能，请降低您的请求速率。您也可以考虑请求降低自定义密钥存储请求限额值。降低此限额值可能会增加节流的风险，但这表示多余的请求在发送到外部密钥存储代理或外部密钥管理器之前会很快被 Amazon KMS 拒绝。要申请下调限额，请访问 Amazon Web Services 支持 中心 并创建工单。

总请求数图表派生自 XksProxyErrors 指标，该指标收集有关 Amazon KMS 从外部密钥存储代理收到的成功和失败响应的数据。当您查看特定数据点时，弹出窗口会显示 CustomKeyStoreId 维度的值以及在该数据点记录的 Amazon KMS 请求总数。CustomKeyStoreId 将始终相同。

可靠性

外部密钥存储代理返回的成功响应或不可重试错误的 Amazon KMS 请求百分比。使用此图表评估外部密钥存储代理的运行状况。

当图表显示的值小于 100% 时，该值表示代理没有响应，或者响应了可重试错误。这可能表明网络存在问题、外部密钥存储代理或外部密钥管理器速度缓慢或实施错误。

如果请求包含错误的凭证，且代理响应了 AuthenticationFailedException，则该图仍将显示 100% 的可靠性，因为代理在外部密钥存储代理 API 请求中识别出了错误的值，因此预计会出现故障。如果可靠性图表的百分比为 100%，则您的外部密钥存储代理将按预期进行响应。如果图表显示的值小于 100%，则代理会响应可重试错误或超时。例如，如果代理由于请求速率过高而响应了 ThrottlingException，则图表将显示较低的可靠性百分比，因为代理无法识别请求中导致其失败的特定问题。这是因为可重试错误可能是临时性问题，可以通过重试请求解决。

以下错误响应将降低可靠性百分比。您可以使用 前 5 个异常 图表和 XksProxyErrors 指标进一步监控代理返回每个可重试错误的频率。

- `InternalException`
- `DependencyTimeoutException`
- `ThrottlingException`
- `XksProxyUnreachableException`

可靠性图表派生自 XksProxyErrors 指标，该指标收集有关 Amazon KMS 从外部密钥存储代理收到的成功和失败响应的数据。只有当响应的 `ErrorType` 值为 `Retryable` 时，可靠性百分比才会降低。当您查看特定数据点时，弹出窗口会显示 `CustomKeyStoreId` 维度的值以及在该数据点记录的 Amazon KMS 请求的可靠性百分比。`CustomKeyStoreId` 将始终相同。

我们建议使用 XksProxyErrors 指标创建 CloudWatch 警报，以在一分钟内记录了超过 5 个可重试错误时通知您潜在的网络问题。有关更多信息，请参阅 创建可重试错误警报。

延迟

外部密钥存储代理响应 Amazon KMS 请求所用的毫秒数。使用此图表来评估您的外部密钥存储代理和外部密钥管理器的性能。

Amazon KMS 希望外部密钥存储代理在 250 毫秒内响应每个请求。如果网络超时，Amazon KMS 将重试一次请求。如果代理再次失败，则记录的延迟是两次请求尝试的合计超时限制，图表将显示大约 500 毫秒。在代理未在 250 毫秒超时限制内响应的所有其他情况下，记录的延迟为 250 毫秒。如果代理在加密和解密操作时经常超时，请咨询您的外部代理管理员。有关解决延迟问题的帮助，请参阅 延迟和超时错误。

响应缓慢还可能表明您的外部密钥管理器无法处理当前的请求流量。Amazon KMS 建议您的外部密钥管理器每秒能够处理高达 1800 个加密操作请求。如果您的外部密钥管理器无法处理每秒 1800 个请求的速率，请考虑请求降低自定义密钥存储中 KMS 密钥的请求限额。使用外部密钥存储中的 KMS 密钥进行加密操作的请求将采用快速失效机制，并出现节流异常，而不是由外部密钥存储代理或外部密钥管理器处理后拒绝。

延迟图表派生自 XksProxyLatency 指标。当您查看特定数据点时，弹出窗口会显示相应的 `KmsOperation` 和 `XksOperation` 维度的值以及该数据点的操作记录的平均延迟。列表项按从最高延迟到最低延迟的顺序排列。

我们建议使用 XksProxyLatency 指标创建 CloudWatch 警报，当延迟接近超时限制时，该警报会通知您。有关更多信息，请参阅 创建响应超时警报。

前 5 个异常

在给定时间范围内失败的加密和管理操作的前五个异常。使用此图表跟踪最常见的错误，因此您可以确定工程工作的优先顺序。

此计数包括 Amazon KMS 从外部密钥存储代理收到的异常以及无法与外部密钥存储代理建立通信时 Amazon KMS 在内部返回的 `XksProxyUnreachableException`。

可重试错误率高可能表示网络错误，而不可重试错误率高可能表示外部密钥存储的配置存在问题。例如，`AuthenticationFailedExceptions` 中的峰值表示 Amazon KMS 中配置的身份验证凭证与外部密钥存储代理之间存在差异。若要查看您的外部密钥存储配置，请参阅 查看外部密钥存储。若要编辑您的外部密钥存储设置，请参阅 编辑外部密钥存储属性。

Amazon KMS 从外部密钥库代理收到的异常与 Amazon KMS 在操作失败时返回的异常不同。对于与外部密钥存储的外部配置或连接状态有关的所有操作失败，Amazon KMS 加密操作都会返回 `KMSInvalidStateException`。若要确定问题，请使用随附的错误消息文本。

下表显示了可能出现在前 5 个异常图表中的异常以及 Amazon KMS 向您返回的相应异常。

错误类型	图表中显示的异常	Amazon KMS 向您返回的异常
不可重试	**AccessDeniedException** 有关问题排查帮助，请参阅代理授权问题。	**CustomKeyStoreInvalidStateException** 响应 `CreateKey` 操作。 **KMSInvalidStateException** 响应加密操作。
不可重试	**AuthenticationFailedException** 有关问题排查帮助，请参阅身份验证凭证错误。	**XksProxyIncorrectAuthenticationCredentialException** 响应 `CreateCustomKeyStore` 和 `UpdateCustomKeyStore` 操作。 **CustomKeyStoreInvalidStateException** 响应 `CreateKey` 操作。

错误类型	图表中显示的异常	Amazon KMS 向您返回的异常
		KMSInvalidStateException 响应加密操作。
可重试	**DependencyTimeoutException** 有关问题排查帮助，请参阅延迟和超时错误。	**XksProxyUriUnreachableException** 响应 CreateCustomKeyStore 和 UpdateCustomKeyStore 操作。 **CustomKeyStoreInvalidStateException** 响应 CreateKey 操作。 **KMSInvalidStateException** 响应加密操作。
可重试	**InternalException** 外部密钥库代理拒绝了该请求，因为该请求无法与外部密钥管理器通信。验证外部密钥存储代理配置是否正确以及外部密钥管理器是否可用。	**XksProxyInvalidResponseException** 响应 CreateCustomKeyStore 和 UpdateCustomKeyStore 操作。 **CustomKeyStoreInvalidStateException** 响应 CreateKey 操作。 **KMSInvalidStateException** 响应加密操作。
不可重试	**InvalidCiphertextException** 有关问题排查帮助，请参阅解密错误。	**KMSInvalidStateException** 响应加密操作。

错误类型	图表中显示的异常	Amazon KMS 向您返回的异常
不可重试	**InvalidKeyUsageException** 有关问题排查帮助，请参阅外部密钥的加密操作错误。	**XksKeyInvalidConfigurationException** 响应 CreateKey 操作。 **KMSInvalidStateException** 响应加密操作。
不可重试	**InvalidStateException** 有关问题排查帮助，请参阅外部密钥的加密操作错误。	**XksKeyInvalidConfigurationException** 响应 CreateKey 操作。 **KMSInvalidStateException** 响应加密操作。
不可重试	**InvalidUriPathException** 有关问题排查帮助，请参阅常规配置错误。	**XksProxyInvalidConfigurationException** 响应 CreateCustomKeyStore 和 UpdateCustomKeyStore 操作。 **CustomKeyStoreInvalidStateException** 响应 CreateKey 操作。 **KMSInvalidStateException** 响应加密操作。
不可重试	**KeyNotFoundException** 有关问题排查帮助，请参阅外部密钥错误。	**XksKeyNotFoundException** 响应 CreateKey 操作。 **KMSInvalidStateException** 响应加密操作。

错误类型	图表中显示的异常	Amazon KMS 向您返回的异常
可重试	**ThrottlingException** 由于请求速率过高，外部密钥存储代理拒绝了该请求。使用外部密钥存储中的 KMS 密钥降低调用频率。	**XksProxyUriUnreachableException** 响应 CreateCustomKeyStore 和 UpdateCustomKeyStore 操作。 **CustomKeyStoreInvalidStateException** 响应 CreateKey 操作。 **KMSInvalidStateException** 响应加密操作。
不可重试	**UnsupportedOperationException** 有关问题排查帮助，请参阅外部密钥的加密操作错误。	**XksKeyInvalidResponseException** 响应 CreateKey 操作。 **KMSInvalidStateException** 响应加密操作。
不可重试	**ValidationException** 有关问题排查帮助，请参阅代理问题。	**XksProxyInvalidResponseException** 响应 CreateCustomKeyStore 和 UpdateCustomKeyStore 操作。 **CustomKeyStoreInvalidStateException** 响应 CreateKey 操作。 **KMSInvalidStateException** 响应加密操作。

错误类型	图表中显示的异常	Amazon KMS 向您返回的异常
可重试	**XksProxyUnreachableException** 如果您反复看到此错误，请验证您的外部密钥存储代理是否处于活动状态并已连接到网络，以及其在外部密钥存储中的 URI 路径和端点 URI 或 VPC 服务名称是否正确。	**XksProxyUriUnreachableException** 响应 `CreateCustomKeyStore` 和 `UpdateCustomKeyStore` 操作。 **CustomKeyStoreInvalidStateException** 响应 `CreateKey` 操作。 **KMSInvalidStateException** 响应加密操作。

前 5 个异常图表派生自 XksProxyErrors 指标。当您查看特定数据点时，弹出窗口会显示 `ExceptionName` 维度的值以及在该数据点记录的异常的次数。五个列表项按最常见的异常到最少见的异常顺序排列。

我们建议使用 XksProxyErrors 指标创建 CloudWatch 警报，以在一分钟内记录了超过 5 个可重试错误时通知您潜在的配置问题。有关更多信息，请参阅 创建不可重试错误警报。

证书过期天数

距离您的外部密钥存储代理端点（`XksProxyUriEndpoint`）的 TLS 证书过期的天数。使用此图表监控即将过期的 TLS 证书。

证书过期后，Amazon KMS 将无法与外部密钥存储代理通信。在您续订证书之前，外部密钥存储中所有受 KMS 密钥保护的数据都将不可访问。

距离证书过期的天数图表派生自 XksProxyCertificateDaysToExpire 指标。我们强烈建议使用此指标创建 CloudWatch 警报，该警报会通知您关于即将过期的信息。证书过期可能会影响您访问加密资源。设置警报，让您的组织有时间在证书过期之前续订证书。有关更多信息，请参阅 创建证书到期警报。

连接和断开外部密钥存储

新外部密钥存储未连接。要在外部密钥存储中创建和使用 Amazon KMS keys，您需要将外部密钥存储连接到其外部密钥存储代理。您可以随时连接和断开外部密钥存储，并查看其连接状态。

在外部密钥存储断开后，Amazon KMS 无法与外部密钥存储代理通信。因此，您可以查看和管理外部密钥存储及其现有 KMS 密钥。不过，您不能在外部密钥存储中创建 KMS 密钥，也不能在加密操作中使用其 KMS 密钥。您可能需要在某些时候断开外部密钥存储的连接，例如编辑外部密钥存储的属性时，但要进行相应计划。断开密钥存储的连接可能会中断使用其 KMS 密钥的 Amazon 服务的运行。

您无需连接外部密钥存储。您可以将外部密钥存储保持在无限期断开的状态并且仅在您需要使用它时进行连接。但是，您可能希望定期测试连接以验证设置是否正确以及自定义密钥存储是否可以连接。

当您断开自定义密钥存储时，密钥存储中的 KMS 密钥立即变得不可用（视最终一致性而定）。不过，在再次使用 KMS 密钥（例如解密数据密钥）之前，使用受 KMS 密钥保护的数据密钥加密的资源不会受到影响。此问题会影响 Amazon Web Services 服务，因为许多服务使用数据密钥来保护您的资源。有关更多信息，请参阅 不可用的 KMS 密钥如何影响数据密钥。

> **Note**
>
> 仅当密钥存储从未连接或您明确断开密钥存储连接时，外部密钥存储才会具有 DISCONNECTED 状态。CONNECTED 状态并不表示外部密钥存储或其支持组件正在高效运行。有关外部密钥存储组件性能的信息，请参阅每个外部密钥存储详细信息页面 Monitoring（监控）部分中的图表。有关更多信息，请参阅 监控外部密钥存储。
>
> 您的外部密钥管理器可能会提供其他方法来停止和重新启动 Amazon KMS 外部密钥存储代理与外部密钥存储代理之间或外部密钥存储代理与外部密钥管理器之间的通信。有关详细信息，请参阅外部密钥管理器的文档。

主题

连接状态

连接和断开会改变自定义密钥存储的连接状态。Amazon CloudHSM 密钥存储和外部密钥存储的连接状态值相同。

要查看自定义密钥存储的连接状态，请使用 DescribeCustomKeyStores 操作或 Amazon KMS 控制台。连接状态显示在每个自定义密钥存储表中、每个自定义密钥存储详细信息页面的 General configuration（常规配置）部分中，以及自定义密钥存储中 KMS 密钥 Cryptographic configuration（加

密配置）选项卡上。有关详细信息，请参阅 查看 Amazon CloudHSM 密钥存储 和 查看外部密钥存储。

自定义密钥存储可能具有以下连接状态之一：

- CONNECTED：自定义密钥存储已连接到其备用密钥存储。您可以在自定义密钥存储中创建和使用 KMS 密钥。

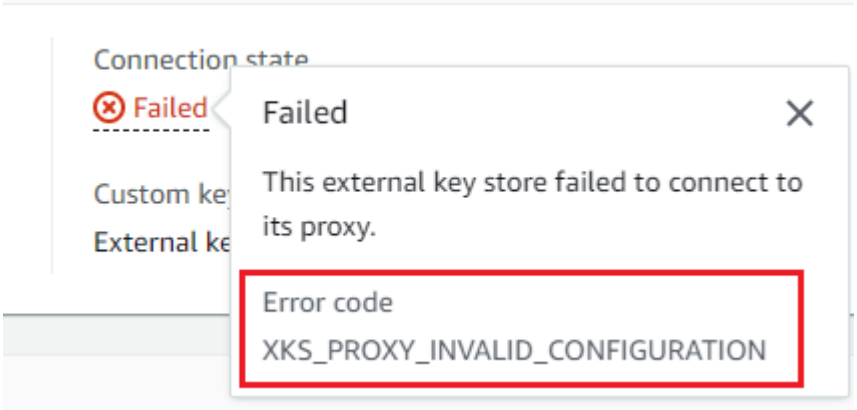
  Amazon CloudHSM 密钥存储的备用密钥存储是其关联的 Amazon CloudHSM 集群。外部密钥存储的备用密钥存储是外部密钥存储代理及其支持的外部密钥管理器。

  CONNECTED（已连接）状态表示连接成功且自定义密钥存储未被故意断开。但该状态并不表示连接运行正常。有关与 Amazon CloudHSM 密钥存储关联的 Amazon CloudHSM 集群状态的信息，请参阅《Amazon CloudHSM User Guide》中的 Getting CloudWatch metrics for Amazon CloudHSM。有关外部密钥存储的状态和操作的信息，请参阅每个外部密钥存储详细信息页面 Monitoring（监控）部分的图表。有关更多信息，请参阅 监控外部密钥存储。
- CONNECTING：连接自定义密钥存储的过程正在进行。这是一种暂时状态。
- DISCONNECTED：自定义密钥存储从未与其后端连接过，或者是使用 Amazon KMS 控制台或 DisconnectCustomKeyStore 操作故意将其断开连接。
- DISCONNECTING：断开自定义密钥存储的过程正在进行。这是一种暂时状态。
- FAILED：尝试连接自定义密钥存储失败。DescribeCustomKeyStores 响应中的 ConnectionErrorCode 表示存在问题。

要连接自定义密钥存储，其连接状态必须为 DISCONNECTED。如果连接状态为 FAILED，则使用 ConnectionErrorCode 来识别和解决问题。接着断开自定义密钥存储，然后再尝试重新连接。如需帮助解决连接失败问题，请参阅 外部密钥存储连接错误。有关响应连接错误代码的帮助信息，请参阅 外部密钥存储的连接错误代码。

要查看连接错误代码，请执行以下操作：

- 在 DescribeCustomKeyStores 响应中查看 ConnectionErrorCode 元素的值。只有当 ConnectionState 为 FAILED 时，此元素才会出现在 DescribeCustomKeyStores 响应中。
- 要在 Amazon KMS 控制台中查看连接错误代码，请将鼠标悬停在外部密钥存储详细信息页面的 Failed（失败）值上。



连接外部密钥存储

外部密钥存储连接到其外部密钥存储代理后，您可以在外部密钥存储中创建 KMS 密钥，然后在加密操作中使用现有 KMS 密钥。

将外部密钥存储连接到其外部密钥存储代理的过程因外部密钥存储的连接而异。

- 当您使用公有端点连接来连接外部密钥存储时，Amazon KMS 会向外部密钥存储代理发送 GetHealthStatus 请求来验证代理 URI 端点、代理 URI 路径和代理身份验证凭证。来自代理的成功响应可确认代理 URI 端点和代理 URI 路径准确且可访问，并且代理对使用外部密钥存储的代理身份验证凭证签名的请求进行了身份验证。
- 在使用 VPC 端点服务连接将外部密钥存储连接到其外部密钥存储代理时，Amazon KMS 会执行以下操作：
  - 确认代理 URI 端点中指定的私有 DNS 名称的域名已通过验证。
  - 创建从 Amazon KMS VPC 到您 VPC 端点服务的接口端点。
  - 为代理 URI 端点中指定的私有 DNS 名称创建私有托管区
  - 向外部密钥存储代理发送 GetHealthStatus 请求。来自代理的成功响应可确认代理 URI 端点和代理 URI 路径准确且可访问，并且代理对使用外部密钥存储的代理身份验证凭证签名的请求进行了身份验证。

连接操作即开始连接您的自定义密钥存储的过程，但是将外部密钥存储连接到其外部代理大约需要五分钟。连接操作的成功响应并不表示外部密钥存储已连接。要确认连接已成功，请使用 Amazon KMS 控制台或 DescribeCustomKeyStores 操作查看外部密钥存储的连接状态。

当连接状态为 FAILED 时，连接错误代码会显示在 Amazon KMS 控制台中并添加到 DescribeCustomKeyStore 响应中。有关解释连接错误代码的帮助信息，请参阅 外部密钥存储的连接错误代码。

连接和重新连接到您的外部密钥存储

您可以在 Amazon KMS 控制台中或使用 ConnectCustomKeyStore 操作连接或重新连接您的外部密钥存储。

使用 Amazon KMS 控制台

您可以使用 Amazon KMS 控制台将外部密钥存储连接到其外部密钥存储代理。

1. 登录到 Amazon Web Services 管理控制台，然后通过以下网址打开 Amazon Key Management Service（Amazon KMS）控制台：https://console.aws.amazon.com/kms。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择 Custom key stores（自定义密钥存储）、External key stores（外部密钥存储）。
4. 选择要连接的外部密钥存储的行。

   如果外部密钥存储的连接状态为 FAILED（失败），则必须在连接之前断开外部密钥存储。
5. 从 Key store actions（密钥存储操作）菜单中选择 Connect（连接）。

该连接过程通常需要五分钟才能完成。操作完成后，连接状态更改为 CONNECTED（已连接）。

如果连接状态为 Failed（失败），请将鼠标悬停在连接状态上方以查看连接错误代码，从中了解错误的原因。有关响应连接错误代码的帮助信息，请参阅 外部密钥存储的连接错误代码。要连接处于 Failed（失败）连接状态的外部密钥存储，必须先断开自定义密钥存储。

使用 Amazon KMS API

要连接已断开的外部密钥存储，请使用 ConnectCustomKeyStore 操作。

在连接之前，外部密钥存储的连接状态必须为 DISCONNECTED。如果连接状态为 FAILED，请断开外部密钥存储，再进行连接。

该连接过程可能需要五分钟才能完成。除非该过程迅速失败，否则 ConnectCustomKeyStore 将返回 HTTP 200 响应和无属性的 JSON 对象。但是，此初始响应不指示连接是否成功。要确定外部密钥存储是否已连接，请参阅 DescribeCustomKeyStores 响应中的连接状态。

本部分中的示例使用 Amazon Command Line Interface (Amazon CLI)，但您可以使用任何受支持的编程语言。

要确定外部密钥存储，请使用自定义密钥存储 ID。您可以在控制台的 Custom key stores（自定义密钥存储）页面上或使用 DescribeCustomKeyStores 操作找到该 ID。在运行此示例之前，请将示例 ID 替换为有效的 ID。

```
$ aws kms connect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

ConnectCustomKeyStore 操作不会在响应中返回 ConnectionState。要验证外部密钥存储是否已连接，请使用 DescribeCustomKeyStores 操作。默认情况下，此操作将返回您的账户和区域中的所有自定义密钥存储。但您可以使用 CustomKeyStoreId 或 CustomKeyStoreName 参数（但不能同时使用两者）将响应限制到特定自定义密钥存储。ConnectionState 值为 CONNECTED 表示外部密钥存储已连接到其外部密钥存储代理。

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksVpc
{
    "CustomKeyStores": [
    {
      "CustomKeyStoreId": "cks-9876543210fedcba9",
      "CustomKeyStoreName": "ExampleXksVpc",
      "ConnectionState": "CONNECTED",
      "CreationDate": "2022-12-13T18:34:10.675000+00:00",
      "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE98765432EXAMPLE",
        "Connectivity": "VPC_ENDPOINT_SERVICE",
        "UriEndpoint": "https://example-proxy-uri-endpoint-vpc",
        "UriPath": "/example/prefix/kms/xks/v1",
        "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-example"
      }
    }
  ]
}
```

如果 DescribeCustomKeyStores 响应中的 ConnectionState 值为 FAILED，则该 ConnectionErrorCode 元素指示失败的原因。

在以下示例中，ConnectionErrorCode 的 XKS_VPC_ENDPOINT_SERVICE_NOT_FOUND 值表示 Amazon KMS 找不到用于与外部密钥存储代理通信的 VPC 端点服务。验证

XksProxyVpcEndpointServiceName 是否正确无误，Amazon KMS 服务主体是否是 Amazon VPC 端点服务允许的主体，以及 VPC 端点服务是否不要求接受连接请求。有关响应连接错误代码的帮助信息，请参阅 外部密钥存储的连接错误代码。

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksVpc
{
    "CustomKeyStores": [
    {
      "CustomKeyStoreId": "cks-9876543210fedcba9",
      "CustomKeyStoreName": "ExampleXksVpc",
      "ConnectionState": "FAILED",
      "ConnectionErrorCode": "XKS_VPC_ENDPOINT_SERVICE_NOT_FOUND",
      "CreationDate": "2022-12-13T18:34:10.675000+00:00",
      "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE98765432EXAMPLE",
        "Connectivity": "VPC_ENDPOINT_SERVICE",
        "UriEndpoint": "https://example-proxy-uri-endpoint-vpc",
        "UriPath": "/example/prefix/kms/xks/v1",
        "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-example"
      }
    }
  ]
}
```

断开外部密钥存储

从外部密钥存储代理断开具有 VPC 端点服务连接的外部密钥存储时，Amazon KMS 会删除其与 VPC 端点服务的接口端点，并移除其为支持连接而创建的网络基础设施。具有公有端点连接的外部密钥存储不需要等效流程。此操作不会影响 VPC 端点服务或其支持的任何组件，也不会影响外部密钥存储代理或任何外部组件。

外部密钥存储断开连接后，Amazon KMS 不会向外部密钥存储代理发送任何请求。外部密钥存储的连接状态为 DISCONNECTED。已断开连接的外部密钥存储中的 KMS 密钥处于 UNAVAILABLE 密钥状态（除非处于待删除状态），这表示此类密钥不能用于加密操作。不过，您仍然可以查看和管理外部密钥存储及其现有 KMS 密钥。

已断开连接状态被设计为临时且可逆的状态。您可以随时重新连接外部密钥存储。通常无需重新配置。不过，如果关联的外部密钥存储代理的任何属性在断开连接时发生了变化，例如轮换了代理身份验证凭证，则必须在重新连接之前编辑外部密钥存储设置。

> **Note**
>
> 虽然自定义密钥存储已断开连接，但在自定义密钥存储中创建 KMS 密钥或在加密操作中使用现有 KMS 密钥的所有尝试都将失败。此操作可以阻止用户存储和访问敏感数据。

为了更好地估计断开外部密钥存储的影响，请在外部密钥存储中标识 KMS 密钥，并确定其过去的使用情况。

您可能出于以下原因断开外部密钥存储：

- 编辑其属性。在外部密钥存储处于连接状态时，您可以编辑自定义密钥存储名称、代理 URI 路径和代理身份验证凭证。不过，要编辑代理连接类型、代理 URI 端点或 VPC 端点服务名称，您必须先断开外部密钥存储的连接。有关更多信息，请参阅 编辑外部密钥存储属性。
- 停止 Amazon KMS 和外部密钥存储代理之间的所有通信。您还可以通过禁用端点或 VPC 端点服务来停止 Amazon KMS 与代理之间的通信。此外，您的外部密钥存储代理或密钥管理软件可能会提供其他机制，防止 Amazon KMS 与代理进行通信或防止代理访问外部密钥管理器。
- 禁用外部密钥存储中的所有 KMS 密钥。您可以通过使用 Amazon KMS 控制台或 DisableKey 操作，在外部密钥存储中禁用和重新启用 KMS 密钥。这些操作会快速完成（视最终一致性而定），但一次只针对一个 KMS 密钥。断开外部密钥存储会将外部密钥存储中所有 KMS 密钥的密钥状态更改为 `Unavailable`，这将阻止在任何加密操作中使用这些 KMS 密钥。
- 修复失败的连接尝试。如果连接外部密钥存储的尝试失败（自定义密钥存储的连接状态为 `FAILED`），则必须在尝试再次连接外部密钥存储之前将其断开。

断开您的外部密钥存储

您可以在 Amazon KMS 控制台中或使用 DisconnectCustomKeyStore 操作断开外部密钥存储的连接。

使用 Amazon KMS 控制台

您可以使用 Amazon KMS 控制台将外部密钥存储连接到其外部密钥存储代理。完成此过程大约需要 5 分钟。

1. 登录到 Amazon Web Services 管理控制台，然后通过以下网址打开 Amazon Key Management Service（Amazon KMS）控制台：https://console.aws.amazon.com/kms。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择 Custom key stores（自定义密钥存储）、External key stores（外部密钥存储）。

4. 选择要断开连接的外部密钥存储的行。
5. 从 Key store actions（密钥存储操作）菜单中选择 Disconnect（断开连接）。

当操作完成时，连接状态将从 DISCONNECTING 变为 DISCONNECTED。如果操作失败，则会出现一条错误消息，描述问题并提供有关如何修复它的帮助。如果您需要更多帮助，请参阅外部密钥存储连接错误。

使用 Amazon KMS API

要断开已连接的外部密钥存储，请使用 DisconnectCustomKeyStore 操作。如果此操作成功，则 Amazon KMS 返回 HTTP 200 响应和无属性的 JSON 对象。该过程可能需要五分钟才能完成。要确定外部密钥存储的连接状态，请使用 DescribeCustomKeyStores 操作。

本部分中的示例使用 Amazon Command Line Interface (Amazon CLI)，但您可以使用任何受支持的编程语言。

此示例断开具有 VPC 端点服务连接的外部密钥存储。在运行此示例之前，请将示例自定义密钥存储 ID 替换为有效 ID。

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

要验证外部密钥存储是否已断开，请使用 DescribeCustomKeyStores 操作。默认情况下，此操作将返回您的账户和区域中的所有自定义密钥存储。但您可以使用 CustomKeyStoreId 和 CustomKeyStoreName 参数（但不能同时使用两者）将响应限制到特定自定义密钥存储。DISCONNECTED 的 ConnectionState 值表示此示例外部密钥存储不再连接到其外部密钥存储代理。

```
$ aws kms describe-custom-key-stores --custom-key-store-name ExampleXksVpc
{
    "CustomKeyStores": [
    {
      "CustomKeyStoreId": "cks-9876543210fedcba9",
      "CustomKeyStoreName": "ExampleXksVpc",
      "ConnectionState": "DISCONNECTED",
      "CreationDate": "2022-12-13T18:34:10.675000+00:00",
      "CustomKeyStoreType": "EXTERNAL_KEY_STORE",
      "XksProxyConfiguration": {
        "AccessKeyId": "ABCDE98765432EXAMPLE",
        "Connectivity": "VPC_ENDPOINT_SERVICE",
        "UriEndpoint": "https://example-proxy-uri-endpoint-vpc",
```

```
        "UriPath": "/example/prefix/kms/xks/v1",
        "VpcEndpointServiceName": "com.amazonaws.vpce.us-east-1.vpce-svc-example"
      }
    }
  ]
}
```

删除外部密钥存储

删除外部密钥存储时，Amazon KMS 会从 Amazon KMS 中删除有关外部密钥存储的所有元数据，包括有关其外部密钥存储代理的信息。此操作不会影响外部密钥存储代理、外部密钥管理器、外部密钥或您为支持外部密钥存储而创建的任何 Amazon 资源，例如 Amazon VPC 或 VPC 端点服务。

在删除外部密钥存储之前，必须从密钥存储中删除所有 KMS 密钥，并从其外部密钥存储代理断开密钥存储。否则，尝试删除密钥存储将失败。

删除外部密钥存储是不可逆的操作，但您可以创建新的外部密钥存储并将其与同一个外部密钥存储代理和外部密钥管理器相关联。不过，即使可以访问相同的外部密钥材料，您也无法在外部密钥存储中重新创建对称加密 KMS 密钥。Amazon KMS 在每个 KMS 密钥独有的对称加密文字中包含元数据。此安全功能确保只有加密数据的 KMS 密钥才能解密数据。

与其删除外部密钥存储，不如考虑断开连接。在外部密钥存储断开连接后，您可以管理外部密钥存储及其 Amazon KMS keys，但无法在外部密钥存储中创建或使用 KMS 密钥。您可以随时重新连接外部密钥存储并恢复使用其 KMS 密钥来加密和解密数据。我们不对已断开连接的外部密钥存储代理或其不可用的 KMS 密钥收取任何费用。

您可以在 Amazon KMS 控制台中或使用 DeleteCustomKeyStore 操作删除外部密钥存储。

使用 Amazon KMS 控制台

您可以使用 Amazon KMS 控制台删除外部密钥存储。

1. 登录到 Amazon Web Services 管理控制台，然后通过以下网址打开 Amazon Key Management Service（Amazon KMS）控制台：https://console.aws.amazon.com/kms。
2. 要更改 Amazon Web Services 区域，请使用页面右上角的区域选择器。
3. 在导航窗格中，选择 Custom key stores（自定义密钥存储）、External key stores（外部密钥存储）。
4. 找到表示要删除的外部密钥存储的行。如果外部密钥存储的 Connection state（连接状态）不是 DISCONNECTED（已断开连接），则必须断开外部密钥存储，然后才能将其删除。

5. 从 Key store actions（密钥存储操作）菜单中选择 Delete（删除）。

在操作完成后，会显示一条成功消息，并且外部密钥存储不再显示在密钥存储列表中。如果操作失败，则会显示一条错误消息，描述问题并提供有关如何解决该问题的帮助。如果您需要更多帮助，请参阅排查外部密钥存储的问题。

使用 Amazon KMS API

要删除外部密钥存储，请使用 DeleteCustomKeyStore 操作。如果此操作成功，则 Amazon KMS 返回 HTTP 200 响应和无属性的 JSON 对象。

首先，断开外部密钥存储的连接。在运行此命令之前，请将示例自定义密钥存储 ID 替换为有效 ID。

```
$ aws kms disconnect-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

在断开外部密钥存储后，您可以使用 DeleteCustomKeyStore 操作将其删除。

```
$ aws kms delete-custom-key-store --custom-key-store-id cks-1234567890abcdef0
```

要确认外部密钥存储已删除，请使用 DescribeCustomKeyStores 操作。

```
$ aws kms describe-custom-key-stores

{
    "CustomKeyStores": []
}
```

如果指定的自定义密钥存储名称或 ID 已不存在，Amazon KMS 则会返回 `CustomKeyStoreNotFoundException` 异常。

```
$ aws kms describe-custom-key-stores --custom-key-store-id cks-1234567890abcdef0

An error occurred (CustomKeyStoreNotFoundException) when calling the
 DescribeCustomKeyStore operation:
```

排查外部密钥存储的问题

大多数外部密钥存储问题的解决方案由错误消息指示，该消息由 Amazon KMS 显示，并提供各个异常，或者由连接错误代码指示，在尝试将外部密钥存储连接到其外部密钥存储代理失败时，Amazon KMS 会返回此错误代码。但是，有些问题较为复杂。

在诊断外部密钥存储的问题时，首先需要找到原因。这将缩小补救措施的范围，提高故障排除的效率。

- Amazon KMS – 问题可能出在 Amazon KMS 内部，例如外部密钥存储配置中的值不正确。
- 外部 – 问题可能源于 Amazon KMS 外部，包括外部密钥存储代理、外部密钥管理器、外部密钥或 VPC 端点服务的配置或操作问题。
- 联网：可能是连接或网络问题，例如代理端点、端口、IP 堆栈或私有 DNS 名称或域存在问题。

> **Note**
>
> 当外部密钥存储的管理操作失败时，这些操作会生成几个不同的异常。但是，对于与外部密钥存储的外部配置或连接状态有关的所有操作失败，Amazon KMS 加密操作都会返回 `KMSInvalidStateException`。若要确定问题，请使用随附的错误消息文本。
> 在连接过程完成之前，ConnectCustomKeyStore 操作会很快成功。要确定连接过程是否成功，请查看外部密钥存储的连接状态。如果连接过程失败，则 Amazon KMS 将返回连接错误代码，用于解释原因并提出补救措施建议。

主题

外部密钥存储故障排除工具

Amazon KMS 提供多种工具来帮助您确定和解决外部密钥存储及其密钥的问题。请将这些工具与随外部密钥存储代理和外部密钥管理器提供的工具结合使用。

> **Note**
>
> 您的外部密钥存储代理和外部密钥管理器可能会提供更简单的方法来创建和维护外部密钥存储及其 KMS 密钥。有关详细信息，请参阅外部工具的相关文档。

Amazon KMS 异常和错误消息

Amazon KMS 提供有关其遇到的任何问题的详细错误消息。您可以在 Amazon Key Management Service API 参考和 Amazon 开发工具包中找到有关 Amazon KMS 异常的更多信息。即使您使用的是 Amazon KMS 控制台，这些参考文献也很有帮助。例如，您可以参阅 `CreateCustomKeyStores` 操作的错误列表。

要优化外部密钥存储代理的性能，Amazon KMS 在给定的 5 分钟聚合期内，根据代理的可靠性返回异常。如果出现 500 内部服务器错误、503 服务不可用或连接超时，可靠性高的代理会返回 `KMSInternalException` 并触发自动重试，以确保请求最终成功。但是，可靠性较低的代理会返回 `KMSInvalidStateException`。有关更多信息，请参阅监控外部密钥存储。

如果问题出现在其他 Amazon 服务中，例如当您在外部密钥存储中使用 KMS 密钥来保护其他 Amazon 服务中的资源时，该 Amazon 服务可能会提供其他信息来帮助您确定问题。如果 Amazon 服务未提供消息，则可以在记录 KMS 密钥使用情况的 CloudTrail 日志中查看错误消息。

CloudTrail 日志

每个 Amazon KMS API 操作，包括 Amazon KMS 控制台中的操作，都记录在 Amazon CloudTrail 日志中。Amazon KMS 会针对成功和失败操作记录日志条目。对于失败的操作，日志条目包括 Amazon KMS 异常名称（`errorCode`）和错误消息（`errorMessage`）。您可以使用此信息来确定和解决错误。有关示例，请参阅使用外部密钥存储中的 KMS 密钥解密失败。

日志条目还包括请求 ID。如果请求到达您的外部密钥存储代理，则您可以使用日志条目中的请求 ID，在代理日志中查找相应的请求（前提是您的代理提供日志）。

CloudWatch 指标

Amazon KMS 会记录有关外部密钥存储运行和性能的详细 Amazon CloudWatch 指标，包括延迟、节流、代理错误、外部密钥管理器状态、TLS 证书过期之前的天数，以及所报告的代理身份验证凭证的期限。您可以使用这些指标，针对外部密钥存储和 CloudWatch 警报（此类警报会在即将出现的问题发生之前提醒您）的运行开发数据模型。

> **Important**
>
> Amazon KMS 建议您创建 CloudWatch 警报来监控外部密钥存储指标。这些警报将在问题出现之前提醒您注意问题的早期迹象。

监控图表

Amazon KMS 会在 Amazon KMS 控制台中每个外部密钥存储的详细信息页面上，显示外部密钥存储 CloudWatch 指标的图表。您可以使用图表中的数据来帮助定位错误来源、检测即将出现的问题、建立基准和优化 CloudWatch 警报阈值。有关监控图标解释和使用其数据的详细信息，请参阅监控外部密钥存储。

显示外部密钥存储和 KMS 密钥

Amazon KMS 会在 Amazon KMS 控制台以及对 DescribeCustomKeyStores 和 DescribeKey 操作的响应中显示有关外部密钥存储及其中的 KMS 密钥的详细信息。这些显示信息包括外部密钥存储和 KMS 密钥的特殊字段，其中包含可用于故障排除的信息，例如外部密钥存储的连接状态和与 KMS 密钥关联的外部密钥的 ID。有关更多信息，请参阅 查看外部密钥存储。

XKS 代理测试客户端

Amazon KMS 提供开源测试客户端，可用于验证您的外部密钥存储代理是否符合 Amazon KMS 外部密钥存储代理 API 规范。您可以使用此测试客户端来确定和解决外部密钥存储代理的问题。

配置错误

创建外部密钥存储时，您需要指定构成外部密钥存储配置的属性值，例如代理身份验证凭证、代理 URI 端点、代理 URI 路径和 VPC 端点服务名称。Amazon KMS 在属性值中检测到错误时，操作将失败并返回错误，指示错误值。

通过修复错误的值，可以解决许多配置问题。您可以修复无效的代理 URI 路径或代理身份验证凭证，而无需断开外部密钥存储的连接。有关这些值的定义，包括唯一性要求，请参阅 汇编先决条件。有关更新这些值的说明，请参阅 编辑外部密钥存储属性。

为避免代理 URI 路径和代理身份验证凭证值出错，在创建或更新外部密钥存储时，请将代理配置文件上传到 Amazon KMS 控制台。这是一个基于 JSON 的文件，其中包含由外部密钥存储代理或外部密钥管理器提供的代理 URI 路径和代理身份验证凭证值。您不能将代理配置文件与 Amazon KMS API 操作结合使用，但您可以使用文件中的值来为 API 请求提供与代理中的值相匹配的参数值。

常规配置错误

异
常：CustomKeyStoreInvalidStateException（CreateKey）、KMSInvalidStateException（加密操作）、XksProxyInvalidConfigurationException（管理操作，CreateKey 除外）

连接错误代
码：XKS_PROXY_INVALID_CONFIGURATION、XKS_PROXY_INVALID_TLS_CONFIGURATION

对于使用公有端点连接的外部密钥存储，Amazon KMS 会在您创建和更新外部密钥存储时测试属性值。对于使用 VPC 端点服务连接的外部密钥存储，Amazon KMS 会在您连接和更新外部密钥存储时测试属性值。

> **Note**
>
> 即使尝试将外部密钥存储连接到其外部密钥存储代理失败，异步 ConnectCustomKeyStore 操作也可能会成功。在这种情况下，不会出现异常，但外部密钥存储的连接状态为“Failed（失败）”，并且连接错误代码将说明错误消息。有关更多信息，请参阅 外部密钥存储连接错误。

如果 Amazon KMS 检测到属性值错误，则操作将会失败并返回 XksProxyInvalidConfigurationException 和以下错误消息中的一种。

由于 URI 路径无效，外部密钥存储代理拒绝了该请求。请验证外部密钥存储的 URI 路径，并在必要时进行更新。

- 代理 URI 路径是对代理 API 的 Amazon KMS 请求的基本路径。如果此路径不正确，则对代理的所有请求都将失败。如需查看外部密钥存储的当前代理 URI 路径，请使用 Amazon KMS 控制台或 DescribeCustomKeyStores 操作。如需查找正确的代理 URI 路径，请参阅您的外部密钥存储代理文档。有关更正代理 URI 路径值的帮助，请参阅 编辑外部密钥存储属性。
- 随着外部密钥存储代理或外部密钥管理器的更新，外部密钥存储代理的代理 URI 路径可能会发生变化。有关此类变化的详细信息，请参阅外部密钥存储代理或外部密钥管理器的文档。

XKS_PROXY_INVALID_TLS_CONFIGURATION
Amazon KMS 无法与外部密钥存储代理建立 TLS 连接。请验证 TLS 配置，包括其证书。

- 所有外部密钥存储代理都需要 TLS 证书。TLS 证书必须由支持外部密钥存储的公有证书颁发机构（CA）颁发。有关受支持的 CA 列表，请参阅 Amazon KMS 外部密钥存储代理 API 规范中的 Trusted Certificate Authorities（受信任的证书颁发机构）。
- 对于公有端点连接，TLS 证书上的主题公用名（CN）必须与外部密钥存储代理的代理 URI 端点中的域名相匹配。例如，如果公有端点是 https://myproxy.xks.example.com，即 TLS，则 TLS 证书上的 CN 必须为 `myproxy.xks.example.com` 或 `*.xks.example.com`。
- 对于 VPC 端点服务连接，TLS 证书上的主题公用名（CN）必须与您的 VPC 端点服务的私有 DNS 名称相匹配。例如，如果私有 DNS 名称是 myproxy-private.xks.example.com，则 TLS 证书上的 CN 必须为 `myproxy-private.xks.example.com` 或 `*.xks.example.com`。
- TLS 证书不得过期。要获取 TLS 证书的到期日期，请使用 SSL 工具，例如 OpenSSL。要监控与外部密钥存储关联的 TLS 证书的到期日期，请使用 XksProxyCertificateDaysToExpire CloudWatch 指标。距离您的 TLS 证书过期日期的天数也显示在 Amazon KMS 控制台的 Monitoring（监控）部分中。
- 如果您使用公有端点连接，请使用 SSL 测试工具来测试您的 SSL 配置。TLS 连接错误可能是由不正确的证书链导致。

VPC 端点服务连接配置错误

异常：`XksProxyVpcEndpointServiceNotFoundException`、`XksProxyVpcEndpointServiceInvali`

除了一般连接问题，在创建、连接或更新使用 VPC 端点服务连接的外部密钥存储时，您可能会遇到以下问题。Amazon KMS 会在创建、连接和更新外部密钥存储时，使用 VPC 端点服务连接测试外部密钥存储的属性值。当管理操作由于配置错误而失败时，这些操作会生成以下异常：

```
XksProxyVpcEndpointServiceNotFoundException
```

原因可能是以下之一：

- VPC 端点服务名称不正确。请验证外部密钥存储的 VPC 端点服务名称是否正确，以及是否与外部密钥存储的代理 URI 端点值相匹配。要查找 VPC 端点服务名称，请使用 Amazon VPC 控制台或 DescribeVpcEndpointServices 操作。要查找现有外部密钥存储的 VPC 端点服务名称和代理 URI 端点，请使用 Amazon KMS 控制台或 DescribeCustomKeyStores 操作。有关更多信息，请参阅 查看外部密钥存储。
- VPC 端点服务可能处于与外部密钥存储不同的 Amazon Web Services 区域。请验证 VPC 端点服务和外部密钥存储是否处于同一区域。（区域名称的外部名称，例如 `us-east-1`，是 VPC 端点服务

名称的一部分，例如 com.amazonaws.vpce.us-east-1.vpce-svc-example。）有关对外部密钥存储的 VPC 端点服务的要求列表，请参阅 VPC 终端节点服务。您无法将 VPC 端点服务或外部密钥存储移至其他区域。但是，您可以在 VPC 端点服务所在的同一区域中创建新的外部密钥存储。有关详细信息，请参阅 配置 VPC 端点服务连接 和 创建外部密钥存储。

- Amazon KMS 不是 VPC 端点服务允许的主体。VPC 端点服务的 Allow principals（允许主体）列表必须包含 `cks.kms.<region>.amazonaws.com` 值，例如 `cks.kms.eu-west-3.amazonaws.com`。有关添加此值的说明，请参阅《Amazon PrivateLink 指南》中的管理权限。

XksProxyVpcEndpointServiceInvalidConfigurationException

当 VPC 端点服务无法满足以下要求之一时，就会出现此错误：

- VPC 需要至少两个私有子网，每个子网均位于不同的可用区内。有关将子网添加到 VPC 中的帮助，请参阅《Amazon VPC 用户指南》中的在您的 VPC 中创建子网。
- 您的 VPC 端点服务类型必须使用网络负载均衡器，而不是网关负载均衡器。
- 不得要求 VPC 端点服务接受请求 [Acceptance required（需要接受）必须是“false”]。如果需要手动接受每个连接请求，则 Amazon KMS 将无法使用 VPC 端点服务连接到外部密钥存储代理。有关详细信息，请参阅《Amazon PrivateLink 指南》中的接受或拒绝连接请求。
- VPC 端点服务必须具有私有 DNS 名称，该名称是公有域的子域。例如，如果私有 DNS 名称为 `https://myproxy-private.xks.example.com`，则 `xks.example.com` 或 `example.com` 域必须具有公有 DNS 服务器。要查看或更改 VPC 端点服务的私有 DNS 名称，请参阅《Amazon PrivateLink 指南》中的管理 VPC 端点服务的 DNS 名称。
- 您的私有 DNS 名称域的 Domain verification status（域验证状态）必须为 `verified`。要查看和更新私有 DNS 名称域的验证状态，请参阅 步骤 5：验证私有 DNS 名称域。添加所需的文本记录后，可能需要几分钟才能显示更新的验证状态。

  > **Note**
  >
  > 只有当私有 DNS 域是公有域的子域时，才能对其进行验证。否则，即使您添加了所需的 TXT 记录，私有 DNS 域的验证状态也不会更改。

- 确保 Amazon KMS 与外部密钥存储代理之间的任何防火墙都允许通过端口 443 流入和流出代理的流量。Amazon KMS 通过端口 443 进行 IPv4 通信。此值不可配置。

- VPC 端点服务的私有 DNS 名称必须与外部密钥存储的代理 URI 端点值相匹配。对于使用 VPC 端点服务连接的外部密钥存储，代理 URI 端点必须是 `https://`，后跟 VPC 端点服务的私有 DNS 名称。要查看代理 URI 端点值，请参阅 查看外部密钥存储。要更改代理 URI 端点值，请参阅 编辑外部密钥存储属性。

外部密钥存储连接错误

将外部密钥存储连接到其外部密钥存储代理的过程大约需要五分钟才能完成。除非该过程迅速失败，否则 `ConnectCustomKeyStore` 操作将返回 HTTP 200 响应和无属性的 JSON 对象。但是，此初始响应不指示连接是否成功。要确定外部密钥存储是否已连接，请查看其连接状态。如果连接失败，外部密钥存储的连接状态将变为 `FAILED`，并且 Amazon KMS 会返回一个 连接错误代码 以解释失败原因。

> **Note**
>
> 当自定义密钥存储的连接状态为 `FAILED` 时，您必须先断开自定义密钥存储的连接，然后再尝试重新连接。您无法连接具有 `FAILED` 连接状态的自定义密钥存储。

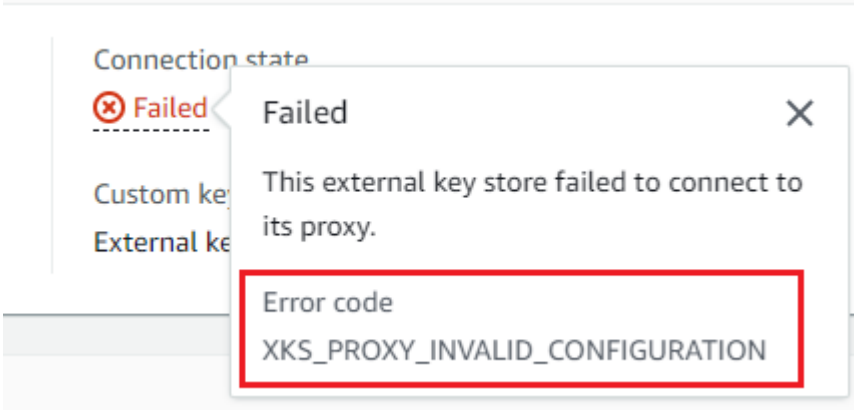
查看外部密钥存储的连接状态：

- 在 DescribeCustomKeyStores 响应中查看 `ConnectionState` 元素的值。
- 在 Amazon KMS控制台中，Connection state（连接状态）显示在外部密钥存储表中。此外，每个外部密钥存储的详细信息页面上的 General configuration（常规配置）部分中也显示了 Connection state（连接状态）。

当连接状态为 `FAILED` 时，连接错误代码有助于解释错误。

要查看连接错误代码，请执行以下操作：

- 在 DescribeCustomKeyStores 响应中查看 `ConnectionErrorCode` 元素的值。只有当 `ConnectionState` 为 `FAILED` 时，此元素才会出现在 `DescribeCustomKeyStores` 响应中。
- 要在 Amazon KMS 控制台中查看连接错误代码，请将鼠标悬停在外部密钥存储详细信息页面的 Failed（失败）值上。



外部密钥存储的连接错误代码

以下连接错误代码适用于外部密钥存储

`INTERNAL_ERROR`

Amazon KMS 因内部错误而无法完成请求。重试请求。对于 `ConnectCustomKeyStore` 请求，先断开自定义密钥存储，然后再尝试重新连接它。

`INVALID_CREDENTIALS`

其中一个或两个 `XksProxyAuthenticationCredential` 值在指定的外部密钥存储代理上无效。

`NETWORK_ERRORS`

网络错误导致 Amazon KMS 无法将自定义密钥存储连接到其备用密钥存储。

`XKS_PROXY_ACCESS_DENIED`

Amazon KMS 请求被拒绝访问外部密钥存储代理。如果外部密钥存储代理有授权规则，请验证这些规则是否允许 Amazon KMS 代表您与代理进行通信。

`XKS_PROXY_INVALID_CONFIGURATION`

配置错误导致外部密钥存储无法连接到其代理。验证 `XksProxyUriPath` 的值。

`XKS_PROXY_INVALID_RESPONSE`

Amazon KMS 无法解析来自外部密钥存储代理的响应。如果您反复看到此连接错误代码，请通知您的外部密钥存储代理供应商。

`XKS_PROXY_INVALID_TLS_CONFIGURATION`

Amazon KMS 无法连接到外部密钥存储代理，因为 TLS 配置无效。请验证外部密钥存储代理是否支持 TLS 1.2 或 1.3。此外，请验证 TLS 证书是否未过期，其是否与 `XksProxyUriEndpoint` 值中的主机名匹配，以及是否由受信任的证书颁发机构列表中包含的受信任的证书颁发机构签名。

`XKS_PROXY_NOT_REACHABLE`

Amazon KMS 无法与您的外部密钥存储代理通信。请验证 `XksProxyUriEndpoint` 和 `XksProxyUriPath` 是否正确。使用外部密钥存储代理的工具来验证该代理是否处于活动状态并在其网络上可用。此外，请验证您的外部密钥管理器实例是否正常运行。如果代理报告所有外部密钥管理器实例都不可用，则连接尝试失败，并显示此连接错误代码。

`XKS_PROXY_TIMED_OUT`

Amazon KMS 可以连接到外部密钥存储代理，但该代理未在分配的时间内响应 Amazon KMS。如果您反复看到此连接错误代码，请通知您的外部密钥存储代理供应商。

`XKS_VPC_ENDPOINT_SERVICE_INVALID_CONFIGURATION`

Amazon VPC 端点服务配置不符合适用于 Amazon KMS 外部密钥存储的要求。

- VPC 端点服务必须是调用方 Amazon Web Services 账户 中的接口端点的端点服务。
- 其必须有至少连接到两个子网的网络负载均衡器（NLB），每个子网位于不同的可用区中。
- `Allow principals` 列表必须包含该区域的 Amazon KMS 服务主体，即 `cks.kms.<region>.amazonaws.com`，例如 `cks.kms.us-east-1.amazonaws.com`。
- 其不得要求接受连接请求。
- 其必须具有私有 DNS 名称。使用 `VPC_ENDPOINT_SERVICE` 连接的外部密钥存储的私有 DNS 名称在其 Amazon Web Services 区域 中必须是唯一的。
- 私有 DNS 名称域的验证状态必须为 `verified`。
- TLS 证书指定了端点可以访问的私有 DNS 主机名。

`XKS_VPC_ENDPOINT_SERVICE_NOT_FOUND`

Amazon KMS 找不到可以用来与外部密钥存储代理通信的 VPC 端点服务。请验证 `XksProxyVpcEndpointServiceName` 是否正确，以及 Amazon KMS 服务主体是否拥有 Amazon VPC 端点服务中的服务使用者权限。

延迟和超时错误

异常：CustomKeyStoreInvalidStateException（CreateKey）、KMSInvalidStateException（加密操作）、XksProxyUriUnreachableException（管理操作）

连接错误代码：XKS_PROXY_NOT_REACHABLE、XKS_PROXY_TIMED_OUT

当 Amazon KMS 无法在 250 毫秒的超时间隔内联系到代理时，将返回异常。CreateCustomKeyStore 和 UpdateCustomKeyStore 会返回 XksProxyUriUnreachableException。加密操作会返回标准 KMSInvalidStateException，其中包含描述问题的错误消息。如果 ConnectCustomKeyStore 失败，则 Amazon KMS 会返回描述问题的连接错误代码。

超时错误可能是暂时性问题，可以通过重试请求来解决。如果问题仍然存在，请验证您的外部密钥存储代理是否处于活动状态并已连接到网络，以及其在外部密钥存储中的代理 URI 端点、代理 URI 路径和 VPC 端点服务名称（如果有）是否正确。此外，请验证您的外部密钥管理器是否靠近外部密钥存储的 Amazon Web Services 区域。如果您需要更新其中任何值，请参阅 编辑外部密钥存储属性。

要跟踪延迟模式，请使用 Amazon KMS 控制台 Monitoring（监控）部分中的 XksProxyLatency CloudWatch 指标和（基于该指标的）Average latency（平均延迟）图表。您的外部密钥存储代理可能还会生成跟踪延迟和超时的日志和指标。

XksProxyUriUnreachableException
Amazon KMS 无法与外部密钥存储代理通信。这可能是暂时的网络问题。如果您反复看到此错误，请验证您的外部密钥存储代理是否处于活动状态并已连接到网络，以及其在外部密钥存储中的端点 URI 是否正确。

- 外部密钥存储代理未在 250 毫秒的超时间隔内响应 Amazon KMS 代理 API 请求。这可能表示出现临时网络问题或代理存在操作或性能问题。如果重试不能解决该问题，请通知您的外部密钥存储代理管理员。

延迟和超时错误通常表现为连接失败。当 ConnectCustomKeyStore 操作失败时，外部密钥存储的连接状态将变为 FAILED，并且 Amazon KMS 会返回一个 连接错误代码 以解释错误。有关连接错误代码的列表和解决错误的建议，请参阅 外部密钥存储的连接错误代码。All custom key stores（所有自定义密钥存储）和 External key stores（外部密钥存储）的连接代码列表适用于外部密钥存储。以下连接错误与延迟和超时有关。

`XKS_PROXY_NOT_REACHABLE`

–或者–

`CustomKeyStoreInvalidStateException` ，`KMSInvalidStateException` ，`XksProxyUriUnreachableException`
Amazon KMS 无法与外部密钥存储代理通信。请验证您的外部密钥存储代理是否处于活动状态并已连接到网络，以及其在外部密钥存储中的 URI 路径和端点 URI 或 VPC 服务名称是否正确。

出现此错误可能的原因如下：

- 外部密钥存储代理未处于活动状态或未连接到网络。
- 外部密钥存储配置中的代理 URI 端点、代理 URI 路径或 VPC 端点服务名称（如果适用）值有错误。要查看外部密钥存储，请使用 DescribeCustomKeyStores 操作，或在 Amazon KMS 控制台中查看外部密钥存储的详细信息页面。
- Amazon KMS 与外部密钥存储代理之间的网络路径上可能存在网络配置错误，例如端口错误。Amazon KMS 通过端口 443 与外部密钥存储代理进行 IPv4 通信。此值不可配置。
- 当外部密钥存储代理（在 GetHealthStatus 响应中）报告所有外部密钥管理器实例均 `UNAVAILABLE` 时，ConnectCustomKeyStore 操作将失败，且 `ConnectionErrorCode` 为 `XKS_PROXY_NOT_REACHABLE`。有关帮助信息，请参阅外部密钥管理器的文档。
- 此错误可能是由外部密钥管理器与外部密钥存储的 Amazon Web Services 区域 之间的物理距离过远所致。Amazon Web Services 区域 与外部密钥管理器之间的 ping 延迟 [网络往返时间（RTT）] 不应超过 35 毫秒。您可能需要在更靠近外部密钥管理器的 Amazon Web Services 区域 中创建外部密钥存储，或者将外部密钥管理器移至离 Amazon Web Services 区域 更近的数据中心。

`XKS_PROXY_TIMED_OUT`

–或者–

`CustomKeyStoreInvalidStateException` ，`KMSInvalidStateException` ，`XksProxyUriUnreachableException`
Amazon KMS 拒绝了该请求，因为外部密钥存储代理没有及时响应。重试请求。如果您反复看到此错误，请将其报告给您的外部密钥存储代理管理员。

出现此错误可能的原因如下：

- 此错误可能是由外部密钥管理器与外部密钥存储代理之间的物理距离过远所致。如可行，请将外部密钥存储代理移到离外部密钥管理器更近的地方。
- 当代理并非专用于处理来自 Amazon KMS 的请求的数量和频率时，可能会出现超时错误。如果您的 CloudWatch 指标表明问题持续存在，请通知您的外部密钥存储代理管理员。
- 当外部密钥管理器与外部密钥存储的 Amazon VPC 之间的连接无法正常运行时，可能会出现超时错误。如果您正在使用 Amazon Direct Connect，请验证您的 VPC 和外部密钥管理器是否可以有效通信。有关解决任何问题的帮助，请参阅《Amazon Direct Connect User Guide》中的 Troubleshooting Amazon Direct Connect。

`XKS_PROXY_TIMED_OUT`

–或者–

`CustomKeyStoreInvalidStateException` ，`KMSInvalidStateException` ，`XksProxyUriUnreachableException`
外部密钥存储代理未在分配的时间内响应请求。重试请求。如果您反复看到此错误，请将其报告给您的外部密钥存储代理管理员。

- 此错误可能是由外部密钥管理器与外部密钥存储代理之间的物理距离过远所致。如可行，请将外部密钥存储代理移到离外部密钥管理器更近的地方。

身份验证凭证错误

异常：`CustomKeyStoreInvalidStateException`（`CreateKey`）、`KMSInvalidStateException`（加密操作）、`XksProxyIncorrectAuthenticationCredentialException`（管理操作，`CreateKey` 除外）

您在外部密钥存储代理中建立和维护 Amazon KMS 身份验证凭证。然后，您在创建外部密钥存储区时告诉 Amazon KMS 凭证值。要更改身份验证凭证，请在外部密钥存储代理中进行更改。然后，针对外部密钥存储更新凭证。如果您的代理轮换凭证，则必须针对外部密钥存储更新凭证。

如果外部密钥存储代理无法针对您的外部密钥存储，对具有代理身份验证凭证签名的请求进行身份验证，则效果取决于请求：

- CreateCustomKeyStore 和 UpdateCustomKeyStore 失败，并显示 XksProxyIncorrectAuthenticationCredentialException。
- ConnectCustomKeyStore 成功，但连接失败。连接状态为 FAILED，连接错误代码为 INVALID_CREDENTIALS。有关更多信息，请参阅 外部密钥存储连接错误。
- 加密操作将返回 KMSInvalidStateException，指出外部密钥存储中的所有外部配置错误和连接状态错误。随附的错误消息描述了问题。

外部密钥存储代理拒绝了该请求，因为其无法对 Amazon KMS 进行身份验证。请验证外部密钥存储的凭证，并在必要时进行更新。

出现此错误可能的原因如下：

- 外部密钥存储的访问密钥 ID 或秘密访问密钥与在外部密钥存储代理上建立的值不匹配。

  要修复此错误，请针对外部密钥存储更新代理身份验证凭证。无需断开外部密钥存储的连接即可进行此更改。
- Amazon KMS 和外部密钥存储代理之间的反向代理可能会操作 HTTP 标头，从而使 SigV4 签名失效。要修复此错误，请通知您的代理管理员。

密钥状态错误

异常：KMSInvalidStateException

KMSInvalidStateException 对自定义密钥存储中的 KMS 密钥有两个不同的用途。

- 当管理操作（例如 CancelKeyDeletion）失败并返回此异常时，这表明 KMS 密钥的密钥状态与该操作不兼容。
- 当对自定义密钥存储中 KMS 密钥的加密操作失败，并显示 KMSInvalidStateException 时，这可能表明 KMS 密钥的密钥状态存在问题。但是，对于外部密钥存储中的所有外部配置错误和连接状态错误，Amazon KMS 加密操作将返回 KMSInvalidStateException。要确定问题，请使用随异常显示的错误消息。

要查找 Amazon KMS API 操作所需的密钥状态，请参阅 Amazon KMS 密钥的密钥状态。要查找 KMS 密钥的密钥状态，请在 Customer managed keys（客户托管密钥）页面上，查看 KMS 密钥的

Status（状态）字段。或者，使用 DescribeKey 操作并查看响应中的 KeyState 元素。有关更多信息，请参阅 识别和查看密钥。

> **Note**
>
> 外部密钥存储中 KMS 密钥的密钥状态并不表明其关联的外部密钥的状态。有关外部密钥状态的信息，请使用外部密钥管理器和外部密钥存储代理工具。CustomKeyStoreInvalidStateException 指的是外部密钥存储的连接状态，而不是 KMS 密钥的密钥状态。

对自定义存储中 KMS 密钥的加密操作可能会失败，因为 KMS 密钥的密钥状态为 Unavailable 或 PendingDeletion。（禁用的密钥将返回 DisabledException。）

- 只有当您在 Amazon KMS 控制台中有意禁用 KMS 密钥或使用 DisableKey 操作禁用 KMS 密钥时，KMS 密钥才会处于 Disabled 密钥状态。当某个 KMS 密钥已禁用时，您可以查看和管理该密钥，但不能在加密操作中使用该密钥。要修复此问题，请启用密钥。有关更多信息，请参阅 启用和禁用密钥。
- 当外部密钥存储与其外部密钥存储代理断开连接时，KMS 密钥会处于 Unavailable 密钥状态。要修复不可用的 KMS 密钥，请重新连接外部密钥存储。重新连接外部密钥存储后，外部密钥存储中的 KMS 密钥的密钥状态将自动还原到之前的状态，例如 Enabled 或 Disabled。

  在计划删除 KMS 密钥，或 KMS 密钥处于等待期时，其会处于 PendingDeletion 密钥状态。待删除的 KMS 密钥的密钥状态错误表明不应删除该密钥，因为其正在用于加密，或者其是解密所必需的密钥。要重新启用 KMS 密钥，请取消计划的删除，然后启用密钥。有关更多信息，请参阅 计划密钥删除。

解密错误

异常：KMSInvalidStateException

当使用外部密钥存储中 KMS 密钥的 解密 操作失败时，对于外部密钥存储的所有外部配置错误和连接状态错误，Amazon KMS 将返回加密操作使用的标准 KMSInvalidStateException。此错误消息指示该问题。

要解密使用 双重加密 进行加密的加密文字，外部密钥管理器应首先使用外部密钥解密加密文字的外层。然后，Amazon KMS 使用 KMS 密钥中的 Amazon KMS 密钥材料解密加密文字的内层。无效或损坏的加密文字可能会被外部密钥管理器或 Amazon KMS 拒绝。

解密失败时，会随 KMSInvalidStateException 出现以下错误消息。它表示请求中的加密文字或可选加密上下文存在问题。

外部密钥存储代理拒绝了该请求，因为指定的加密文字或其他经过身份验证的数据已损坏、丢失或无效。

- 当外部密钥存储代理或外部密钥管理器报告加密文字或其加密上下文无效时，通常表示发送到 Amazon KMS 的 Decrypt 请求中的加密文字或加密上下文存在问题。对于 Decrypt 操作，Amazon KMS 向代理发送与其在 Decrypt 请求中接收的相同的加密文字和加密上下文。

  此错误可能由传输过程中的网络问题引起，例如位反转。重试 Decrypt 请求。如果问题仍然存在，请验证加密文字是否未被更改或损坏。此外，请验证对 Amazon KMS 的 Decrypt 请求中的加密上下文是否与加密数据的请求中的加密上下文相匹配。

外部密钥存储代理提交解密的加密文字或加密上下文已损坏、丢失或无效。

- 当 Amazon KMS 拒绝其从代理收到的加密文字时，这表示外部密钥管理器或代理向 Amazon KMS 返回了无效或损坏的加密文字。

  此错误可能由传输过程中的网络问题引起，例如位反转。重试 Decrypt 请求。如果问题仍然存在，请验证外部密钥管理器是否正常运行，以及外部密钥存储代理在将从外部密钥管理器接收的加密文字返回到 Amazon KMS 之前，是否不会对之进行更改。

外部密钥错误

外部密钥是外部密钥管理器中的加密密钥，用作 KMS 密钥的外部密钥材料。Amazon KMS 无法直接访问外部密钥。其必须要求外部密钥管理器（通过外部密钥存储代理）使用外部密钥来加密数据或解密加密文字。

在外部密钥存储中创建 KMS 密钥时，您可以在其外部密钥管理器中指定外部密钥的 ID。创建 KMS 密钥后，您无法更改外部密钥 ID。为防止 KMS 密钥出现问题，CreateKey 操作会要求外部密钥存储代理验证外部密钥的 ID 和配置。如果外部密钥不符合与 KMS 密钥一起使用的要求，则 CreateKey 操作将失败，并显示说明问题的异常和错误消息。

但是，在创建 KMS 密钥后，可能会出现问题。如果加密操作因外部密钥问题而失败，则此操作将会失败并返回 KMSInvalidStateException，以及一条包含说明问题的错误消息。

外部密钥的 CreateKey 错误

异常：XksKeyAlreadyInUseException、XksKeyNotFoundException、XksKeyInvalidConfigurati

CreateKey 操作会尝试验证您在 External key ID（外部密钥 ID）（控制台）或 XksKeyId（API）参数中提供的外部密钥的 ID 和属性。这种做法是为了在您尝试结合使用 KMS 密钥和外部密钥之前及早发现错误。

正在使用外部密钥

外部密钥存储中的每个 KMS 密钥都必须使用不同的外部密钥。当 CreateKey 识别出 KMS 密钥的外部密钥 ID（XksKeyId）在外部密钥存储中不是唯一时，其将失败并显示 XksKeyAlreadyInUseException。

如果您为同一个外部密钥使用多个 ID，CreateKey 无法识别重复的密钥。但是，具有相同外部密钥的 KMS 密钥不可互操作，因为它们的 Amazon KMS 密钥材料和元数据不同。

未找到外部密钥

当外部密钥存储代理报告无法使用 KMS 密钥的外部密钥 ID（XksKeyId）找到该外部密钥时，CreateKey 操作将会失败并返回 XksKeyNotFoundException，以及以下错误消息。

```
外部密钥存储代理拒绝了该请求，因为其无法找到外部密钥。
```

出现此错误可能的原因如下：

- KMS 密钥的外部密钥（XksKeyId）ID 可能无效。要查找外部密钥代理用于识别外部密钥的 ID，请参阅外部密钥存储代理或外部密钥管理器文档。
- 外部密钥可能已从您的外部密钥管理器中删除。要进行调查，请使用外部密钥管理器工具。如果外部密钥已永久删除，请将其他外部密钥与 KMS 密钥结合使用。有关外部密钥的列表或要求，请参阅 外部密钥存储中 KMS 密钥的要求。

未符合外部密钥要求

当外部密钥存储代理报告外部密钥不 符合要求，无法与 KMS 密钥一起使用时，CreateKey 操作将会失败并返回 XksKeyInvalidConfigurationException，以及如下错误消息中的一条。

外部密钥的密钥规格必须为 AES_256。指定的外部密钥的密钥规格为 *<key-spec>* 。

- 外部密钥必须是 256 位对称加密密钥，密钥规格为 AES_256。如果指定的外部密钥是不同的类型，请指定符合此要求的外部密钥的 ID。

外部密钥的状态必须为“ENABLED”（已启用）。指定外部密钥的状态为 *<status>*。

- 必须在外部密钥管理器中启用外部密钥。如果未启用指定的外部密钥，请使用外部密钥管理器工具将其启用，或指定已启用的外部密钥。

外部密钥的密钥用法必须包括“ENCRYPT”（加密）和“DECRYPT”（解密）。指定的外部密钥的密钥用法是 <*key-usage* >。

- 必须在外部密钥管理器中配置外部密钥以进行加密和解密。如果指定的外部密钥不包括这些操作，请使用外部密钥管理器工具更改操作，或指定其他外部密钥。

外部密钥的加密操作错误

异常：`KMSInvalidStateException`

当外部密钥存储代理找不到与 KMS 密钥关联的外部密钥，或者外部密钥不符合与 KMS 密钥一起使用的要求时，加密操作将失败。

在加密操作期间检测到的外部密钥问题比在创建 KMS 密钥之前检测到的外部密钥问题更难解决。创建 KMS 密钥后，您无法更改外部密钥 ID。如果 KMS 密钥尚未加密任何数据，则可以删除 KMS 密钥并使用其他外部密钥 ID 创建一个新密钥。但是，使用 KMS 密钥生成的加密文字无法由其他任何 KMS 密钥解密，即使是具有相同外部密钥的 KMS 密钥也无法解密，因为密钥具有不同的密钥元数据和不同的 Amazon KMS 密钥材料。相反，应尽可能使用外部密钥管理器工具来解决外部密钥的问题。

当外部密钥存储代理报告外部密钥问题时，加密操作会返回 `KMSInvalidStateException`，以及一条说明该问题的错误消息题。

未找到外部密钥

当外部密钥存储代理报告无法使用 KMS 密钥的外部密钥 ID（XksKeyId）找到外部密钥时，加密操作将返回 KMSInvalidStateException，其中包含以下错误消息。

```
外部密钥存储代理拒绝了该请求，因为其无法找到外部密钥。
```

出现此错误可能的原因如下：

- KMS 密钥的外部密钥（XksKeyId）ID 不再有效。

  要查找与您的 KMS 密钥关联的外部密钥 ID，请查看 KMS 密钥的详细信息。要查找外部密钥代理用于识别外部密钥的 ID，请参阅外部密钥存储代理或外部密钥管理器文档。

  Amazon KMS 在外部密钥存储中创建 KMS 密钥时，会验证外部密钥 ID。但是，ID 可能会失效，尤其是在外部密钥 ID 值是别名或可变名称的情况下。您无法更改与现有 KMS 密钥关联的外部密钥 ID。要对通过 KMS 密钥加密的任何加密文字进行解密，必须将外部密钥与现有外部密钥 ID 重新关联。

  如果您尚未使用 KMS 密钥加密数据，则可以使用有效的外部密钥 ID 创建新的 KMS 密钥。但是，如果您已使用 KMS 密钥生成加密文字，则即使使用相同的外部密钥，也无法使用任何其他 KMS 密钥来对加密文字进行解密。
- 外部密钥可能已从您的外部密钥管理器中删除。要进行调查，请使用外部密钥管理器工具。如果可能，请尝试从外部密钥管理器的副本或备份中恢复密钥材料。如果外部密钥已永久删除，则以关联的 KMS 密钥加密的任何加密文字都将无法恢复。

外部密钥配置错误

当外部密钥存储代理报告外部密钥不符合要求，不能与 KMS 密钥一起使用时，加密操作将返回 KMSInvalidStateException，以及如下错误消息中的一条。

```
外部密钥存储代理拒绝了该请求，因为外部密钥不支持所请求的操作。
```

- 外部密钥必须同时支持加密和解密。如果密钥用法不包括加密和解密，请使用外部密钥管理器工具更改密钥用法。

```
外部密钥存储代理拒绝了该请求，因为该外部密钥未在外部密钥管理器中启用。
```

- 外部密钥必须在外部密钥管理器中启用并处于可用状态。如果外部密钥的状态不是 Enabled，请使用外部密钥管理器工具将其启用。

代理问题

异常：

CustomKeyStoreInvalidStateException（CreateKey）、KMSInvalidStateException（加密操
作）、UnsupportedOperationException、XksProxyUriUnreachableException、XksProxyInva
理操作，CreateKey 除外）

外部密钥存储代理负责调解 Amazon KMS 与外部密钥管理器之间的所有通信。其还将通用 Amazon KMS 请求转换为外部密钥管理器可以解析的格式。如果外部密钥存储代理不符合 Amazon KMS 外部密钥存储代理 API 规范，或者运行不正常或无法与 Amazon KMS 通信，则您将无法在外部密钥存储中创建或使用 KMS 密钥。

尽管许多错误都提到了外部密钥存储代理（因其在外部密钥存储架构中起着关键作用），但这些问题可能源于外部密钥管理器或外部密钥。

本节中的问题与外部密钥存储代理的设计或运行问题有关。要解决这些问题，可能需要更改代理软件。请咨询您的代理管理员。为帮助解决代理问题，Amazon KMS 提供 XKS 代理测试客户端，这是一个开源测试客户端，可用于验证您的外部密钥存储代理是否符合 Amazon KMS 外部密钥存储代理 API 规范。

CustomKeyStoreInvalidStateException 、KMSInvalidStateException 或 XksProxyUriUnreachableException
外部密钥存储代理运行状况不佳。如果您反复看到此消息，请通知您的外部密钥存储代理管理员。

- 此错误可能表示外部密钥存储代理存在运行问题或软件错误。您可以找到生成每个错误的 Amazon KMS API 操作的 CloudTrail 日志条目。重试该操作可能会解决此错误。但是，如果问题仍然存在，请通知您的外部密钥存储代理管理员。
- 当外部密钥存储代理（在 GetHealthStatus 响应中）报告所有外部密钥管理器实例均处于 UNAVAILABLE 状态时，对创建或更新外部密钥存储的尝试会失败，并出现此异常。如果此错误仍然存在，请查阅您的外部密钥管理器文档。

`CustomKeyStoreInvalidStateException` 、`KMSInvalidStateException` 或 `XksProxyInvalidResponseException`
Amazon KMS 无法解析来自外部密钥存储代理的响应。如果您反复看到此错误，请咨询您的外部密钥存储代理管理员。

- 当代理返回 Amazon KMS 无法解析或解释的未定义响应时，Amazon KMS 操作会生成此异常。此错误可能由于暂时的外部问题或偶然的网络错误而偶尔发生。但是，如果此错误持续存在，则可能表明外部密钥存储代理不符合 Amazon KMS 外部密钥存储代理 API 规范。通知您的外部密钥存储管理员或供应商。

`CustomKeyStoreInvalidStateException` 、`KMSInvalidStateException` 或 `UnsupportedOperationException`

外部密钥存储代理拒绝了该请求，因为其不支持所请求的加密操作。

- 外部密钥存储代理应支持 Amazon KMS 外部密钥存储代理 API 规范中定义的所有代理 API。此错误表示代理不支持与请求相关的操作。通知您的外部密钥存储管理员或供应商。

代理授权问题

异常：`CustomKeyStoreInvalidStateException`、`KMSInvalidStateException`

一些外部密钥存储代理针对其外部密钥的使用执行授权要求。允许（但并非必需）外部密钥存储代理设计并实施授权方案，以允许特定用户在特定条件下请求特定操作。例如，代理可能允许用户使用特定的外部密钥进行加密，但不允许使用该外部密钥进行解密。有关更多信息，请参阅 外部密钥存储代理授权（可选）。

代理授权基于 Amazon KMS 包含在其对代理的请求中的元数据。仅当请求来自 VPC 端点且调用者与 KMS 密钥位于同一个账户时，`awsSourceVpc` 和 `awsSourceVpce` 字段才包含在元数据中。

```
"requestMetadata": {
    "awsPrincipalArn": string,
    "awsSourceVpc": string, // optional
    "awsSourceVpce": string, // optional
    "kmsKeyArn": string,
```

```
    "kmsOperation": string,
    "kmsRequestId": string,
    "kmsViaService": string // optional
}
```

当代理由于授权失败而拒绝请求时，相关的 Amazon KMS 操作将会失败。`CreateKey` 会返回 `CustomKeyStoreInvalidStateException`。Amazon KMS 加密操作将返回 `KMSInvalidStateException`。两者都使用以下错误消息：

> 外部密钥存储代理拒绝对该操作的访问。请验证用户和外部密钥是否均已获得执行此操作的授权，然后重试请求。

- 要解决错误，请使用外部密钥管理器或外部密钥存储代理工具来确定授权失败的原因。然后，更新导致未经授权的请求的过程，或使用外部密钥存储代理工具更新授权策略。您无法在 Amazon KMS 中解决此错误。

的安全 Amazon Key Management Service

云安全 Amazon 是重中之重。作为 Amazon 客户，您可以从专为满足大多数安全敏感型组织的要求而构建的数据中心和网络架构中受益。

安全是双方 Amazon 的共同责任。责任共担模式将此描述为云的安全性和云中的安全性：

- 云安全 — Amazon 负责保护在 Amazon 云中运行 Amazon 服务的基础架构。 Amazon 还为您提供可以安全使用的服务。作为的一部分，第三方审计师定期测试和验证我们安全的有效性。要了解适用于 Amazon Key Management Service (Amazon KMS) 的合规计划，请参阅按合规计划划分的划分的范围内的服务。
- 云端安全 -您的责任由您使用的 Amazon 服务决定。此外 Amazon KMS，除了您的配置和使用外 Amazon KMS keys，您还应对其他因素负责，包括数据的敏感性、贵公司的要求以及适用的法律和法规

本文档可帮助您了解在使用时如何应用分担责任模型 Amazon Key Management Service。它向您展示了如何进行配置 Amazon KMS 以满足您的安全和合规性目标。

主题

Amazon Key Management Service 中的数据保护

Amazon Key Management Service 存储和保护您的加密密钥，使其具有高度可用性，同时为您提供强大而灵活的访问控制。

主题

保护密钥材料

默认情况下，Amazon KMS 生成并保护 KMS 密钥的加密密钥材料。此外，Amazon KMS 还为在 Amazon KMS 外部创建和保护的密钥材料提供选项。

保护 Amazon KMS 中生成的密钥材料

默认情况下，当您创建 KMS 密钥时，Amazon KMS 会为该 KMS 密钥生成并保护加密材料。

为了保护 KMS 密钥的密钥材料，Amazon KMS 依靠经过 FIPS 140-3 安全级别 3 级验证的硬件安全模块（HSM）组成的分布式实例集。每个 Amazon KMS HSM 都是专用、独立的硬件设备，旨在提供专用的加密功能，以满足 Amazon KMS 的安全性和可扩展性要求。（在中国区域中，Amazon KMS 使用的 HSM 已经过 OSCCA 认证并符合所有相关的中国法规，但未经过 FIPS 140-3 加密模块验证计划验证。）

在 HSM 中生成 KMS 密钥的密钥材料时，默认情况下会对其进行加密。密钥材料仅在 HSM 易失性存储器中解密，并且仅在加密操作中使用该密钥所需的几毫秒内解密。每当密钥材料未处于活跃使用状态时，都会在 HSM 中对其进行加密，然后传输到高度耐用（99.999999999%）、低延迟的永久存储中，在那里与 HSM 保持分离和隔离。明文密钥材料永远不会离开 HSM 安全边界；并且永远不会写入磁盘或持久性存放在任何存储介质中。（唯一的例外是非对称密钥对的公有密钥，此密钥对不是秘密的。）

Amazon 断言，作为一项基本的安全原则，任何 Amazon Web Services 服务 中任何类型的明文加密密钥材料都不存在人为交互。任何人（包括 Amazon Web Services 服务 操作员）都无法查看、访问或导出明文密钥材料。即使在灾难性故障和灾难恢复事件中，该原则也适用。Amazon KMS 中的明文客户密钥材料仅用于 Amazon KMS FIPS 140-3 认证 HSM 中的密码操作，以响应客户或其代表向服务提出的授权请求。

对于客户托管密钥，创建密钥的 Amazon Web Services 账户 是密钥的唯一且不可转让的拥有者。拥有者账户对控制密钥访问权限的授权策略拥有完全和排他性的控制权。对于 Amazon 托管式密钥，Amazon Web Services 账户 可以完全控制授权向 Amazon Web Services 服务 提出请求的 IAM policy。

保护 Amazon KMS 外部生成的密钥材料

Amazon KMS 提供 Amazon KMS 中生成的密钥材料的替代方案。

自定义密钥存储是一项可选的 Amazon KMS 功能，允许您创建由 Amazon KMS 外部生成和使用的密钥材料提供支持的 KMS 密钥。Amazon CloudHSM 密钥存储中的 KMS 密钥由您控制的 Amazon

CloudHSM 硬件安全模块中的密钥提供支持。这些 HSM 已通过 FIPS 140-2 安全性 3 级或 140-3 安全性 3 级认证。外部密钥存储中的 KMS 密钥由您在 Amazon 外部控制和管理的外部密钥管理器中的密钥支持，例如私有数据中心的物理 HSM。

另一个可选功能可使您为 KMS 密钥导入密钥材料。为了在导入的密钥材料传输到 Amazon KMS 时对其进行保护，您可以使用 Amazon KMS HSM 中生成的 RSA 密钥对中的公钥来加密密钥材料。导入的密钥材料在 Amazon KMS HSM 中进行解密，并使用 HSM 中的对称密钥进行重新加密。与所有 Amazon KMS 密钥材料一样，明文导入的密钥材料绝不会让 HSM 处于未加密状态。但是，提供密钥材料的客户负责密钥材料在 Amazon KMS 之外的安全使用、持久性和维护。

数据加密

Amazon KMS 中的数据由 Amazon KMS keys 及其所代表的加密密钥材料组成。此密钥材料仅在其使用时以明文形式存在于 Amazon KMS 硬件安全模块 (HSM) 中。否则，密钥材料将被加密并存储在持久性存储中。

Amazon KMS 生成为 KMS 密钥生成的密钥材料永远不会使 Amazon KMS HSM 未加密。它不会在任何 Amazon KMS API 操作中导出或传输。多区域密钥除外，此时 Amazon KMS 会使用跨区域复制机制将多区域密钥的密钥材料从一个 Amazon Web Services 区域 中的 HSM 复制到另一个 Amazon Web Services 区域 中的 HSM。有关详细信息，请参阅 Amazon Key Management Service 加密详细信息中的多区域密钥复制过程。

主题

- 静态加密
- 传输中加密

静态加密

Amazon KMS 在兼容 FIPS 140-3 安全 3 级的硬件安全模块（HSM）中为 Amazon KMS keys 生成密钥材料。唯一的例外是中国区域，在这些区域中，Amazon KMS 用于生成 KMS 密钥的 HSM 符合所有相关的中国法规，但未经过 FIPS 140-3 加密模块验证计划验证。密钥材料未使用时，会利用 HSM 密钥进行加密，并写入耐久的持久性存储中。KMS 密钥的密钥材料和保护密钥材料的加密密钥永远不会以明文形式离开 HSM。

KMS 密钥的密钥材料的加密和管理完全由 Amazon KMS 处理。

有关更多详细信息，请参阅 Amazon Key Management Service 加密详细信息中的使用 Amazon KMS keys。

传输中加密

Amazon KMS 为 KMS 密钥生成的密钥材料永远不会在 Amazon KMS API 操作中导出或传输。Amazon KMS 使用密钥标识符来表示 API 操作中的 KMS 密钥。同样，Amazon KMS 自定义密钥存储中 KMS 密钥的密钥材料不可导出，并且永远不会在 Amazon KMS 或 Amazon CloudHSM API 操作中传输。

然而，一些 Amazon KMS API 操作会返回数据密钥。此外，客户可以使用 API 操作来为选定的 KMS 密钥导入密钥材料。

所有 Amazon KMS API 调用必须使用传输层安全性协议（TLS）进行签名和传输。Amazon KMS 需要 TLS 1.2，但建议所有区域使用 TLS 1.3。Amazon KMS 还支持所有区域（中国区域除外）的 Amazon KMS 服务端点的混合后量子 TLS。Amazon KMS 不支持 Amazon GovCloud (US) 中的 FIPS 端点的混合后量子 TLS。对 Amazon KMS 的调用还需要一个现代化的密码套件，该套件支持完美向前保密，这意味着任何密钥（如私有密钥）的泄露都不会影响会话密钥。

如果在通过命令行界面或 API 访问 Amazon 时需要经过 FIPS 140-3 验证的加密模块，请使用 FIPS 端点。要使用标准 Amazon KMS 端点或 Amazon KMS FIPS 端点，客户端必须支持 TLS 1.2 或更高版本。有关可用的 FIPS 端点的更多信息，请参阅《美国联邦信息处理标准（FIPS）第 140-3 版》。有关 Amazon KMS FIPS 端点的列表，请参阅 Amazon Web Services 一般参考 中的 Amazon Key Management Service 端点和限额。

Amazon KMS 服务主机和 HSM 之间的通信在经过身份验证的加密方案中使用椭圆曲线加密 (ECC) 和高级加密标准 (AES) 进行保护。有关更多详细信息，请参阅 Amazon Key Management Service 加密详细信息中的内部通信安全。

互联网络流量隐私

Amazon KMS 支持 Amazon Web Services 管理控制台 以及一组使您能够创建和管理 Amazon KMS keys 并在加密操作中使用它们的 API 操作。

Amazon KMS 支持两种网络连接选项，从您的私有网络到 Amazon。

- Internet 上的 IPsec VPN 连接
- Amazon Direct Connect，该服务通过标准的以太网光纤电缆将您的内部网络链接到 Amazon Direct Connect 位置。

所有 Amazon KMS API 调用必须使用传输层安全性 (TLS) 进行签名和传输。这些调用还需要一个现代化的密码套件，该套件支持完美向前保护。仅允许通过 Amazon 内部网络从已知的 Amazon KMS API 主机向存储 KMS 密钥的密钥材料的硬件安全模块 (HSM) 传输流量。

要在不通过公共互联网发送流量的情况下从您的虚拟私有云（VPC）直接连接到 Amazon KMS，请使用由 Amazon PrivateLink 支持的 VPC 端点。有关更多信息，请参阅 Amazon KMS 通过 VPC 终端节点连接到。

Amazon KMS 还支持对传输层安全 (TLS) 网络加密协议使用混合后量子密钥交换选项。当您连接到 Amazon KMS API 终端节点时，可以结合使用此选项与 TLS。

的身份和访问管理 Amazon Key Management Service

Amazon Identity and Access Management (IAM) 可帮助您安全地控制对 Amazon 资源的访问权限。管理员控制谁可以进行身份验证（登录）和授权（拥有权限）使用 Amazon KMS 资源。有关更多信息，请参阅 将 IAM 策略与配合使用 Amazon KMS。

密钥策略是控制中对 KMS 密钥的访问的主要机制 Amazon KMS。每个 KMS 密钥都必须有一个密钥策略。您可以使用 IAM policy 和授权以及密钥政策来控制对您的 KMS 密钥的访问。有关更多信息，请参阅 KMS 密钥访问权限和权限。

如果您使用的是亚马逊虚拟私有云（亚马逊 VPC），则可以创建一个接口 VPC 终端节点来 Amazon KMS 提供支持Amazon PrivateLink。您还可以使用 VPC 终端节点策略来确定哪些委托人可以访问您的 Amazon KMS 终端节点、他们可以进行哪些 API 调用以及他们可以访问哪个 KMS 密钥。

主题

- Amazon 的托管策略 Amazon Key Management Service
- 将服务关联角色用于 Amazon KMS

Amazon 的托管策略 Amazon Key Management Service

Amazon 托管策略是由创建和管理的独立策略 Amazon。 Amazon 托管策略旨在为许多常见用例提供权限，以便您可以开始为用户、组和角色分配权限。

请记住， Amazon 托管策略可能不会为您的特定用例授予最低权限权限，因为它们可供所有 Amazon 客户使用。我们建议通过定义特定于使用案例的客户管理型策略来进一步减少权限。

您无法更改 Amazon 托管策略中定义的权限。如果 Amazon 更新 Amazon 托管策略中定义的权限，则更新会影响该策略所关联的所有委托人身份（用户、组和角色）。 Amazon 最有可能在启动新的 API 或现有服务可以使用新 Amazon Web Services 服务 的 API 操作时更新 Amazon 托管策略。

有关更多信息，请参阅Amazon 《IAM 用户指南》中的托管策略。

Amazon 托管策略： AWSKeyManagementServicePowerUser

您可以将 AWSKeyManagementServicePowerUser 策略附加到 IAM 身份。

您可以使用 AWSKeyManagementServicePowerUser 托管式策略为您账户中的 IAM 主体授予高级用户的权限。高级用户可以创建 KMS 密钥、使用和管理他们创建的 KMS 密钥，以及查看所有 KMS 密钥和 IAM 身份。具有 AWSKeyManagementServicePowerUser 托管式策略的主体还可以从其他来源获取权限，包括密钥策略、其他 IAM policy 和授权。

AWSKeyManagementServicePowerUser是一项 Amazon 托管 IAM 策略。有关 Amazon 托管策略的更多信息，请参阅 IAM 用户指南中的Amazon 托管策略。

> **Note**
>
> 此策略中特定于 KMS 密钥的权限（例如 kms:TagResource 和 kms:GetKeyRotationStatus）仅在该 KMS 密钥的密钥策略明确允许 Amazon Web Services 账户 使用 IAM policy 以控制对密钥的访问时才有效。要确认权限是否特定于 KMS 密钥，请参阅 Amazon KMS 权限 并在 Resources（资源）列中查找 KMS 密钥的值。
> 此策略授予高级用户对任何 KMS 密钥执行操作的权限，以及允许该操作的密钥策略。对于跨账户权限（例如 kms:DescribeKey 和 kms:ListGrants），这可能包括不可信 Amazon Web Services 账户中的 KMS 密钥。有关详细信息，请参阅 IAM policy 的最佳实践 和 允许其他账户中的用户使用 KMS 密钥。要确认权限是否对其他账户中的 KMS 密钥有效，请参阅 Amazon KMS 权限 并查找 Cross-account use（跨账户使用）列中 Yes（是）的值。
> 为了让委托人能够毫无错误地查看 Amazon KMS 控制台，委托人需要标记：permis GetResources sion，该标签未包含在AWSKeyManagementServicePowerUser策略中。您可以在单独的 IAM policy 中允许此权限。

AWSKeyManagementServicePowerUser 托管 IAM policy 必须包含以下权限。

- 允许主体创建 KMS 密钥。由于此过程包括设置密钥策略，因此高级用户可以授予自己和其他人使用和管理他们创建的 KMS 密钥的权限。
- 允许主体创建和删除所有 KMS 密钥上的别名和标签。更改标签或别名可以允许或拒绝使用和管理 KMS 密钥的权限。有关更多信息，请参阅 ABAC for Amazon KMS。
- 允许主体获取有关所有 KMS 密钥的详细信息，包括其密钥 ARN、加密配置、密钥策略、别名、标签和轮换状态。
- 允许主体列出 IAM 用户、组和角色。

- 此策略不允许主体使用或管理他们未创建的 KMS 密钥。但他们可以更改所有 KMS 密钥上的别名和标签，这可能会允许或拒绝其使用或管理 KMS 密钥的权限。

要查看此策略的权限，请参阅《 Amazon 托管策略参考》AWSKeyManagementServicePowerUser中的。

Amazon 托管策略： AWSServiceRoleForKeyManagementServiceCustomKeyStores

您不能将 `AWSServiceRoleForKeyManagementServiceCustomKeyStores` 附加到自己的 IAM 实体。此策略附加到服务相关角色，该角色 Amazon KMS 允许查看与您的 Amazon CloudHSM 密钥库关联的 Amazon CloudHSM 集群，并创建网络以支持您的自定义密钥库与其 Amazon CloudHSM 集群之间的连接。有关更多信息，请参阅 授权 Amazon KMS 管理 Amazon CloudHSM 和 Amazon 资源 EC2 。

Amazon 托管策略： AWSServiceRoleForKeyManagementServiceMultiRegionKeys

您不能将 `AWSServiceRoleForKeyManagementServiceMultiRegionKeys` 附加到自己的 IAM 实体。此策略会附加到一个向 Amazon KMS 授予相关权限的服务相关角色，以将对多区域主密钥的密钥材料做出的任何更改同步到其副本密钥。有关更多信息，请参阅 授权同步多 Amazon KMS 区域密钥。

Amazon KMSAmazon 托管策略的更新

查看 Amazon KMS 自该服务开始跟踪这些更改以来 Amazon 托管策略更新的详细信息。要获得有关此页面更改的自动提示，请订阅 Amazon KMS 文档历史记录 页面上的 RSS 源。

更改	描述	日期
AWSKeyManagementServiceMultiRegionKeysServiceRolePolicy – 对现有策略的更新	Amazon KMS 在策略版本 v2 的托管策略中添加了声明 ID (`Sid`) 字段。	2024 年 11 月 21 日
AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy - 对现有策略的更新	Amazon KMS 添加了`ec2:DescribeVpcs` `ec2:DescribeNetworkAcls` 、和`ec2:DescribeNetwor`	2023 年 11 月 10 日

更改	描述	日期
	`kInterfaces` 权限以监控包含您的 Amazon CloudHSM 集群的 VPC 中的变化，以便在出现故障时 Amazon KMS 可以提供清晰的错误消息。	
Amazon KMS 已开始跟踪更改	Amazon KMS 开始跟踪其 Amazon 托管策略的更改。	2023 年 11 月 10 日

将服务关联角色用于 Amazon KMS

Amazon Key Management Service 使用 Amazon Identity and Access Management (IAM) 服务相关角色。服务相关角色是一种与之直接关联的 IAM 角色的独特类型。 Amazon KMS服务相关角色由服务定义 Amazon KMS 并包含该服务代表您调用其他 Amazon 服务所需的所有权限。

服务相关角色使设置变得 Amazon KMS 更加容易，因为您不必手动添加必要的权限。 Amazon KMS 定义其服务相关角色的权限，除非另有定义，否则 Amazon KMS 只能担任其角色。定义的权限包括信任策略和权限策略，而且权限策略不能附加到任何其他 IAM 实体。

只有在先删除相关资源后，才能删除服务相关角色。这样可以保护您的 Amazon KMS 资源，因为您不会无意中删除访问资源的权限。

有关支持服务相关角色的其他服务的信息，请参阅使用 IAM 的Amazon 服务并查找服务相关角色列中显示为是的服务。选择是和链接，查看该服务的服务关联角色文档。

要查看有关本主题中所讨论服务相关角色的更新的详细信息，请参阅 Amazon KMSAmazon 托管策略的更新。

主题

- 授权 Amazon KMS 管理 Amazon CloudHSM 和 Amazon 资源 EC2
- 授权同步多 Amazon KMS 区域密钥

授权 Amazon KMS 管理 Amazon CloudHSM 和 Amazon 资源 EC2

要支持您的 Amazon CloudHSM 密钥存储， Amazon KMS 需要获得有关您的 Amazon CloudHSM 集群信息的权限。它还需要权限才能创建将您的 Amazon CloudHSM 密钥库连接到

其 Amazon CloudHSM 集群的网络基础架构。要获得这些权限， Amazon KMS 请在 Amazon Web Services 账户中创建AWSServiceRoleForKeyManagementServiceCustomKeyStores服务相关角色。创建 Amazon CloudHSM 密钥存储库的用户必须拥有允许他们创建服务相关角色的`iam:CreateServiceLinkedRole`权限。

要查看有关AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy托管策略更新的详细信息，请参阅Amazon KMSAmazon 托管策略的更新。

主题

- 关于 Amazon KMS 服务相关角色
- 创建服务相关角色
- 编辑服务相关角色描述
- 删除服务相关角色

关于 Amazon KMS 服务相关角色

服务相关角色是一个 IAM 角色，它授予一项 Amazon 服务代表您调用其他 Amazon 服务的权限。它旨在让您更轻松地使用多种集成 Amazon 服务的功能，而无需创建和维护复杂的 IAM 策略。有关更多信息，请参阅 将服务关联角色用于 Amazon KMS。

对于 Amazon CloudHSM 密钥存储，使用AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy托管策略 Amazon KMS 创建AWSServiceRoleForKeyManagementServiceCustomKeyStores服务相关角色。此策略向该角色授予以下权限：

- cloudhsm: describe* — 检测连接到您的自定义密钥存储 Amazon CloudHSM 库的集群中的更改。
- ec2: CreateSecurityGroup — 在连接 Amazon CloudHSM 密钥存储库以创建安全组时使用，该组允许 Amazon CloudHSM 集群 Amazon KMS 之间的网络流量流动。
- ec2: AuthorizeSecurityGroupIngress — 当您连接 Amazon CloudHSM 密钥存储以允许网络访问包含您的 Amazon CloudHSM 集群的 VPC 时使用。 Amazon KMS
- ec2: CreateNetworkInterface — 在连接 Amazon CloudHSM 密钥库以创建用于 Amazon CloudHSM 集群 Amazon KMS 之间通信的网络接口时使用。
- ec2: RevokeSecurityGroupEgress — 当您连接 Amazon CloudHSM 密钥存储库以从 Amazon KMS 创建的安全组中删除所有出站规则时使用。
- ec2: DeleteSecurityGroup — 用于断开 Amazon CloudHSM 密钥存储的连接，以删除连接 Amazon CloudHSM 密钥库时创建的安全组。

- ec2: DescribeSecurityGroups — 用于监控在包含您的 Amazon CloudHSM 集群的 VPC 中 Amazon KMS 创建的安全组中的更改，以便在出现故障时 Amazon KMS 可以提供清晰的错误消息。
- ec2: DescribeVpcs — 用于监控包含您的 Amazon CloudHSM 集群的 VPC 中的更改， Amazon KMS 以便在出现故障时提供清晰的错误消息。
- ec2: DescribeNetworkAcls — 用于监控包含您的 Amazon CloudHSM 集群的 VPC 的网络 ACLs变化， Amazon KMS 以便在出现故障时提供清晰的错误消息。
- ec2: DescribeNetworkInterfaces — 用于监控在包含您的 Amazon CloudHSM 集群的 VPC 中 Amazon KMS 创建的网络接口的变化，以便在出现故障时 Amazon KMS 可以提供清晰的错误消息。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "cloudhsm:Describe*",
        "ec2:CreateNetworkInterface",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:CreateSecurityGroup",
        "ec2:DescribeSecurityGroups",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:DeleteSecurityGroup",
        "ec2:DescribeVpcs",
        "ec2:DescribeNetworkAcls",
        "ec2:DescribeNetworkInterfaces"
      ],
      "Resource": "*"
    }
  ]
}
```

由于AWSServiceRoleForKeyManagementServiceCustomKeyStores服务相关角色仅受信任`cks.kms.amazonaws.com`，因此 Amazon KMS 只能担任此服务相关角色。此角色仅限于查看您的 Amazon CloudHSM 集群以及将 Amazon CloudHSM 密钥库连接到其关联 Amazon CloudHSM 集群

Amazon KMS 所需的操作。它不提供 Amazon KMS 任何其他权限。例如， Amazon KMS 无权创建、管理或删除您的 Amazon CloudHSM 集群或备份。 HSMs

区域

与 Amazon CloudHSM 密钥库功能一样，该AWSServiceRoleForKeyManagementServiceCustomKeyStores角色在所有可用 Amazon Web Services 区域 的地方 Amazon KMS Amazon CloudHSM 都受支持。有关每项服务支持的列表 Amazon Web Services 区域 ，请参阅中的Amazon Key Management Service 终端节点和配额以及Amazon CloudHSM 终端节点和配额Amazon Web Services 一般参考。

有关服务如何使用 Amazon 服务相关角色的更多信息，请参阅 IAM 用户指南中的使用服务相关角色。

创建服务相关角色

Amazon KMS 如果角色尚不存在，则在创建 Amazon CloudHSM 密钥库 Amazon Web Services 账户时会在中自动创建AWSServiceRoleForKeyManagementServiceCustomKeyStores服务相关角色。您无法直接创建或重新创建此服务相关角色。

编辑服务相关角色描述

您无法在AWSServiceRoleForKeyManagementServiceCustomKeyStores 服务相关角色中编辑角色名称或策略语句，但可以编辑角色描述。有关说明，请参阅《IAM 用户指南》中的编辑服务相关角色。

删除服务相关角色

Amazon KMS Amazon Web Services 账户 即使您已经删除了所有 Amazon CloudHSM 密钥库，也不会从中删除AWSServiceRoleForKeyManagementServiceCustomKeyStores服务相关角色。尽管目前没有删除AWSServiceRoleForKeyManagementServiceCustomKeyStores服务相关角色的程序，Amazon KMS 但除非您有有效的 Amazon CloudHSM 密钥存储，否则不要代入该角色或使用其权限。

授权同步多 Amazon KMS 区域密钥

要支持多区域密钥， Amazon KMS 需要权限才能将多区域主键的共享属性与其副本密钥同步。要获得这些权限， Amazon KMS 请在 Amazon Web Services 账户中创建AWSServiceRoleForKeyManagementServiceMultiRegionKeys服务相关角色。创建多区域密钥的用户必须具有能让其创建服务相关角色的 `iam:CreateServiceLinkedRole` 权限。

您可以在 Amazon CloudTrail 日志中查看记录 Amazon KMS 同步共享属性的SynchronizeMultiRegionKey CloudTrail 事件。

要查看有关AWSKeyManagementServiceMultiRegionKeysServiceRolePolicy托管策略更新的详细信息，请参阅Amazon KMSAmazon 托管策略的更新。

主题

- 关于多区域密钥的服务相关角色
- 创建服务相关角色
- 编辑服务相关角色描述
- 删除服务相关角色

关于多区域密钥的服务相关角色

服务相关角色是一个 IAM 角色，它授予一项 Amazon 服务代表您调用其他 Amazon 服务的权限。它旨在让您更轻松地使用多种集成 Amazon 服务的功能，而无需创建和维护复杂的 IAM 策略。

对于多区域密钥，使用AWSKeyManagementServiceMultiRegionKeysServiceRolePolicy托管策略 Amazon KMS 创建AWSServiceRoleForKeyManagementServiceMultiRegionKeys服务相关角色。此策略将授予角色 `kms:SynchronizeMultiRegionKey` 权限，从而允许它同步多区域密钥的共享属性。

由于AWSServiceRoleForKeyManagementServiceMultiRegionKeys服务相关角色仅受信任`mrk.kms.amazonaws.com`，因此 Amazon KMS 只能担任此服务相关角色。此角色仅限于 Amazon KMS 需要同步多区域共享属性的操作。它不提供 Amazon KMS 任何其他权限。例如，Amazon KMS 没有创建、复制或删除任何 KMS 密钥的权限。

有关服务如何使用 Amazon 服务相关角色的更多信息，请参阅 IAM 用户指南中的使用服务相关角色。

JSON

```
{
    "Version":"2012-10-17",
    "Statement" : [
        {
            "Sid" : "KMSSynchronizeMultiRegionKey",
            "Effect" : "Allow",
            "Action" : [
                "kms:SynchronizeMultiRegionKey"
            ],
            "Resource" : "*"
        }
    ]
```

```
}
```

创建服务相关角色

Amazon KMS 如果您创建多区域密钥 Amazon Web Services 账户 时会自动在中创建AWSServiceRoleForKeyManagementServiceMultiRegionKeys服务相关角色（如果该角色尚不存在）。您无法直接创建或重新创建此服务相关角色。

编辑服务相关角色描述

您无法编辑AWSServiceRoleForKeyManagementServiceMultiRegionKeys服务相关角色中的角色名称或策略声明，但可以编辑角色描述。有关说明，请参阅 IAM 用户指南中的编辑服务相关角色。

删除服务相关角色

Amazon KMS 不会从您的中删除AWSServiceRoleForKeyManagementServiceMultiRegionKeys服务相关角色 Amazon Web Services 账户 ，也无法将其删除。但是，除非您的 Amazon Web Services 账户 和区域中有多区域密钥，否则 Amazon KMS 不会代入该AWSServiceRoleForKeyManagementServiceMultiRegionKeys角色或使用其任何权限。

登录和监控 Amazon Key Management Service

监控是了解 Amazon KMS中的 Amazon KMS keys 的可用性、状态和使用情况的重要环节。监控有助于维护 Amazon 解决方案的安全性、可靠性、可用性和性能。 Amazon 提供了多种用于监控您的 KMS 密钥的工具。

Amazon CloudTrail 日志

对 Amazon KMS API 操作的每一次调用都会作为事件捕获 Amazon CloudTrail 到日志中。这些日志记录了来自 Amazon KMS 控制台的所有 API 调用， Amazon KMS 以及和其他 Amazon 服务发出的调用。跨账户 API 调用（例如调用不同 Amazon Web Services 账户账户中的 KMS 密钥）会记录在两个账户的 CloudTrail 日志中。

在进行故障排除或审核时，您可以使用日志重新构建 KMS 密钥的生命周期。您还可以查看其在加密操作中对 KMS 密钥的管理和使用情况。有关更多信息，请参阅 the section called “使用 Amazon CloudTrail 进行日志记录”。

Amazon CloudWatch 日志

监控、存储和访问来自 Amazon CloudTrail 和其他来源的日志文件。有关更多信息，请参阅 Amazon CloudWatch 用户指南。

对于 Amazon KMS， CloudWatch 存储有用的信息，可帮助您防止 KMS 密钥及其保护的资源出现问题。有关更多信息，请参阅 the section called “使用 CloudWatch 监控密钥”。

Amazon EventBridge

Amazon KMS 当您的 KMS 密钥被轮换或删除，或者您的 KMS 密钥中导入的密钥材料过期时，会生成 EventBridge 事件。搜索 Amazon KMS 事件（API 操作）并将其路由到一个或多个目标函数或流，以捕获状态信息。有关更多信息，请参阅the section called “使用 Amazon EventBridge 来监控密钥”和 Amazon EventBridge 用户指南。

亚马逊 CloudWatch 指标

您可以使用指标监控您的 KMS 密钥，这些 CloudWatch 指标收集原始数据并将其处理 Amazon KMS 为性能指标。数据以两周为间隔记录，因此您可以查看当前信息和历史信息的趋势。这有助于您了解 KMS 密钥的使用情况以及它们的使用情况随时间推移的变化。有关使用 CloudWatch 指标监控 KMS 密钥的信息，请参阅Amazon KMS 指标和维度。

亚马逊 CloudWatch 警报

监控您指定的时间段内的某个指标的变化。然后在多个时间段内根据相对于给定阈值的指标值执行操作。例如，您可以创建一个 CloudWatch 警报，当有人尝试使用计划在加密操作中删除的 KMS 密钥时触发该警报。这表示 KMS 密钥仍在使用中，可能不应删除。有关更多信息，请参阅 the section called “创建警报”。

Amazon Security Hub CSPM

您可以使用监控自己的 Amazon KMS 使用情况，以了解安全行业标准和最佳实践的合规性 Amazon Security Hub CSPM。Security Hub CSPM 使用安全控制来评估资源配置和安全标准，以帮助您遵守各种合规框架。有关更多信息，请参阅《Amazon Security Hub CSPM User Guide》中的 Amazon Key Management Service controls。

合规性验证 Amazon Key Management Service

Amazon Key Management Service 作为多个合规计划的一部分，第三方审计师对安全性和 Amazon 合规性进行评估。其中包括 SOC、PCI、FedRAMP、HIPAA 及其他。

主题

- 合规性和安全性文档
- 了解更多

合规性和安全性文档

以下合规和安全文件包括 Amazon KMS。要查看它们，请使用 Amazon Artifact。

- 云计算合规性控制目录 (C5)
- ISO 27001:2013 适用性声明 (SoA)
- ISO 27001:2013 认证
- ISO 27017:2015 适用性声明 (SoA)
- ISO 27017:2015 认证
- ISO 27018:2015 适用性声明 (SoA)
- ISO 27018:2014 认证
- ISO 9001:2015 认证
- PCI DSS 合规证明 (AOC) 和责任摘要
- 服务组织控制 (SOC) 1 报告
- 服务组织控制 (SOC) 2 报告
- 服务组织控制 (SOC) 2 保密性报告
- FedRAMP-高

有关使用的帮助 Amazon Artifact，请参阅在 Artifact 中 Amazon 下载报告。

了解更多

您在使用 Amazon KMS 时的合规责任取决于您的数据的敏感性、贵公司的合规目标以及适用的法律和法规。如果您在使用 Amazon KMS 时必须遵守已发布的标准，请 Amazon 提供资源来帮助：

- Amazon 按合规计划划分的范围内的 Amazon 服务-此页面列出了特定合规计划范围内的服务。有关一般信息，请参阅 Amazon 合规性计划。
- 安全与合规性快速入门指南 — 这些部署指南讨论了架构注意事项，并提供了在上部署以安全性和合规性为重点的基准环境的步骤。 Amazon
- Amazon 合规资源 — 此工作簿和指南集可能适用于您所在的行业和所在地。
- Amazon Config— 该 Amazon 服务评估您的资源配置在多大程度上符合内部实践、行业指导方针和法规。

- Amazon Security Hub CSPM— 此 Amazon 服务可全面了解您的安全状态 Amazon。Security Hub CSPM 使用安全控制来评估您的 Amazon 资源，并检查您的资源是否符合安全行业标准和最佳实践。有关支持的服务和控件的列表，请参阅 Sec urity Hub CSPM 控件参考。

Amazon Key Management Service 中的故障恢复能力

Amazon全球基础设施围绕Amazon Web Services 区域和可用区构建。Amazon Web Services 区域提供多个在物理上独立且隔离的可用区，这些可用区通过延迟低、吞吐量高且冗余性高的网络连接在一起。利用可用区，您可以设计和操作在可用区之间无中断地自动实现失效转移的应用程序和数据库。与传统的单个或多个数据中心基础架构相比，可用区具有更高的可用性、容错性和可扩展性。

除了 Amazon 全球基础设施之外，Amazon KMS 还提供了多种功能，以帮助支持您的数据弹性和备份需求。有关 Amazon Web Services 区域 和可用区的更多信息，请参阅 Amazon 全球基础设施。

区域隔离

Amazon Key Management Service（Amazon KMS）是一种可自我维持的区域性服务，在所有 Amazon Web Services 区域开放。Amazon KMS 采用区域隔离设计，可确保任何一个 Amazon Web Services 区域的可用性问题不会影响在任何其他区域的 Amazon KMS 操作。Amazon KMS 的设计目标是确保零计划停机时间，所有软件更新和扩缩操作都将在不知不觉中无缝执行。

Amazon KMS 服务水平协议（SLA）为所有 KMS API 均提供 99.999% 的服务承诺。为履行这一承诺，Amazon KMS 会确保执行 API 请求所需的所有数据和授权信息在接收该请求的所有区域主机上都可用。

Amazon KMS 基础设施会在每个区域的至少三个可用区（AZ）中复制。为确保多个主机故障不会影响 Amazon KMS 性能，Amazon KMS 旨在服务于来自区域中任何 AZ 的客户流量。

您对 KMS 密钥属性或权限所做的更改将复制到该区域中的所有主机，以确保该区域中的任何主机都能正确处理后续请求。有关使用 KMS 密钥的加密操作请求将会转发给某个 Amazon KMS 硬件安全模块（HSM）队列，其中任何一个模块都可以使用 KMS 密钥执行操作。

多租户设计

Amazon KMS 的多租户设计使其能够达到 99.999% 的可用性 SLA，并保持较高的请求率，同时保护密钥和数据的保密性。

通过部署多个完整性控制执行机制，以确保实际用于执行加密操作的 KMS 密钥始终是您为该操作指定的密钥。

KMS 密钥的明文密钥材料受到全面保护。密钥材料在创建后将立即在 HSM 中加密，并且加密后的密钥材料会立即移动到安全、低延迟的存储中。加密后的密钥仅在使用时才在 HSM 中检索和解密。明文密钥仅在完成加密操作所需的时间内驻留在 HSM 内存中。然后会在 HSM 中重新加密，并将加密后的密钥退回存储。明文密钥材料永远不会离开 HSM；并且永远不会写入持久性存储。

Amazon KMS 中的弹性最佳实践

为了优化 Amazon KMS 资源的弹性，请考虑以下策略。

- 要支持备份和灾难恢复策略，请考虑使用多区域密钥，这是指在一个 Amazon Web Services 区域中创建并且仅复制到您指定区域的 KMS 密钥。借助多区域密钥，您可以在 Amazon Web Services 区域之间（在同一个分区内）移动加密资源而永远不会暴露密钥明文，并且可根据需要在其中的任何一个任何目标区域解密该资源。相关的多区域密钥是可互操作的，因为它们共享相同的密钥材料和密钥 ID，但使用独立的密钥策略以实现高精度访问控制。有关详细信息，请参阅 Amazon KMS 中的多区域密钥。
- 要保护 Amazon KMS 等多租户服务中的密钥，请务必使用访问控制，包括密钥策略和 IAM policy。此外，您可以使用 Amazon PrivateLink 支持的 VPC 接口端点将请求发送至 Amazon KMS。执行此操作时，您的 Amazon VPC 和 Amazon KMS 之间的所有通信将完全位于 Amazon 网络内，使用您的 VPC 限定使用的专用 Amazon KMS 端点进行。您可以使用 VPC 端点策略创建额外的授权层，从而进一步保护这些请求的安全。有关详细信息，请参阅通过 VPC 端点连接到 Amazon KMS。

基础设施安全 Amazon Key Management Service

作为一项托管服务， Amazon Key Management Service (Amazon KMS) 受《Amazon Web Services：安全流程概述》中描述的 Amazon 全球网络安全程序的保护。

要通过 Amazon KMS 网络进行访问，您可以调用 Amazon KMS API 参考中描述的Amazon Key Management Service API 操作。 Amazon KMS 需要 TLS 1.2，建议在所有地区使用 TLS 1.3。Amazon KMS 还支持除中国区域之外的所有地区的 Amazon KMS 服务终端节点的混合后量子 TLS。Amazon KMS 不支持中的 FIPS 端点的混合后量子 TLS。 Amazon GovCloud (US)要使用标准 Amazon KMS 端点或 Amazon KMS FIPS 端点，客户端必须支持 TLS 1.2 或更高版本。客户端还必须支持具有完全向前保密（PFS）的密码套件，例如 Ephemeral Diffie-Hellman（DHE）或 Elliptic Curve Ephemeral Diffie-Hellman（ECDHE）。大多数现代系统（如 Java 7 及更高版本）都支持这些模式。

此外，必须使用访问密钥 ID 和与 IAM 主体关联的秘密访问密钥来对请求进行签名。或者，您可以使用 Amazon Security Token Service（Amazon STS）生成临时安全凭证来对请求进行签名。

您可以从任何网络位置调用这些 API 操作，但 Amazon KMS 支持全局策略条件，允许您根据源 IP 地址、VPC 和 VPC 终端节点控制对 KMS 密钥的访问。您可以在密钥策略和 IAM policy 中使用这些条件键。但是，这些情况可能会 Amazon 阻止您代表您使用 KMS 密钥。有关更多信息，请参阅 Amazon 全局条件键。

例如，以下密钥策略声明允许可以代入该KMSTestRole角色的用户将其 Amazon KMS key 用于指定的加密操作，除非源 IP 地址是策略中指定的 IP 地址之一。

JSON

```
{
  "Version":"2012-10-17",
  "Statement": {
    "Effect": "Allow",
    "Principal": {"AWS":
    "arn:aws:iam::111122223333:role/KMSTestRole"},
    "Action": [
      "kms:Encrypt",
      "kms:Decrypt",
      "kms:ReEncrypt*",
      "kms:GenerateDataKey*",
      "kms:DescribeKey"
    ],
    "Resource": "*",
    "Condition": {
      "NotIpAddress": {
        "aws:SourceIp": [
          "192.0.2.0/24",
          "203.0.113.0/24"
        ]
      }
    }
  }
}
```

物理主机的隔离

所 Amazon KMS 使用的物理基础设施的安全受到《Amazon Web Services：安全流程概述》的 “物理和环境安全” 部分所述的控制措施的约束。您可以在上一节中列出的合规性报告和第三方审计结果中找到更多详细信息。

Amazon KMS 由专用的强化硬件安全模块 (HSMs) 提供支持，该模块具有抵御物理攻击的特定控件。HSMs 它们是没有虚拟化层的物理设备，例如虚拟机管理程序，在多个逻辑租户之间共享物理设备。的密钥材料 Amazon KMS keys 仅存储在上的易失性存储器中 HSMs，并且仅在使用 KMS 密钥时存储。当 HSM 退出操作状态（包括预期和意外关机和重置）时，此内存将被擦除。有关操作的详细信息 Amazon KMS HSMs，请参阅Amazon Key Management Service 加密详细信息。

限额

为使 Amazon KMS 对所有用户具备响应能力和实现较高性能，Amazon KMS 应用了两种配额类型：资源配额和请求配额。对于每个 Amazon Web Services 账户 的每个区域，每个配额均单独计算。

所有 Amazon KMS 限额均可调整，但按需轮换资源限额和Amazon CloudHSM 密钥存储请求限额除外。要请求提高配额，请参阅《服务配额用户指南》中的请求提高配额。要申请减少限额、更改服务限额中未列出的限额，或在 Amazon KMS 服务限额不可用的 Amazon Web Services 区域 中更改限额，请访问 Amazon Web Services 支持 中心并创建案例。

主题

资源配额

Amazon KMS 建立资源配额，以确保它能够为我们的所有客户提供快速而有弹性的服务。某些资源配额仅适用于您创建的资源，而不适用于 Amazon 服务为您创建的资源。如果您使用的资源不属于您的 Amazon Web Services 账户，例如 Amazon 拥有的密钥，那么这些资源不会计入相应配额。

如果已超出资源限制，那么创建该资源类型的其他请求会生成 `LimitExceededException` 错误消息。

除按需轮换 Amazon KMS 资源配额外，所有资源配额均可调整。要请求提高配额，请参阅《服务配额用户指南》中的请求提高配额。要申请减少配额、更改未在 Service Quotas 中列出的配额，或者在没有服务配额 Amazon Web Services 区域 的情况下更改配额，请访问Amazon Web Services 支持 中心并创建案例。 Amazon KMS

下表列出并描述了每个 Amazon Web Services 账户 区域的 Amazon KMS 资源配额。

限额名称	默认 值	适用于	可调整
Amazon KMS keys	100000	客户托管密钥	是
每个 KMS 密钥的别名	10	客户创建的别名	是

限额名称	默认 值	适用于	可调整
每个 KMS 密钥的授权数	50000	客户托管密钥	是
自定义密钥存储资源限额	10	Amazon Web Services 账户 和区域	是
按需轮换	50	客户托管密钥	否

除资源配额外，还 Amazon KMS 使用请求配额来确保服务的响应能力。有关更多信息，请参阅 the section called “请求配额”。

Amazon KMS keys：100000

在您的 Amazon Web Services 账户的每个区域中，您最多可以拥有 100000 个客户托管的密钥。此配额适用于所有 Amazon Web Services 区域 中的所有客户托管的密钥，不考虑其密钥规范或密钥状态。每个 KMS 密钥都视为一个资源。Amazon 托管式密钥 和 Amazon 拥有的密钥 不计入此限额。

每个 KMS 密钥的别名数：50

您最多可以将 50 个别名与每个客户托管密钥关联。 Amazon 关联的别名Amazon 托管式密钥不计入此配额。您在创建或更新别名时可能会遇到此配额。

> **Note**
>
> 仅当 KMS 密钥符合此配额时，kms: ResourceAliases 条件才有效。如果 KMS 密钥超出此配额，则由 `kms:ResourceAliases` 条件授权使用 KMS 密钥的委托人将被拒绝访问 KMS 密钥。有关更多信息，请参阅 由于别名配额而拒绝访问。

每个 KMS 的别名密钥配额取代了限制每个区域中别名总数的每个区域的别名配额。 Amazon Web Services 账户 Amazon KMS 取消了每个区域的别名配额。

每个 KMS 密钥的授权数：50000

每个客户托管密钥最多可以拥有 50000 个授权数，其中包括与 Amazon KMS集成的Amazon 服务所创建的授权。此配额不适用于 Amazon 托管式密钥 或 Amazon 拥有的密钥。

此配额的作用之一是，您不能同时执行超过 50000 个使用相同 KMS 密钥的授权操作。在达到配额之后，您只能在停用或撤消了有效授权时才能在 KMS 密钥上创建新授权。

例如，当您将亚马逊弹性区块存储 (Amazon EBS) 卷附加到亚马逊弹性计算云 (Amazon) 实例时，该卷将被解密，以便您可以读取它。 EC2为获得解密数据的权限，Amazon EBS 将为每个卷创建授权。因此，如果所有 Amazon EBS 卷都使用相同的 KMS 密钥，那么您一次附加的卷不能超过 50000 个。

自定义密钥存储资源限额：10

您最多可以在每个 Amazon Web Services 账户 地区创建 10 个自定义密钥存储库。如果您尝试创建更多内容，则CreateCustomKeyStore操作将失败。

此限额适用于每个账户和区域中的自定义密钥存储总数，包括所有 Amazon CloudHSM 密钥存储和外部密钥存储，无论其连接状态如何。

按需轮换：10

您最多可以对每个 KMS 密钥执行按需轮换 10 次。如果您尝试执行更多按需轮换，则RotateKeyOnDemand操作将失败。

此限额不可调整。您不能通过使用 Service Quotas 或在中创建案例来增加配额 Amazon Web Services 支持。为防止达到按需轮换限额，我们建议尽可能使用自动密钥轮换。

请求配额

Amazon KMS 为每秒请求的 API 操作数量设定配额。请求配额因 API 操作 Amazon Web Services 区域、和其他因素（例如 KMS 密钥类型）而异。当您超过 API 请求配额时， Amazon KMS 会限制该请求。

除Amazon CloudHSM 密钥库 Amazon KMS 请求配额外，所有请求配额均可调整。要请求提高配额，请参阅《服务配额用户指南》中的请求提高配额。要申请减少配额、更改未在 Service Quotas 中列出的配额，或者在没有服务配额 Amazon Web Services 区域 的情况下更改配额，请访问Amazon Web Services 支持 中心并创建案例。 Amazon KMS

如果您超出了GenerateDataKey操作的请求配额，请考虑使用的数据密钥缓存功能 Amazon Encryption SDK。重复使用数据密钥可能会将您的请求频率降至 Amazon KMS.

除了请求配额外，还 Amazon KMS 使用资源配额来确保所有用户的容量。有关更多信息，请参阅 资源配额。

要查看请求速率的趋势，请使用 Service Quotas 控制台。您还可以创建一个 Amazon CloudWatch 警报，当您的请求率达到配额值的特定百分比时，提醒您。有关详情，请参阅Amazon 安全博客 CloudWatch中的使用 Service Quotas 和 Amazon 管理您的 Amazon KMS API 请求速率。

主题

每个 Amazon KMS API 操作的请求配额

下表列出了 S ervice Quotas 配额代码和每个 Amazon KMS 请求配额的默认值。除Amazon CloudHSM 密钥库 Amazon KMS 请求配额外，所有请求配额均可调整。

> **Note**
>
> 您可能需要水平或垂直滚动才能查看此表中的所有数据。

限额名称	默认值（每秒请求数）

`Cryptographic operations (symmetric) request rate`
 	
适用对象：
 	
• `Decrypt`
• `Encrypt`
• `GenerateDataKey`
• `GenerateDataKeyWithoutPlaintext`
• `GenerateMac`
	这些共享配额因请求中使用的 KMS 密钥 Amazon Web Services 区域 和类型而异。每个配额都单独计算。
	
	• 10000（共享）
	• 在以下区域中为 20000（共享）：
	  • 美国东部（俄亥俄），us-east-2
	  • 亚太地区（新加坡），ap-southeast-1
	  • 亚太区域（悉尼），ap-southeast-2
	  • 亚太区域（东京），ap-northeast-1
	  • 欧洲（法兰克福），eu-central-1

限额名称	默认值（每秒请求数）
• GenerateRandom • ReEncrypt • VerifyMac	• 欧洲（伦敦），eu-west-2 • 在以下区域中为 100000（共享）： • 美国东部（弗吉尼亚北部），us-east-1 • 美国西部（俄勒冈），us-west-2 • 欧洲（爱尔兰），eu-west-1
Cryptographic operations (RSA) request rate 适用对象： • Decrypt • Encrypt • ReEncrypt • Sign • Verify	RSA KMS 密钥为 1000（共享）：
Cryptographic operations (ML-DSA) request rate 适用对象： • Sign • Verify	ML-DSA KMS 密钥为 1000（共享）

限额名称	默认值（每秒请求数）
Cryptographic operations (ECC and SM2) request rate 适用对象： • Decrypt—仅支持 SM2（仅限中国区域）KMS 密钥 • DeriveSharedSecret • Encrypt—仅支持 SM2（仅限中国区域）KMS 密钥 • ReEncrypt —仅支持 SM2（仅限中国区域）KMS 密钥 • Sign • Verify	椭圆曲线 (ECC) 和（仅限 SM2 中国区域）KMS 密钥为 1,000（共享）
Custom key store request quotas 适用对象： • Decrypt • DeriveSharedSecret • Encrypt • GenerateDataKey • GenerateDataKeyWithoutPlaintext • GenerateRandom • ReEncrypt	自定义密钥存储限额 针对每个自定义密钥存储单独计算 • 每个 Amazon CloudHSM 密钥库有 1,800（共享） • 每个外部密钥存储 1800 次（共享）
CancelKeyDeletion request rate	5
ConnectCustomKeyStore request rate	5
CreateAlias request rate	5

限额名称	默认值（每秒请求数）
CreateCustomKeyStore request rate	5
CreateGrant request rate	50
CreateKey request rate	5
DeleteAlias request rate	15
DeleteCustomKeyStore request rate	5
DeleteImportedKeyMaterial request rate	15
DescribeCustomKeyStores request rate	5
DescribeKey request rate	2000
DisableKey request rate	5
DisableKeyRotation request rate	5
DisconnectCustomKeyStore request rate	5
EnableKey request rate	5
EnableKeyRotation request rate	15
GenerateDataKeyPair (ECC_NIST_P256) request rate 适用对象： • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	100

限额名称	默认值（每秒请求数）
GenerateDataKeyPair (ECC_NIST_P384) request rate 适用对象： • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	100
GenerateDataKeyPair (ECC_NIST_P521) request rate 适用对象： • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	100
GenerateDataKeyPair (ECC_SECG_P256K1) request rate 适用对象： • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	100
GenerateDataKeyPair (RSA_2048) request rate 适用对象： • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	20

限额名称	默认值（每秒请求数）
GenerateDataKeyPair (RSA_3072) request rate 适用对象： • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	4
GenerateDataKeyPair (RSA_4096) request rate 适用对象： • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	1
GenerateDataKeyPair (SM2 – China Regions only) request rate 适用对象： • GenerateDataKeyPair • GenerateDataKeyPairWithoutPlaintext	25
GetKeyPolicy request rate	1000
GetKeyRotationStatus request rate	1000
GetParametersForImport request rate	0.25（每 4 秒间隔为 1）
GetPublicKey request rate	2000
ImportKeyMaterial request rate	15

限额名称	默认值（每秒请求数）
ListAliases request rate	500
ListGrants request rate	100
ListKeyPolicies request rate	100
ListKeys request rate	500
ListKeyRotations request rate	100
ListResourceTags request rate	2000
ListRetirableGrants request rate	100
PutKeyPolicy request rate	15
ReplicateKey request rate ReplicateKey 操作在主键的区域中计为一个 ReplicateKey 请求，在副本密钥区域中计为两个 CreateKey 请求。其中一个 CreateKey 请求是在创建密钥之前检测潜在问题的排练。	5
RetireGrant request rate	50
RevokeGrant request rate	50
RotateKeyOnDemand request rate	5
ScheduleKeyDeletion request rate	15
TagResource request rate	10
UntagResource request rate	5
UpdateAlias request rate	5
UpdateCustomKeyStore request rate	5

限额名称	默认值（每秒请求数）
UpdateKeyDescription request rate	5
UpdatePrimaryRegion request rate UpdatePrimaryRegion 操作计数为两个 UpdatePrimaryRegion 请求；两个受影响区域中每个区域都有一个请求。	5

应用请求配额

审查请求配额时，请记住以下信息。

- 请求配额同时适用于客户托管密钥和 Amazon 托管式密钥。的使用Amazon 拥有的密钥不计入您的请求配额 Amazon Web Services 账户，即使这些配额用于保护您账户中的资源也是如此。
- 请求配额适用于发送到 FIPS 终端节点和非 FIPS 终端节点的请求。有关 Amazon KMS 服务终端节点的列表，请参阅中的Amazon Key Management Service 终端节点和配额 Amazon Web Services 一般参考。
- 限制基于对区域内所有类型 KMS 密钥的所有请求。此总数包括来自所有委托人的请求 Amazon Web Services 账户，包括代表您的 Amazon 服务部门提出的请求。
- 每个请求配额都是单独计算的。例如，对CreateKey操作的请求不会影响该CreateAlias操作的请求配额。如果 CreateAlias 请求受到限制，CreateKey 请求仍可成功完成。
- 虽然加密操作共享一个配额，但共享配额是独立于其他操作的配额计算的。例如，对 Encrypt 和 Decrypt 操作的调用共享请求配额，但该配额与管理操作的配额无关，例如。EnableKey例如，在欧洲（伦敦）区域，每秒可使用对称 KMS 密钥执行 10000 次加密操作，加上 5 次 EnableKey 操作，而不受限制。

加密操作的共享配额

Amazon KMS 加密操作共享请求配额。您可以请求 KMS 密钥支持的加密操作的任意组合，只要加密操作的总数不超过该 KMS 密钥类型的请求配额。唯一的例外是GenerateDataKeyPair和 GenerateDataKeyPairWithoutPlaintext，它们共享单独的配额。

不同类型的 KMS 密钥的配额单独计算。每个配额适用于在每隔一秒钟的时间间隔内使用给定密钥类型在 Amazon Web Services 账户 和区域中执行这些操作的所有请求。

- 加密操作（对称）请求速率是在账户和区域中使用对称 KMS 密钥进行加密操作的共享请求配额。此配额适用于使用对称加密密钥和 HMAC 密钥的加密操作，这些密钥也是对称的。

  例如，您可能在共享配额为每秒 10,000 个请求的 Amazon Web Services 区域 中使用对称 KMS 密钥。当你每秒发出 7,000 个GenerateDataKey请求和每秒 2,000 个解密请求时， Amazon KMS 不会限制你的请求。但是，如果您每秒发出 9500 个 `GenerateDataKey` 请求和 1000 个加密请求，Amazon KMS 会限制您的请求，因为请求数量超出了共享配额。

  对 自定义密钥存储 中 对称加密 KMS 密钥 的加密操作将会计入账户的加密操作（对称）请求率 和自定义密钥存储的 自定义密钥存储请求限额。

- 加密操作 (RSA) 请求速率 是使用 RSA 非对称 KMS 密钥的加密操作的共享请求配额。

  例如，如果请求配额为每秒 1000 个操作，您可以使用可以加密和解密的 RSA KMS 密钥发出 400 个 Encrypt 请求和 200 个 Decrypt 请求；另外，通过可以签名和验证的 RSA KMS 密钥发出 250 个 Sign 请求和 150 个 Verify 请求。

- 加密操作（ECC）请求速率是使用椭圆曲线（ECC）非对称 KMS 密钥和 SM 非对称 KMS 密钥的加密操作的共享请求配额。

  例如，请求配额为每秒 1,000 个操作，您可以使用可以签名和验证的 ECC KMS 密钥发出 400 个签名请求和 200 个验证请求，再加上 250 个签名请求和 150 个使用可以签名和验证的 SM2 KMS 密钥进行验证请求。

- 自定义密钥存储请求限额是对自定义密钥存储中 KMS 密钥进行加密操作的共享请求限额。此限制针对每个自定义密钥存储单独计算。

  对自定义密钥存储中对称加密 KMS 密钥执行的密码操作将会同时消耗该账户的密码操作（对称）请求速率和该自定义密钥存储的 自定义密钥存储请求配额。

不同密钥类型的配额也是单独计算的。例如，在亚太地区（新加坡）区域中，如果同时使用对称 KMS 密钥和非对称 KMS 密钥，则使用对称 KMS 密钥（包括 HMAC 密钥）每秒最多可发出 10000 次调用，另外加上使用 RSA 非对称 KMS 密钥每秒最多可发出 500 次调用，另外再加上使用基于 ECC 的 KMS 密钥每秒最多可发出 300 个请求。

代表您发出的 API 请求

您可以直接发出 API 请求，也可以使用代表您发出 API 请求 Amazon KMS 的集成 Amazon 服务。该配额对两种类型的请求都适用。

例如，您可以使用借助 KMS 密钥的服务器端加密 (SSE-KMS)，将数据存储在 Amazon S3 中。每次您上传或下载使用 SSE-KMS 加密的 S3 对象时，Amazon S3 都会代表您发出`GenerateDataKey`（用于上传）或`Decrypt`（用于下载）请求。 Amazon KMS 这些请求计入您的配额，因此，如果您每秒上传或下载使用 SSE-KM Amazon KMS S 加密的 S3 对象的总数超过 5,500 个（或 10,000 或 50,000 个，视您而定 Amazon Web Services 区域），则会限制这些请求。

跨账户请求

当一个应用程序 Amazon Web Services 账户 使用另一个账户拥有的 KMS 密钥时，它被称为跨账户请求。对于跨账户请求， Amazon KMS 会限制发出请求的帐户，而不是拥有 KMS 密钥的账户。例如，如果账户 A 中的应用程序使用账户 B 中的 KMS 密钥，那么仅对该 KMS 密钥的使用应用账户 A 中的配额。

自定义密钥存储请求限额

Amazon KMS 维护对自定义密钥存储库中的 KMS 密钥进行加密操作的请求配额。这些请求限额针对每个自定义密钥存储单独计算。

自定义密钥存储请求限额	每个自定义密钥存储的默认值（每秒请求数）	可调整
Amazon CloudHSM 密钥库请求配额	1800	否
外部密钥存储 请求限额	1800	是

> **Note**
>
> Amazon KMS 自定义密钥库请求配额不会显示在 Service Quotas 控制台中。您无法通过服务限额 API 操作查看或管理这些限额。要请求更改您的外部密钥存储请求限额，请访问 Amazon Web Services 支持 中心 并创建工单。
>
> 如果与 Amazon CloudHSM 密钥库关联的 Amazon CloudHSM 集群正在处理大量命令，包括那些与自定义密钥存储无关的命令，那么你可能会得到一个 Amazon KMS `ThrottlingException`速 lower-than-expected率。如果发生这种情况，请将请求速率降低到 Amazon KMS，减少不相关的负载，或者为 Amazon CloudHSM 密钥存储使用专用 Amazon CloudHSM 集群。

Amazon KMS 报告指标中外部密钥存储请求的`ExternalKeyStoreThrottle` CloudWatch 限制。您可以使用此指标来查看节流模式、创建警报和调整外部密钥存储请求限额。

对自定义密钥存储中的 KMS 密钥进行 加密操作 的请求将计入这两个限额：

- 加密操作（对称）请求率限额（每账户）

  对自定义密钥存储中 KMS 密钥进行加密操作的请求将计入每个 Amazon Web Services 账户 和区域的 `Cryptographic operations (symmetric) request rate` 限额。例如，在美国东部（弗吉尼亚州北部）（us-east-1），每个 Amazon Web Services 账户 每秒最多可以有 10 万个对称加密 KMS 密钥请求，包括使用自定义密钥存储中的 KMS 密钥的请求。

- 自定义密钥存储请求限额（每自定义密钥存储）

  对自定义密钥存储中 KMS 密钥进行加密操作的请求也将计入每秒 1800 次操作的 `Custom key store request quota`。这些限额针对每个自定义密钥存储单独计算。它们可能包括来自多个 Amazon Web Services 账户 在自定义密钥存储中使用 KMS 密钥的请求。

例如，在美国东部（弗吉尼亚州北部）（us-east-1）区域，对一个自定义密钥存储中的一个 KMS 密钥执行的 Encrypt 操作将计入该账户和区域的 `Cryptographic operations (symmetric) request rate` 账户级别配额（每秒 10 万个请求），同时计入其自定义密钥存储的 `Custom key store request quota`（每秒 1800 个请求）。但是，对自定义密钥存储库中的 KMS 密钥进行管理操作的请求仅适用于其账户级别配额（每秒 15 个请求）。PutKeyPolicy

限制 Amazon KMS 请求

为了确保 Amazon KMS 可以对来自所有客户的 API 请求提供快速可靠的响应，它会限制超出特定边界的 API 请求。

当 Amazon KMS 拒绝本来可能有效的请求并返回类似以下内容的 `ThrottlingException` 错误时，会进行限制。

```
You have exceeded the rate at which you may call KMS. Reduce the frequency of your
 calls.
(Service: AWSKMS; Status Code: 400; Error Code: ThrottlingException; Request ID: <ID>
```

Amazon KMS 限制满足以下条件的请求。

- 每秒请求的速率超过一个账户和区域的 Amazon KMS 请求配额。

  例如，如果您账户中的用户在一秒钟提交了 1000 个 `DescribeKey` 请求，Amazon KMS 会限制这一秒钟的所有后续 `DescribeKey` 请求。

  要对限制进行响应，请使用退避和重试策略。在某些 Amazon 开发工具包中，会针对 HTTP 400 错误自动实施这一策略。
- 用于更改同一 KMS 密钥状态的请求突发或持续高速率。这种情况通常称为“热键”。

  例如，如果您账户中的某个应用程序针对同一个 KMS 密钥发送一串持久的 `EnableKey` 和 `DisableKey` 请求，Amazon KMS 将限制请求。即使请求没有超出 `EnableKey` 和 `DisableKey` 操作的每秒请求数请求限制，也会进行此限制。

  要响应限制，请调整您的应用程序逻辑，使其只发出必需的请求或合并多个函数的请求。
- 当与 Amazon CloudHSM 密钥存储相关联的 Amazon CloudHSM 集群正在处理大量命令（包括与 Amazon CloudHSM 密钥存储不相关的命令）时，对Amazon CloudHSM 密钥存储中的 KMS 密钥的操作请求可能会以低于预期的速率被节流。

  （当 Amazon CloudHSM 集群中没有可用的 PKCS #11 会话时，Amazon KMS 将不再针对对 Amazon CloudHSM 密钥存储中的 KMS 密钥进行操作的请求进行节流。相反，其将引发 `KMSInternalException`，以及请您重试请求的建议。）

要查看请求速率的趋势，请使用 Service Quotas 控制台。您也可以创建 Amazon CloudWatch 告警，当您的请求速率达到配额值的特定百分比时向您发出提示。有关详细信息，请参阅 Amazon 安全博客中使用 Service Quotas 和 Amazon CloudWatch 管理您的 Amazon KMS API 请求速度。

所有 Amazon KMS 限额均可调整，但按需轮换资源限额和Amazon CloudHSM 密钥存储请求限额除外。要请求提高配额，请参阅《服务配额用户指南》中的请求提高配额。要申请减少限额、更改服务限额中未列出的限额，或在 Amazon KMS 服务限额不可用的 Amazon Web Services 区域 中更改限额，请访问 Amazon Web Services 支持 中心并创建案例。

> **Note**
>
> Amazon KMS 自定义密钥存储请求限额 不会在服务限额控制台中显示。您无法通过服务限额 API 操作查看或管理这些限额。要请求更改您的外部密钥存储请求限额，请访问 Amazon Web Services 支持 中心 并创建工单。

使用的代码示 Amazon KMS 例 Amazon SDKs

以下代码示例说明如何 Amazon KMS 使用 Amazon 软件开发套件 (SDK)。

基本功能是向您展示如何在服务中执行基本操作的代码示例。

操作是大型程序的代码摘录，必须在上下文中运行。您可以通过操作了解如何调用单个服务函数，还可以通过函数相关场景的上下文查看操作。

场景是向您展示如何通过在一个服务中调用多个函数或与其他 Amazon Web Services 服务服务结合来完成特定任务的代码示例。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

开始使用

你好 Amazon Key Management Service

以下代码示例展示了如何开始使用 Amazon Key Management Service。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
import software.amazon.awssdk.services.kms.KmsAsyncClient;
import software.amazon.awssdk.services.kms.model.ListKeysRequest;
import software.amazon.awssdk.services.kms.paginators.ListKeysPublisher;
import java.util.concurrent.CompletableFuture;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
```

```
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class HelloKMS {
    public static void main(String[] args) {
        listAllKeys();
    }

    public static void listAllKeys() {
        KmsAsyncClient kmsAsyncClient = KmsAsyncClient.builder()
            .build();
        ListKeysRequest listKeysRequest = ListKeysRequest.builder()
            .limit(15)
            .build();

        /*
         * The `subscribe` method is required when using paginator methods in the
 AWS SDK
         * because paginator methods return an instance of a `ListKeysPublisher`,
 which is
         * based on a reactive stream. This allows asynchronous retrieval of
 paginated
         * results as they become available. By subscribing to the stream, we can
 process
         * each page of results as they are emitted.
         */
        ListKeysPublisher keysPublisher =
 kmsAsyncClient.listKeysPaginator(listKeysRequest);
        CompletableFuture<Void> future = keysPublisher
            .subscribe(r -> r.keys().forEach(key ->
                System.out.println("The key ARN is: " + key.keyArn() + ". The key
 Id is: " + key.keyId())))
            .whenComplete((result, exception) -> {
                if (exception != null) {
                    System.err.println("Error occurred: " +
 exception.getMessage());
                } else {
                    System.out.println("Successfully listed all keys.");
                }
            });

        try {
            future.join();
```

```
        } catch (Exception e) {
            System.err.println("Failed to list keys: " + e.getMessage());
        }
    }
}
```

- 有关 API 的详细信息，请参阅 Amazon SDK for Java 2.x API 参考ListKeys中的。

PHP

适用于 PHP 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
include "vendor/autoload.php";

use Aws\Kms\KmsClient;

echo "This file shows how to connect to the KmsClient, uses a paginator to get
 the keys for the account, and lists the KeyIds for up to 10 keys.\n";

$client = new KmsClient([]);

$pageLength = 10; // Change this value to change the number of records shown, or
 to break up the result into pages.

$keys = [];
$keysPaginator = $client->getPaginator("ListKeys", ['Limit' => $pageLength]);
foreach($keysPaginator as $page){
    foreach($page['Keys'] as $index => $key){
        echo "The $index index Key's ID is: {$key['KeyId']}\n";
    }
    echo "End of page one of results. Alter the \$pageLength variable to see more
 results.\n";
    break;
}
```

- 有关 API 的详细信息，请参阅 适用于 PHP 的 Amazon SDK API 参考ListKeys中的。

代码示例

- 使用的基本示 Amazon KMS 例 Amazon SDKs
  - 你好 Amazon Key Management Service
  - 学习 Amazon KMS 使用 Amazon SDK 的基础知识
  - 用于 Amazon KMS 使用的操作 Amazon SDKs
    - CreateAlias与 Amazon SDK 或 CLI 配合使用
    - CreateGrant与 Amazon SDK 或 CLI 配合使用
    - CreateKey与 Amazon SDK 或 CLI 配合使用
    - Decrypt与 Amazon SDK 或 CLI 配合使用
    - DeleteAlias与 Amazon SDK 或 CLI 配合使用
    - DescribeKey与 Amazon SDK 或 CLI 配合使用
    - DisableKey与 Amazon SDK 或 CLI 配合使用
    - EnableKey与 Amazon SDK 或 CLI 配合使用
    - EnableKeyRotation与 Amazon SDK 或 CLI 配合使用
    - Encrypt与 Amazon SDK 或 CLI 配合使用
    - GenerateDataKey与 Amazon SDK 或 CLI 配合使用
    - GenerateDataKeyWithoutPlaintext与 Amazon SDK 或 CLI 配合使用
    - GenerateRandom与 Amazon SDK 或 CLI 配合使用
    - GetKeyPolicy与 Amazon SDK 或 CLI 配合使用
    - ListAliases与 Amazon SDK 或 CLI 配合使用
    - ListGrants与 Amazon SDK 或 CLI 配合使用
    - ListKeyPolicies与 Amazon SDK 或 CLI 配合使用
    - ListKeys与 Amazon SDK 或 CLI 配合使用
    - PutKeyPolicy与 Amazon SDK 或 CLI 配合使用
    - ReEncrypt与 Amazon SDK 或 CLI 配合使用
    - RetireGrant与 Amazon SDK 或 CLI 配合使用
    - RevokeGrant与 Amazon SDK 或 CLI 配合使用

    - ScheduleKeyDeletion与 Amazon SDK 或 CLI 配合使用
    - Sign与 Amazon SDK 或 CLI 配合使用
    - TagResource与 Amazon SDK 或 CLI 配合使用
    - UpdateAlias与 Amazon SDK 或 CLI 配合使用
    - Verify与 Amazon SDK 或 CLI 配合使用
- Amazon KMS 使用场景 Amazon SDKs
    - 使用 v2 使用 DynamoDB 表加密 Amazon Command Line Interface

使用的基本示 Amazon KMS 例 Amazon SDKs

以下代码示例说明如何使用 with 的基础 Amazon Key Management Service 知识 Amazon SDKs。

示例

- 你好 Amazon Key Management Service
- 学习 Amazon KMS 使用 Amazon SDK 的基础知识
- 用于 Amazon KMS 使用的操作 Amazon SDKs
    - CreateAlias与 Amazon SDK 或 CLI 配合使用
    - CreateGrant与 Amazon SDK 或 CLI 配合使用
    - CreateKey与 Amazon SDK 或 CLI 配合使用
    - Decrypt与 Amazon SDK 或 CLI 配合使用
    - DeleteAlias与 Amazon SDK 或 CLI 配合使用
    - DescribeKey与 Amazon SDK 或 CLI 配合使用
    - DisableKey与 Amazon SDK 或 CLI 配合使用
    - EnableKey与 Amazon SDK 或 CLI 配合使用
    - EnableKeyRotation与 Amazon SDK 或 CLI 配合使用
    - Encrypt与 Amazon SDK 或 CLI 配合使用
    - GenerateDataKey与 Amazon SDK 或 CLI 配合使用
    - GenerateDataKeyWithoutPlaintext与 Amazon SDK 或 CLI 配合使用
    - GenerateRandom与 Amazon SDK 或 CLI 配合使用
    - GetKeyPolicy与 Amazon SDK 或 CLI 配合使用
    - ListAliases与 Amazon SDK 或 CLI 配合使用

- ListGrants与 Amazon SDK 或 CLI 配合使用
- ListKeyPolicies与 Amazon SDK 或 CLI 配合使用
- ListKeys与 Amazon SDK 或 CLI 配合使用
- PutKeyPolicy与 Amazon SDK 或 CLI 配合使用
- ReEncrypt与 Amazon SDK 或 CLI 配合使用
- RetireGrant与 Amazon SDK 或 CLI 配合使用
- RevokeGrant与 Amazon SDK 或 CLI 配合使用
- ScheduleKeyDeletion与 Amazon SDK 或 CLI 配合使用
- Sign与 Amazon SDK 或 CLI 配合使用
- TagResource与 Amazon SDK 或 CLI 配合使用
- UpdateAlias与 Amazon SDK 或 CLI 配合使用
- Verify与 Amazon SDK 或 CLI 配合使用

你好 Amazon Key Management Service

以下代码示例展示了如何开始使用 Amazon Key Management Service。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
import software.amazon.awssdk.services.kms.KmsAsyncClient;
import software.amazon.awssdk.services.kms.model.ListKeysRequest;
import software.amazon.awssdk.services.kms.paginators.ListKeysPublisher;
import java.util.concurrent.CompletableFuture;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
```

```
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class HelloKMS {
    public static void main(String[] args) {
        listAllKeys();
    }

    public static void listAllKeys() {
        KmsAsyncClient kmsAsyncClient = KmsAsyncClient.builder()
            .build();
        ListKeysRequest listKeysRequest = ListKeysRequest.builder()
            .limit(15)
            .build();

        /*
         * The `subscribe` method is required when using paginator methods in the
 AWS SDK
         * because paginator methods return an instance of a `ListKeysPublisher`,
 which is
         * based on a reactive stream. This allows asynchronous retrieval of
 paginated
         * results as they become available. By subscribing to the stream, we can
 process
         * each page of results as they are emitted.
         */
        ListKeysPublisher keysPublisher =
 kmsAsyncClient.listKeysPaginator(listKeysRequest);
        CompletableFuture<Void> future = keysPublisher
            .subscribe(r -> r.keys().forEach(key ->
                System.out.println("The key ARN is: " + key.keyArn() + ". The key
 Id is: " + key.keyId())))
            .whenComplete((result, exception) -> {
                if (exception != null) {
                    System.err.println("Error occurred: " +
 exception.getMessage());
                } else {
                    System.out.println("Successfully listed all keys.");
                }
            });

        try {
```

```
            future.join();
        } catch (Exception e) {
            System.err.println("Failed to list keys: " + e.getMessage());
        }
    }
}
```

- 有关 API 的详细信息，请参阅 Amazon SDK for Java 2.x API 参考ListKeys中的。

PHP

适用于 PHP 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
include "vendor/autoload.php";

use Aws\Kms\KmsClient;

echo "This file shows how to connect to the KmsClient, uses a paginator to get
 the keys for the account, and lists the KeyIds for up to 10 keys.\n";

$client = new KmsClient([]);

$pageLength = 10; // Change this value to change the number of records shown, or
 to break up the result into pages.

$keys = [];
$keysPaginator = $client->getPaginator("ListKeys", ['Limit' => $pageLength]);
foreach($keysPaginator as $page){
    foreach($page['Keys'] as $index => $key){
        echo "The $index index Key's ID is: {$key['KeyId']}\n";
    }
    echo "End of page one of results. Alter the \$pageLength variable to see more
 results.\n";
    break;
}
```

- 有关 API 的详细信息，请参阅 适用于 PHP 的 Amazon SDK API 参考ListKeys中的。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

学习 Amazon KMS 使用 Amazon SDK 的基础知识

以下代码示例演示了如何：

- 创建 KMS 密钥。
- 列出您账户的 KMS 密钥并获取有关它们的详细信息。
- 启用和禁用 KMS 密钥。
- 生成可用于客户端加密的对称数据密钥。
- 生成用于对数据进行数字签名的非对称密钥。
- 标签键。
- 删除 KMS 密钥。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

在命令提示符中运行场景。

```
import software.amazon.awssdk.core.SdkBytes;
import software.amazon.awssdk.regions.Region;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.services.kms.model.AlreadyExistsException;
import software.amazon.awssdk.services.kms.model.DisabledException;
```

```
import software.amazon.awssdk.services.kms.model.EnableKeyRotationResponse;
import software.amazon.awssdk.services.kms.model.KmsException;
import software.amazon.awssdk.services.kms.model.NotFoundException;
import software.amazon.awssdk.services.kms.model.RevokeGrantResponse;
import java.util.List;
import java.util.Scanner;
import java.util.concurrent.CompletableFuture;
import java.util.concurrent.CompletionException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */

public class KMSScenario {
    public static final String DASHES = new String(new char[80]).replace("\0",
 "-");
    private static String accountId = "";

    private static final Logger logger =
 LoggerFactory.getLogger(KMSScenario.class);

    static KMSActions kmsActions = new KMSActions();

    static Scanner scanner = new Scanner(System.in);

    static String aliasName = "alias/dev-encryption-key";

    public static void main(String[] args) {
        final String usage = """
            Usage: <granteePrincipal>

            Where:
               granteePrincipal - The principal (user, service account, or group)
 to whom the grant or permission is being given.
            """;

        if (args.length != 1) {
            logger.info(usage);
```

```
            return;
        }
        String granteePrincipal = args[0];
        String policyName = "default";

        accountId = kmsActions.getAccountId();
        String keyDesc = "Created by the AWS KMS API";

        logger.info(DASHES);
        logger.info("""
            Welcome to the AWS Key Management SDK Basics scenario.

            This program demonstrates how to interact with AWS Key Management
using the AWS SDK for Java (v2).
            The AWS Key Management Service (KMS) is a secure and highly available
service that allows you to create
            and manage AWS KMS keys and control their use across a wide range of
AWS services and applications.
            KMS provides a centralized and unified approach to managing
encryption keys, making it easier to meet your
            data protection and regulatory compliance requirements.

            This Basics scenario creates two key types:

            - A symmetric encryption key is used to encrypt and decrypt data.
            - An asymmetric key used to digitally sign data.

            Let's get started...
            """);
        waitForInputToContinue(scanner);

        try {
        // Run the methods that belong to this scenario.
        String targetKeyId = runScenario(granteePrincipal, keyDesc, policyName);
        requestDeleteResources(aliasName, targetKeyId);

        } catch (Throwable rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
```

```
        }
    }

    private static String runScenario(String granteePrincipal, String keyDesc,
String policyName) throws Throwable {
        logger.info(DASHES);
        logger.info("1. Create a symmetric KMS key\n");
        logger.info("First, the program will creates a symmetric KMS key that you
can used to encrypt and decrypt data.");
        waitForInputToContinue(scanner);
        String targetKeyId;
        try {
            CompletableFuture<String> futureKeyId =
kmsActions.createKeyAsync(keyDesc);
            targetKeyId = futureKeyId.join();
            logger.info("A symmetric key was successfully created " +
targetKeyId);

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            throw cause;
        }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("""
            2. Enable a KMS key

            By default, when the SDK creates an AWS key, it is enabled. The next
bit of code checks to
            determine if the key is enabled.
             """);
        waitForInputToContinue(scanner);
        boolean isEnabled;
        try {
            CompletableFuture<Boolean> futureIsKeyEnabled =
kmsActions.isKeyEnabledAsync(targetKeyId);
            isEnabled = futureIsKeyEnabled.join();
```

```
            logger.info("Is the key enabled? {}", isEnabled);

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            throw cause;
        }

        if (!isEnabled)
            try {
                CompletableFuture<Void> future =
kmsActions.enableKeyAsync(targetKeyId);
                future.join();

            } catch (RuntimeException rt) {
                Throwable cause = rt.getCause();
                if (cause instanceof KmsException kmsEx) {
                    logger.info("KMS error occurred: Error message: {}, Error
code {}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
                } else {
                    logger.info("An unexpected error occurred: " +
rt.getMessage());
                }
                throw cause;
            }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("3. Encrypt data using the symmetric KMS key");
        String plaintext = "Hello, AWS KMS!";
        logger.info("""
            One of the main uses of symmetric keys is to encrypt and decrypt
data.
            Next, the code encrypts the string {} with the SYMMETRIC_DEFAULT
encryption algorithm.
            """, plaintext);
        waitForInputToContinue(scanner);
        SdkBytes encryptedData;
        try {
```

```
            CompletableFuture<SdkBytes> future =
kmsActions.encryptDataAsync(targetKeyId, plaintext);
            encryptedData = future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof DisabledException kmsDisabledEx) {
                logger.info("KMS error occurred due to a disabled
key: Error message: {}, Error code {}", kmsDisabledEx.getMessage(),
kmsDisabledEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            deleteKey(targetKeyId);
            throw cause;
        }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("4. Create an alias");
        logger.info("""

            The alias name should be prefixed with 'alias/'.
            The default, 'alias/dev-encryption-key'.
             """);
        waitForInputToContinue(scanner);

        try {
            CompletableFuture<Void> future =
kmsActions.createCustomAliasAsync(targetKeyId, aliasName);
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof AlreadyExistsException kmsExistsEx) {
                if (kmsExistsEx.getMessage().contains("already exists")) {
                    logger.info("The alias '" + aliasName + "' already exists.
Moving on...");
                }
            } else {
                logger.error("An unexpected error occurred: " + rt.getMessage(),
rt);
                deleteKey(targetKeyId);
                throw cause;
```

```
            }
        }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("5. List all of your aliases");
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<Object> future = kmsActions.listAllAliasesAsync();
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            deleteAliasName(aliasName);
            deleteKey(targetKeyId);
            throw cause;
        }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("6. Enable automatic rotation of the KMS key");
        logger.info("""

            By default, when the SDK enables automatic rotation of a KMS key,
            KMS rotates the key material of the KMS key one year (approximately
365 days) from the enable date and every year
            thereafter.
            """);
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<EnableKeyRotationResponse> future =
kmsActions.enableKeyRotationAsync(targetKeyId);
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
```

```
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            deleteAliasName(aliasName);
            deleteKey(targetKeyId);
            throw cause;
        }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("""
            7. Create a grant

            A grant is a policy instrument that allows Amazon Web Services
principals to use KMS keys.
            It also can allow them to view a KMS key (DescribeKey) and create and
manage grants.
            When authorizing access to a KMS key, grants are considered along
with key policies and IAM policies.
            """);

        waitForInputToContinue(scanner);
        String grantId = null;
        try {
            CompletableFuture<String> futureGrantId =
kmsActions.grantKeyAsync(targetKeyId, granteePrincipal);
            grantId = futureGrantId.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            deleteKey(targetKeyId);
            throw cause;
        }
        waitForInputToContinue(scanner);
        logger.info(DASHES);
```

```
        logger.info(DASHES);
        logger.info("8. List grants for the KMS key");
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<Object> future =
kmsActions.displayGrantIdsAsync(targetKeyId);
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            deleteAliasName(aliasName);
            deleteKey(targetKeyId);
            throw cause;
        }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("9. Revoke the grant");
        logger.info("""
            The revocation of a grant immediately removes the permissions and
access that the grant had provided.
            This means that any principal (user, role, or service) that was
granted access to perform specific
            KMS operations on a KMS key will no longer be able to perform those
operations.
            """);
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<RevokeGrantResponse> future =
kmsActions.revokeKeyGrantAsync(targetKeyId, grantId);
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                if (kmsEx.getMessage().contains("Grant does not exist")) {
                    logger.info("The grant ID '" + grantId + "' does not exist.
Moving on...");
```

```
                } else {
                    logger.info("KMS error occurred: Error message: {}, Error
code {}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
                    throw cause;
                }
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
                deleteAliasName(aliasName);
                deleteKey(targetKeyId);
                throw cause;
            }
        }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("10. Decrypt the data\n");
        logger.info("""
            Lets decrypt the data that was encrypted in an early step.
            The code uses the same key to decrypt the string that we encrypted
earlier in the program.
            """);
        waitForInputToContinue(scanner);
        String decryptedData = "";
        try {
            CompletableFuture<String> future =
kmsActions.decryptDataAsync(encryptedData, targetKeyId);
            decryptedData = future.join();
            logger.info("Decrypted data: " + decryptedData);

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            deleteAliasName(aliasName);
            deleteKey(targetKeyId);
            throw cause;
        }
        logger.info("Decrypted text is: " + decryptedData);
        waitForInputToContinue(scanner);
```

```
        logger.info(DASHES);
        logger.info("11. Replace a key policy\n");
        logger.info("""
            A key policy is a resource policy for a KMS key. Key policies are the
primary way to control
            access to KMS keys. Every KMS key must have exactly one key policy.
The statements in the key policy
            determine who has permission to use the KMS key and how they can use
it.
            You can also use IAM policies and grants to control access to the KMS
key, but every KMS key
            must have a key policy.

            By default, when you create a key by using the SDK, a policy is
created that
            gives the AWS account that owns the KMS key full access to the KMS
key.

            Let's try to replace the automatically created policy with the
following policy.

                "Version":"2012-10-17",
                "Statement": [{
                "Effect": "Allow",
                "Principal": {"AWS": "arn:aws:iam::0000000000:root"},
                "Action": "kms:*",
                "Resource": "*"
                }]
            """);

        waitForInputToContinue(scanner);
        try {
            CompletableFuture<Boolean> future =
kmsActions.replacePolicyAsync(targetKeyId, policyName, accountId);
            boolean success = future.join();
            if (success) {
                logger.info("Key policy replacement succeeded.");
            } else {
                logger.error("Key policy replacement failed.");
            }

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
```

```
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            deleteAliasName(aliasName);
            deleteKey(targetKeyId);
            throw cause;
        }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("12. Get the key policy\n");
        logger.info("The next bit of code that runs gets the key policy to make
sure it exists.");
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<String> future =
kmsActions.getKeyPolicyAsync(targetKeyId, policyName);
            String policy = future.join();
            if (!policy.isEmpty()) {
                logger.info("Retrieved policy: " + policy);
            }

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            deleteAliasName(aliasName);
            deleteKey(targetKeyId);
            throw cause;
        }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("13. Create an asymmetric KMS key and sign your data\n");
        logger.info("""
             Signing your data with an AWS key can provide several benefits that
make it an attractive option
```

```
            for your data signing needs. By using an AWS KMS key, you can
leverage the
            security controls and compliance features provided by AWS,
            which can help you meet various regulatory requirements and enhance
the overall security posture
            of your organization.
            """);
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<Boolean> future = kmsActions.signVerifyDataAsync();
            boolean success = future.join();
            if (success) {
                logger.info("Sign and verify data operation succeeded.");
            } else {
                logger.error("Sign and verify data operation failed.");
            }

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            deleteAliasName(aliasName);
            deleteKey(targetKeyId);
            throw cause;
        }
        waitForInputToContinue(scanner);

        logger.info(DASHES);
        logger.info("14. Tag your symmetric KMS Key\n");
        logger.info("""
            By using tags, you can improve the overall management, security, and
governance of your
            KMS keys, making it easier to organize, track, and control access to
your encrypted data within
            your AWS environment
            """);
        waitForInputToContinue(scanner);
        try {
            CompletableFuture<Void> future =
kmsActions.tagKMSKeyAsync(targetKeyId);
```

```
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
 {}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
            deleteAliasName(aliasName);
            deleteKey(targetKeyId);
            throw cause;
        }
        waitForInputToContinue(scanner);
        return targetKeyId;
    }

    // Deletes KMS resources with user input.
    private static void requestDeleteResources(String aliasName, String
 targetKeyId) {
        logger.info(DASHES);
        logger.info("15. Schedule the deletion of the KMS key\n");
        logger.info("""
            By default, KMS applies a waiting period of 30 days,
            but you can specify a waiting period of 7-30 days. When this
 operation is successful,
            the key state of the KMS key changes to PendingDeletion and the key
 can't be used in any
            cryptographic operations. It remains in this state for the duration
 of the waiting period.

            Deleting a KMS key is a destructive and potentially dangerous
 operation. When a KMS key is deleted,
            all data that was encrypted under the KMS key is unrecoverable.
            """);
        logger.info("Would you like to delete the Key Management resources? (y/
n)");
        String delAns = scanner.nextLine().trim();
        if (delAns.equalsIgnoreCase("y")) {
            logger.info("You selected to delete the AWS KMS resources.");
            waitForInputToContinue(scanner);
            try {
```

```
                CompletableFuture<Void> future =
kmsActions.deleteSpecificAliasAsync(aliasName);
                future.join();

            } catch (RuntimeException rt) {
                Throwable cause = rt.getCause();
                if (cause instanceof KmsException kmsEx) {
                    logger.info("KMS error occurred: Error message: {}, Error
code {}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
                } else {
                    logger.info("An unexpected error occurred: " +
rt.getMessage());
                }
            }
            waitForInputToContinue(scanner);
            try {
                CompletableFuture<Void> future =
kmsActions.disableKeyAsync(targetKeyId);
                future.join();

            } catch (RuntimeException rt) {
                Throwable cause = rt.getCause();
                if (cause instanceof KmsException kmsEx) {
                    logger.info("KMS error occurred: Error message: {}, Error
code {}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
                } else {
                    logger.info("An unexpected error occurred: " +
rt.getMessage());
                }
            }

            try {
                CompletableFuture<Void> future =
kmsActions.deleteKeyAsync(targetKeyId);
                future.join();

            } catch (RuntimeException rt) {
                Throwable cause = rt.getCause();
                if (cause instanceof KmsException kmsEx) {
                    logger.info("KMS error occurred: Error message: {}, Error
code {}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
                } else {
                    logger.info("An unexpected error occurred: " +
rt.getMessage());
```

```
                }
            }

        } else {
            logger.info("The Key Management resources will not be deleted");
        }

        logger.info(DASHES);
        logger.info("This concludes the AWS Key Management SDK scenario");
        logger.info(DASHES);
    }

    // This method is invoked from Exceptions to clean up the resources.
    private static void deleteKey(String targetKeyId) {
        try {
            CompletableFuture<Void> future =
kmsActions.disableKeyAsync(targetKeyId);
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
        }

        try {
            CompletableFuture<Void> future =
kmsActions.deleteKeyAsync(targetKeyId);
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
{}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
        }
    }
```

```
    // This method is invoked from Exceptions to clean up the resources.
    private static void deleteAliasName(String aliasName) {
        try {
            CompletableFuture<Void> future =
 kmsActions.deleteSpecificAliasAsync(aliasName);
            future.join();

        } catch (RuntimeException rt) {
            Throwable cause = rt.getCause();
            if (cause instanceof KmsException kmsEx) {
                logger.info("KMS error occurred: Error message: {}, Error code
 {}", kmsEx.getMessage(), kmsEx.awsErrorDetails().errorCode());
            } else {
                logger.info("An unexpected error occurred: " + rt.getMessage());
            }
        }
    }

    private static void waitForInputToContinue(Scanner scanner) {
        while (true) {
            logger.info("");
            logger.info("Enter 'c' followed by <ENTER> to continue:");
            String input = scanner.nextLine();

            if (input.trim().equalsIgnoreCase("c")) {
                logger.info("Continuing with the program...");
                logger.info("");
                break;
            } else {
                // Handle invalid input.
                logger.info("Invalid input. Please try again.");
            }
        }
    }
}
```

定义一个包装 KMS 操作的类。

```
public class KMSActions {
    private static final Logger logger =
 LoggerFactory.getLogger(KMSActions.class);
```

```
    private static KmsAsyncClient kmsAsyncClient;

    /**
     * Retrieves an asynchronous AWS Key Management Service (KMS) client.
     * <p>
     * This method creates and returns a singleton instance of the KMS async
client, with the following configurations:
     * <ul>
     *   <li>Max concurrency: 100</li>
     *   <li>Connection timeout: 60 seconds</li>
     *   <li>Read timeout: 60 seconds</li>
     *   <li>Write timeout: 60 seconds</li>
     *   <li>API call timeout: 2 minutes</li>
     *   <li>API call attempt timeout: 90 seconds</li>
     *   <li>Retry policy: up to 3 retries</li>
     *   <li>Credentials provider: environment variable credentials provider</li>
     * </ul>
     * <p>
     * If the client instance has already been created, it is returned instead of
creating a new one.
     *
     * @return the KMS async client instance
     */
    private static KmsAsyncClient getAsyncClient() {
        if (kmsAsyncClient == null) {
            SdkAsyncHttpClient httpClient = NettyNioAsyncHttpClient.builder()
                .maxConcurrency(100)
                .connectionTimeout(Duration.ofSeconds(60))
                .readTimeout(Duration.ofSeconds(60))
                .writeTimeout(Duration.ofSeconds(60))
                .build();

            ClientOverrideConfiguration overrideConfig =
ClientOverrideConfiguration.builder()
                .apiCallTimeout(Duration.ofMinutes(2))
                .apiCallAttemptTimeout(Duration.ofSeconds(90))
                .retryPolicy(RetryPolicy.builder()
                    .numRetries(3)
                    .build())
                .build();

            kmsAsyncClient = KmsAsyncClient.builder()
                .httpClient(httpClient)
                .overrideConfiguration(overrideConfig)
```

```
                .build();
        }
        return kmsAsyncClient;
    }

    /**
     * Creates a new symmetric encryption key asynchronously.
     *
     * @param keyDesc the description of the key to be created
     * @return a {@link CompletableFuture} that completes with the ID of the
newly created key
     * @throws RuntimeException if an error occurs while creating the key
     */
    public CompletableFuture<String> createKeyAsync(String keyDesc) {
        CreateKeyRequest keyRequest = CreateKeyRequest.builder()
            .description(keyDesc)
            .keySpec(KeySpec.SYMMETRIC_DEFAULT)
            .keyUsage(KeyUsageType.ENCRYPT_DECRYPT)
            .build();

        return getAsyncClient().createKey(keyRequest)
            .thenApply(resp -> resp.keyMetadata().keyId())
            .exceptionally(ex -> {
                throw new RuntimeException("An error occurred while creating the
key: " + ex.getMessage(), ex);
            });
    }

    /**
     * Asynchronously checks if a specified key is enabled.
     *
     * @param keyId the ID of the key to check
     * @return a {@link CompletableFuture} that, when completed, indicates
whether the key is enabled or not
     *
     * @throws RuntimeException if an exception occurs while checking the key
state
     */
    public CompletableFuture<Boolean> isKeyEnabledAsync(String keyId) {
        DescribeKeyRequest keyRequest = DescribeKeyRequest.builder()
            .keyId(keyId)
            .build();
```

```
        CompletableFuture<DescribeKeyResponse> responseFuture =
getAsyncClient().describeKey(keyRequest);
        return responseFuture.whenComplete((resp, ex) -> {
            if (resp != null) {
                KeyState keyState = resp.keyMetadata().keyState();
                if (keyState == KeyState.ENABLED) {
                    logger.info("The key is enabled.");
                } else {
                    logger.info("The key is not enabled. Key state: {}",
keyState);
                }
            } else {
                throw new RuntimeException(ex);
            }
        }).thenApply(resp -> resp.keyMetadata().keyState() == KeyState.ENABLED);
    }

    /**
     * Asynchronously enables the specified key.
     *
     * @param keyId the ID of the key to enable
     * @return a {@link CompletableFuture} that completes when the key has been
enabled
     */
    public CompletableFuture<Void> enableKeyAsync(String keyId) {
        EnableKeyRequest enableKeyRequest = EnableKeyRequest.builder()
            .keyId(keyId)
            .build();

        CompletableFuture<EnableKeyResponse> responseFuture =
getAsyncClient().enableKey(enableKeyRequest);
        responseFuture.whenComplete((response, exception) -> {
            if (exception == null) {
                logger.info("Key with ID [{}] has been enabled.", keyId);
            } else {
                if (exception instanceof KmsException kmsEx) {
                    throw new RuntimeException("KMS error occurred while enabling
key: " + kmsEx.getMessage(), kmsEx);
                } else {
                    throw new RuntimeException("An unexpected error occurred
while enabling key: " + exception.getMessage(), exception);
                }
            }
        });
```

```
        return responseFuture.thenApply(response -> null);
    }

    /**
     * Encrypts the given text asynchronously using the specified KMS client and
key ID.
     *
     * @param keyId the ID of the KMS key to use for encryption
     * @param text the text to encrypt
     * @return a CompletableFuture that completes with the encrypted data as an
SdkBytes object
     */
    public CompletableFuture<SdkBytes> encryptDataAsync(String keyId, String
text) {
        SdkBytes myBytes = SdkBytes.fromUtf8String(text);
        EncryptRequest encryptRequest = EncryptRequest.builder()
            .keyId(keyId)
            .plaintext(myBytes)
            .build();

        CompletableFuture<EncryptResponse> responseFuture =
getAsyncClient().encrypt(encryptRequest).toCompletableFuture();
        return responseFuture.whenComplete((response, ex) -> {
            if (response != null) {
                String algorithm = response.encryptionAlgorithm().toString();
                logger.info("The string was encrypted with algorithm {}.",
algorithm);
            } else {
                throw new RuntimeException(ex);
            }
        }).thenApply(EncryptResponse::ciphertextBlob);
    }

    /**
     * Creates a custom alias for the specified target key asynchronously.
     *
     * @param targetKeyId the ID of the target key for the alias
     * @param aliasName   the name of the alias to create
     * @return a {@link CompletableFuture} that completes when the alias creation
operation is finished
     */
    public CompletableFuture<Void> createCustomAliasAsync(String targetKeyId,
String aliasName) {
```

```
        CreateAliasRequest aliasRequest = CreateAliasRequest.builder()
            .aliasName(aliasName)
            .targetKeyId(targetKeyId)
            .build();

        CompletableFuture<CreateAliasResponse> responseFuture =
getAsyncClient().createAlias(aliasRequest);
        responseFuture.whenComplete((response, exception) -> {
            if (exception == null) {
                logger.info("{} was successfully created.", aliasName);
            } else {
                if (exception instanceof ResourceExistsException) {
                    logger.info("Alias [{}] already exists. Moving on...",
aliasName);
                } else if (exception instanceof KmsException kmsEx) {
                    throw new RuntimeException("KMS error occurred while creating
alias: " + kmsEx.getMessage(), kmsEx);
                } else {
                    throw new RuntimeException("An unexpected error occurred
while creating alias: " + exception.getMessage(), exception);
                }
            }
        });

        return responseFuture.thenApply(response -> null);
    }

    /**
     * Asynchronously lists all the aliases in the current AWS account.
     *
     * @return a {@link CompletableFuture} that completes when the list of
aliases has been processed
     */
    public CompletableFuture<Object> listAllAliasesAsync() {
        ListAliasesRequest aliasesRequest = ListAliasesRequest.builder()
            .limit(15)
            .build();

        ListAliasesPublisher paginator =
getAsyncClient().listAliasesPaginator(aliasesRequest);
        return paginator.subscribe(response -> {
                response.aliases().forEach(alias ->
                    logger.info("The alias name is: " + alias.aliasName())
                );
```

```
            })
            .thenApply(v -> null)
            .exceptionally(ex -> {
                if (ex.getCause() instanceof KmsException) {
                    KmsException e = (KmsException) ex.getCause();
                    throw new RuntimeException("A KMS exception occurred: " +
e.getMessage());
                } else {
                    throw new RuntimeException("An unexpected error occurred: " +
ex.getMessage());
                }
            });
    }

    /**
     * Enables key rotation asynchronously for the specified key ID.
     *
     * @param keyId the ID of the key for which to enable key rotation
     * @return a CompletableFuture that represents the asynchronous operation of
enabling key rotation
     * @throws RuntimeException if there was an error enabling key rotation,
either due to a KMS exception or an unexpected error
     */
    public CompletableFuture<EnableKeyRotationResponse>
enableKeyRotationAsync(String keyId) {
        EnableKeyRotationRequest enableKeyRotationRequest =
EnableKeyRotationRequest.builder()
            .keyId(keyId)
            .build();

        CompletableFuture<EnableKeyRotationResponse> responseFuture =
getAsyncClient().enableKeyRotation(enableKeyRotationRequest);
        responseFuture.whenComplete((response, exception) -> {
            if (exception == null) {
                logger.info("Key rotation has been enabled for key with id [{}]",
keyId);
            } else {
                if (exception instanceof KmsException kmsEx) {
                    throw new RuntimeException("Failed to enable key rotation: "
+ kmsEx.getMessage(), kmsEx);
                } else {
                    throw new RuntimeException("An unexpected error occurred: " +
exception.getMessage(), exception);
                }
```

```
            }
        });

        return responseFuture;
    }

    /**
     * Grants permissions to a specified principal on a customer master key (CMK)
asynchronously.
     *
     * @param keyId             The unique identifier for the customer master key
(CMK) that the grant applies to.
     * @param granteePrincipal  The principal that is given permission to perform
the operations that the grant permits on the CMK.
     * @return A {@link CompletableFuture} that, when completed, contains the ID
of the created grant.
     * @throws RuntimeException If an error occurs during the grant creation
process.
     */
    public CompletableFuture<String> grantKeyAsync(String keyId, String
granteePrincipal) {
        List<GrantOperation> grantPermissions = List.of(
            GrantOperation.ENCRYPT,
            GrantOperation.DECRYPT,
            GrantOperation.DESCRIBE_KEY
        );

        CreateGrantRequest grantRequest = CreateGrantRequest.builder()
            .keyId(keyId)
            .name("grant1")
            .granteePrincipal(granteePrincipal)
            .operations(grantPermissions)
            .build();

        CompletableFuture<CreateGrantResponse> responseFuture =
getAsyncClient().createGrant(grantRequest);
        responseFuture.whenComplete((response, ex) -> {
            if (ex == null) {
                logger.info("Grant created successfully with ID: " +
response.grantId());
            } else {
                if (ex instanceof KmsException kmsEx) {
                    throw new RuntimeException("Failed to create grant: " +
kmsEx.getMessage(), kmsEx);
```

```
                } else {
                    throw new RuntimeException("An unexpected error occurred: " +
ex.getMessage(), ex);
                }
            }
        });

        return responseFuture.thenApply(CreateGrantResponse::grantId);
    }

    /**
     * Asynchronously displays the grant IDs for the specified key ID.
     *
     * @param keyId the ID of the AWS KMS key for which to list the grants
     * @return a {@link CompletableFuture} that, when completed, will be null
if the operation succeeded, or will throw a {@link RuntimeException} if the
operation failed
     * @throws RuntimeException if there was an error listing the grants, either
due to an {@link KmsException} or an unexpected error
     */
    public CompletableFuture<Object> displayGrantIdsAsync(String keyId) {
        ListGrantsRequest grantsRequest = ListGrantsRequest.builder()
            .keyId(keyId)
            .limit(15)
            .build();

        ListGrantsPublisher paginator =
getAsyncClient().listGrantsPaginator(grantsRequest);
        return paginator.subscribe(response -> {
                response.grants().forEach(grant -> {
                    logger.info("The grant Id is: " + grant.grantId());
                });
            })
            .thenApply(v -> null)
            .exceptionally(ex -> {
                Throwable cause = ex.getCause();
                if (cause instanceof KmsException) {
                    throw new RuntimeException("Failed to list grants: " +
cause.getMessage(), cause);
                } else {
                    throw new RuntimeException("An unexpected error occurred: " +
cause.getMessage(), cause);
                }
            });
```

```
    }

    /**
     * Revokes a grant for the specified AWS KMS key asynchronously.
     *
     * @param keyId   The ID or key ARN of the AWS KMS key.
     * @param grantId The identifier of the grant to be revoked.
     * @return A {@link CompletableFuture} representing the asynchronous
operation of revoking the grant.
     *         The {@link CompletableFuture} will complete with a {@link
RevokeGrantResponse} object
     *         if the operation is successful, or with a {@code null} value if an
error occurs.
     */
    public CompletableFuture<RevokeGrantResponse> revokeKeyGrantAsync(String
keyId, String grantId) {
        RevokeGrantRequest grantRequest = RevokeGrantRequest.builder()
            .keyId(keyId)
            .grantId(grantId)
            .build();

        CompletableFuture<RevokeGrantResponse> responseFuture =
getAsyncClient().revokeGrant(grantRequest);
        responseFuture.whenComplete((response, exception) -> {
            if (exception == null) {
                logger.info("Grant ID: [" + grantId + "] was successfully
revoked!");
            } else {
                if (exception instanceof KmsException kmsEx) {
                    if (kmsEx.getMessage().contains("Grant does not exist")) {
                        logger.info("The grant ID '" + grantId + "' does not
exist. Moving on...");
                    } else {
                        throw new RuntimeException("KMS error occurred: " +
kmsEx.getMessage(), kmsEx);
                    }
                } else {
                    throw new RuntimeException("An unexpected error occurred: " +
exception.getMessage(), exception);
                }
            }
        });

        return responseFuture;
```

```
    }


    /**
     * Asynchronously decrypts the given encrypted data using the specified key
ID.
     *
     * @param encryptedData The encrypted data to be decrypted.
     * @param keyId The ID of the key to be used for decryption.
     * @return A CompletableFuture that, when completed, will contain the
decrypted data as a String.
     *         If an error occurs during the decryption process, the
CompletableFuture will complete
     *         exceptionally with the error, and the method will return an empty
String.
     */
    public CompletableFuture<String> decryptDataAsync(SdkBytes encryptedData,
String keyId) {
        DecryptRequest decryptRequest = DecryptRequest.builder()
            .ciphertextBlob(encryptedData)
            .keyId(keyId)
            .build();

        CompletableFuture<DecryptResponse> responseFuture =
getAsyncClient().decrypt(decryptRequest);
        responseFuture.whenComplete((decryptResponse, exception) -> {
            if (exception == null) {
                logger.info("Data decrypted successfully for key ID: " + keyId);
            } else {
                if (exception instanceof KmsException kmsEx) {
                    throw new RuntimeException("KMS error occurred while
decrypting data: " + kmsEx.getMessage(), kmsEx);
                } else {
                    throw new RuntimeException("An unexpected error occurred
while decrypting data: " + exception.getMessage(), exception);
                }
            }
        });

        return responseFuture.thenApply(decryptResponse ->
decryptResponse.plaintext().asString(StandardCharsets.UTF_8));
    }

    /**
```

```
    * Asynchronously replaces the policy for the specified KMS key.
    *
    * @param keyId       the ID of the KMS key to replace the policy for
    * @param policyName  the name of the policy to be replaced
    * @param accountId   the AWS account ID to be used in the policy
    * @return a {@link CompletableFuture} that completes with a boolean
indicating
    *         whether the policy replacement was successful or not
    */
   public CompletableFuture<Boolean> replacePolicyAsync(String keyId, String
policyName, String accountId) {
       String policy = """
   {
     "Version":"2012-10-17",
     "Statement": [{
       "Effect": "Allow",
       "Principal": {"AWS": "arn:aws:iam::%s:root"},
       "Action": "kms:*",
       "Resource": "*"
     }]
   }
   """.formatted(accountId);

       PutKeyPolicyRequest keyPolicyRequest = PutKeyPolicyRequest.builder()
           .keyId(keyId)
           .policyName(policyName)
           .policy(policy)
           .build();

       // First, get the current policy to check if it exists
       return getAsyncClient().getKeyPolicy(r ->
r.keyId(keyId).policyName(policyName))
           .thenCompose(response -> {
               logger.info("Current policy exists. Replacing it...");
               return getAsyncClient().putKeyPolicy(keyPolicyRequest);
           })
           .thenApply(putPolicyResponse -> {
               logger.info("The key policy has been replaced.");
               return true;
           })
           .exceptionally(throwable -> {
               if (throwable.getCause() instanceof LimitExceededException) {
                   logger.error("Cannot replace policy, as only one policy is
allowed per key.");
```

```
                    return false;
                }
                throw new RuntimeException("Error replacing policy", throwable);
            });
    }


    /**
     * Asynchronously retrieves the key policy for the specified key ID and
policy name.
     *
     * @param keyId       the ID of the AWS KMS key for which to retrieve the
policy
     * @param policyName the name of the key policy to retrieve
     * @return a {@link CompletableFuture} that, when completed, contains the key
policy as a {@link String}
     */
    public CompletableFuture<String> getKeyPolicyAsync(String keyId, String
policyName) {
        GetKeyPolicyRequest policyRequest = GetKeyPolicyRequest.builder()
            .keyId(keyId)
            .policyName(policyName)
            .build();

        return getAsyncClient().getKeyPolicy(policyRequest)
            .thenApply(response -> {
                String policy = response.policy();
                logger.info("The response is: " + policy);
                return policy;
            })
            .exceptionally(ex -> {
                throw new RuntimeException("Failed to get key policy", ex);
            });
    }

    /**
     * Asynchronously signs and verifies data using AWS KMS.
     *
     * <p>The method performs the following steps:
     * <ol>
     *     <li>Creates an AWS KMS key with the specified key spec, key usage, and
origin.</li>
     *     <li>Signs the provided message using the created KMS key and the
RSASSA-PSS-SHA-256 algorithm.</li>
```

```
    *     <li>Verifies the signature of the message using the created KMS key
and the RSASSA-PSS-SHA-256 algorithm.</li>
    * </ol>
    *
    * @return a {@link CompletableFuture} that completes with the result of the
signature verification,
    *         {@code true} if the signature is valid, {@code false} otherwise.
    * @throws KmsException if any error occurs during the KMS operations.
    * @throws RuntimeException if an unexpected error occurs.
    */
   public CompletableFuture<Boolean> signVerifyDataAsync() {
       String signMessage = "Here is the message that will be digitally signed";

       // Create an AWS KMS key used to digitally sign data.
       CreateKeyRequest createKeyRequest = CreateKeyRequest.builder()
           .keySpec(KeySpec.RSA_2048)
           .keyUsage(KeyUsageType.SIGN_VERIFY)
           .origin(OriginType.AWS_KMS)
           .build();

       return getAsyncClient().createKey(createKeyRequest)
           .thenCompose(createKeyResponse -> {
               String keyId = createKeyResponse.keyMetadata().keyId();

               SdkBytes messageBytes = SdkBytes.fromString(signMessage,
Charset.defaultCharset());
               SignRequest signRequest = SignRequest.builder()
                   .keyId(keyId)
                   .message(messageBytes)
                   .signingAlgorithm(SigningAlgorithmSpec.RSASSA_PSS_SHA_256)
                   .build();

               return getAsyncClient().sign(signRequest)
                   .thenCompose(signResponse -> {
                       byte[] signedBytes =
signResponse.signature().asByteArray();

                       VerifyRequest verifyRequest = VerifyRequest.builder()
                           .keyId(keyId)

.message(SdkBytes.fromByteArray(signMessage.getBytes(Charset.defaultCharset())))

.signature(SdkBytes.fromByteBuffer(ByteBuffer.wrap(signedBytes)))
```

```
.signingAlgorithm(SigningAlgorithmSpec.RSASSA_PSS_SHA_256)
                            .build();

                        return getAsyncClient().verify(verifyRequest)
                            .thenApply(verifyResponse -> {
                                return (boolean) verifyResponse.signatureValid();
                            });
                    });
            })
            .exceptionally(throwable -> {
               throw new RuntimeException("Failed to sign or verify data",
throwable);
            });
    }

    /**
     * Asynchronously tags a KMS key with a specific tag.
     *
     * @param keyId the ID of the KMS key to be tagged
     * @return a {@link CompletableFuture} that completes when the tagging
operation is finished
     */
    public CompletableFuture<Void> tagKMSKeyAsync(String keyId) {
        Tag tag = Tag.builder()
            .tagKey("Environment")
            .tagValue("Production")
            .build();

        TagResourceRequest tagResourceRequest = TagResourceRequest.builder()
            .keyId(keyId)
            .tags(tag)
            .build();

        return getAsyncClient().tagResource(tagResourceRequest)
            .thenRun(() -> {
                logger.info("{} key was tagged", keyId);
            })
            .exceptionally(throwable -> {
                throw new RuntimeException("Failed to tag the KMS key",
throwable);
            });
    }
```

```
   /**
    * Deletes a specific KMS alias asynchronously.
    *
    * @param aliasName the name of the alias to be deleted
    * @return a {@link CompletableFuture} representing the asynchronous
operation of deleting the specified alias
    */
   public CompletableFuture<Void> deleteSpecificAliasAsync(String aliasName) {
       DeleteAliasRequest deleteAliasRequest = DeleteAliasRequest.builder()
           .aliasName(aliasName)
           .build();

       return getAsyncClient().deleteAlias(deleteAliasRequest)
           .thenRun(() -> {
               logger.info("Alias {} has been deleted successfully", aliasName);
           })
           .exceptionally(throwable -> {
               throw new RuntimeException("Failed to delete alias: " +
aliasName, throwable);
           });
   }

   /**
    * Asynchronously disables the specified AWS Key Management Service (KMS)
key.
    *
    * @param keyId the ID or Amazon Resource Name (ARN) of the KMS key to be
disabled
    * @return a CompletableFuture that, when completed, indicates that the key
has been disabled successfully
    */
   public CompletableFuture<Void> disableKeyAsync(String keyId) {
       DisableKeyRequest keyRequest = DisableKeyRequest.builder()
           .keyId(keyId)
           .build();

       return getAsyncClient().disableKey(keyRequest)
           .thenRun(() -> {
               logger.info("Key {} has been disabled successfully",keyId);
           })
           .exceptionally(throwable -> {
               throw new RuntimeException("Failed to disable key: " + keyId,
throwable);
           });
```

```
    }

    /**
     * Deletes a KMS key asynchronously.
     *
     * <p><strong>Warning:</strong> Deleting a KMS key is a destructive and
 potentially dangerous operation.
     * When a KMS key is deleted, all data that was encrypted under the KMS key
 becomes unrecoverable.
     * This means that any files, databases, or other data that were encrypted
 using the deleted KMS key
     * will become permanently inaccessible. Exercise extreme caution when
 deleting KMS keys.</p>
     *
     * @param keyId the ID of the KMS key to delete
     * @return a {@link CompletableFuture} that completes when the key deletion
 is scheduled
     */
    public CompletableFuture<Void> deleteKeyAsync(String keyId) {
        ScheduleKeyDeletionRequest deletionRequest =
 ScheduleKeyDeletionRequest.builder()
            .keyId(keyId)
            .pendingWindowInDays(7)
            .build();

        return getAsyncClient().scheduleKeyDeletion(deletionRequest)
            .thenRun(() -> {
                logger.info("Key {} will be deleted in 7 days", keyId);
            })
            .exceptionally(throwable -> {
                throw new RuntimeException("Failed to schedule key deletion for
 key ID: " + keyId, throwable);
            });
    }


    public String getAccountId(){
        try (StsClient stsClient = StsClient.create()){
            GetCallerIdentityResponse callerIdentity =
 stsClient.getCallerIdentity();
            return callerIdentity.account();
        }
    }
}
```

- 有关 API 详细信息，请参阅《Amazon SDK for Java 2.x API Reference》中的以下主题。
  - CreateAlias
  - CreateGrant
  - CreateKey
  - Decrypt
  - DescribeKey
  - DisableKey
  - EnableKey
  - Encrypt
  - GetKeyPolicy
  - ListAliases
  - ListGrants
  - ListKeys
  - RevokeGrant
  - ScheduleKeyDeletion
  - Sign
  - TagResource

PHP

适用于 PHP 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库 中查找完整示例，了解如何进行设置和运行。

```
        echo "\n";
        echo "-------------------------------------\n";
        echo <<<WELCOME
Welcome to the AWS Key Management Service SDK Basics scenario.
```

```
This program demonstrates how to interact with AWS Key Management Service using
 the AWS SDK for PHP (v3).
The AWS Key Management Service (KMS) is a secure and highly available service
 that allows you to create
and manage AWS KMS keys and control their use across a wide range of AWS services
 and applications.
KMS provides a centralized and unified approach to managing encryption keys,
 making it easier to meet your
data protection and regulatory compliance requirements.

This KMS Basics scenario creates two key types:
- A symmetric encryption key is used to encrypt and decrypt data.
- An asymmetric key used to digitally sign data.

Let's get started...\n
WELCOME;
        echo "-------------------------------------\n";
        $this->pressEnter();

        $this->kmsClient = new KmsClient([]);
        // Initialize the KmsService class with the client. This allows you to
 override any defaults in the client before giving it to the service class.
        $this->kmsService = new KmsService($this->kmsClient);

        // 1. Create a symmetric KMS key.
        echo "\n";
        echo "1. Create a symmetric KMS key.\n";
        echo "First, we will create a symmetric KMS key that is used to encrypt
 and decrypt data by invoking createKey().\n";
        $this->pressEnter();

        $key = $this->kmsService->createKey();
        $this->resources['symmetricKey'] = $key['KeyId'];
        echo "Created a customer key with ARN {$key['Arn']}.\n";
        $this->pressEnter();

        // 2. Enable a KMS key.
        echo "\n";
        echo "2. Enable a KMS key.\n";
        echo "By default when you create an AWS key, it is enabled. The code
 checks to
determine if the key is enabled. If it is not enabled, the code enables it.\n";
        $this->pressEnter();
```

```
        $keyInfo = $this->kmsService->describeKey($key['KeyId']);
        if(!$keyInfo['Enabled']){
            echo "The key was not enabled, so we will enable it.\n";
            $this->pressEnter();
            $this->kmsService->enableKey($key['KeyId']);
            echo "The key was successfully enabled.\n";
        }else{
            echo "The key was already enabled, so there was no need to enable it.
\n";
        }
        $this->pressEnter();

        // 3. Encrypt data using the symmetric KMS key.
        echo "\n";
        echo "3. Encrypt data using the symmetric KMS key.\n";
        echo "One of the main uses of symmetric keys is to encrypt and decrypt
 data.\n";
        echo "Next, we'll encrypt the string 'Hello, AWS KMS!' with the
 SYMMETRIC_DEFAULT encryption algorithm.\n";
        $this->pressEnter();
        $text = "Hello, AWS KMS!";
        $encryption = $this->kmsService->encrypt($key['KeyId'], $text);
        echo "The plaintext data was successfully encrypted with the algorithm:
 {$encryption['EncryptionAlgorithm']}.\n";
        $this->pressEnter();

        // 4. Create an alias.
        echo "\n";
        echo "4. Create an alias.\n";
        $aliasInput = testable_readline("Please enter an alias prefixed with
 \"alias/\" or press enter to use a default value: ");
        if($aliasInput == ""){
            $aliasInput = "alias/dev-encryption-key";
        }
        $this->kmsService->createAlias($key['KeyId'], $aliasInput);
        $this->resources['alias'] = $aliasInput;
        echo "The alias \"$aliasInput\" was successfully created.\n";
        $this->pressEnter();

        // 5. List all of your aliases.
        $aliasPageSize = 10;
        echo "\n";
        echo "5. List all of your aliases, up to $aliasPageSize.\n";
```

```
        $this->pressEnter();
        $aliasPaginator = $this->kmsService->listAliases();
        foreach($aliasPaginator as $pages){
            foreach($pages['Aliases'] as $alias){
                echo $alias['AliasName'] . "\n";
            }
            break;
        }
        $this->pressEnter();

        // 6. Enable automatic rotation of the KMS key.
        echo "\n";
        echo "6. Enable automatic rotation of the KMS key.\n";
        echo "By default, when the SDK enables automatic rotation of a KMS key,
KMS rotates the key material of the KMS key one year (approximately 365 days)
 from the enable date and every year
thereafter.";
        $this->pressEnter();
        $this->kmsService->enableKeyRotation($key['KeyId']);
        echo "The key's rotation was successfully set for key:
 {$key['KeyId']}\n";
        $this->pressEnter();

        // 7. Create a grant.
        echo "7. Create a grant.\n";
        echo "\n";
        echo "A grant is a policy instrument that allows Amazon Web Services
 principals to use KMS keys.
It also can allow them to view a KMS key (DescribeKey) and create and manage
 grants.
When authorizing access to a KMS key, grants are considered along with key
 policies and IAM policies.\n";
        $granteeARN = testable_readline("Please enter the Amazon Resource Name
 (ARN) of an Amazon Web Services principal. Valid principals include Amazon
 Web Services accounts, IAM users, IAM roles, federated users, and assumed
 role users. For help with the ARN syntax for a principal, see IAM ARNs in the
 Identity and Access Management User Guide. \nTo skip this step, press enter
 without any other values: ");
        if($granteeARN){
            $operations = [
                "ENCRYPT",
                "DECRYPT",
                "DESCRIBE_KEY",
            ];
```

```
            $grant = $this->kmsService->createGrant($key['KeyId'], $granteeARN,
$operations);
            echo "The grant Id is: {$grant['GrantId']}\n";
        }else{
            echo "Steps 7, 8, and 9 will be skipped.\n";
        }
        $this->pressEnter();

        // 8. List grants for the KMS key.
        if($granteeARN){
            echo "8. List grants for the KMS key.\n\n";
            $grantsPaginator = $this->kmsService->listGrants($key['KeyId']);
            foreach($grantsPaginator as $page){
                foreach($page['Grants'] as $grant){
                    echo $grant['GrantId'] . "\n";
                }
            }
        }else{
            echo "Skipping step 8...\n";
        }
        $this->pressEnter();

        // 9. Revoke the grant.
        if($granteeARN) {
            echo "\n";
            echo "9. Revoke the grant.\n";
            $this->pressEnter();
            $this->kmsService->revokeGrant($grant['GrantId'], $keyInfo['KeyId']);
            echo "{$grant['GrantId']} was successfully revoked!\n";
        }else{
            echo "Skipping step 9...\n";
        }
        $this->pressEnter();

        // 10. Decrypt the data.
        echo "\n";
        echo "10. Decrypt the data.\n";
        echo "Let's decrypt the data that was encrypted before.\n";
        echo "We'll use the same key to decrypt the string that we encrypted
earlier in the program.\n";
        $this->pressEnter();
        $decryption = $this->kmsService->decrypt($keyInfo['KeyId'],
$encryption['CiphertextBlob'], $encryption['EncryptionAlgorithm']);
        echo "The decrypted text is: {$decryption['Plaintext']}\n";
```

```
        $this->pressEnter();

        // 11. Replace a Key Policy.
        echo "\n";
        echo "11. Replace a Key Policy.\n";
        echo "A key policy is a resource policy for a KMS key. Key policies are
 the primary way to control access to KMS keys.\n";
        echo "Every KMS key must have exactly one key policy. The statements in
 the key policy determine who has permission to use the KMS key and how they can
 use it.\n";
        echo " You can also use IAM policies and grants to control access to the
 KMS key, but every KMS key must have a key policy.\n";
        echo "We will replace the key's policy with a new one:\n";
        $stsClient = new StsClient([]);
        $result = $stsClient->getCallerIdentity();
        $accountId = $result['Account'];
        $keyPolicy = <<< KEYPOLICY
{
    "Version":"2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Principal": {"AWS": "arn:aws:iam::$accountId:root"},
        "Action": "kms:*",
        "Resource": "*"
    }]
}
KEYPOLICY;
        echo $keyPolicy;
        $this->pressEnter();
        $this->kmsService->putKeyPolicy($keyInfo['KeyId'], $keyPolicy);
        echo "The Key Policy was successfully replaced!\n";
        $this->pressEnter();

        // 12. Retrieve the key policy.
        echo "\n";
        echo "12. Retrieve the key policy.\n";
        echo "Let's get some information about the new policy and print it to the
 screen.\n";
        $this->pressEnter();
        $policyInfo = $this->kmsService->getKeyPolicy($keyInfo['KeyId']);
        echo "We got the info! Here is the policy: \n";
        echo $policyInfo['Policy'] . "\n";
        $this->pressEnter();
```

```
        // 13. Create an asymmetric KMS key and sign data.
        echo "\n";
        echo "13. Create an asymmetric KMS key and sign data.\n";
        echo "Signing your data with an AWS key can provide several benefits that
make it an attractive option for your data signing needs.\n";
        echo "By using an AWS KMS key, you can leverage the security controls and
compliance features provided by AWS, which can help you meet various regulatory
requirements and enhance the overall security posture of your organization.\n";
        echo "First we'll create the asymmetric key.\n";
        $this->pressEnter();
        $keySpec = "RSA_2048";
        $keyUsage = "SIGN_VERIFY";
        $asymmetricKey = $this->kmsService->createKey($keySpec, $keyUsage);
        $this->resources['asymmetricKey'] = $asymmetricKey['KeyId'];
        echo "Created the key with ID: {$asymmetricKey['KeyId']}\n";
        echo "Next, we'll sign the data.\n";
        $this->pressEnter();
        $algorithm = "RSASSA_PSS_SHA_256";
        $sign = $this->kmsService->sign($asymmetricKey['KeyId'], $text,
$algorithm);
        $verify = $this->kmsService->verify($asymmetricKey['KeyId'], $text,
$sign['Signature'], $algorithm);
        echo "Signature verification result: {$sign['signature']}\n";
        $this->pressEnter();

        // 14. Tag the symmetric KMS key.
        echo "\n";
        echo "14. Tag the symmetric KMS key.\n";
        echo "By using tags, you can improve the overall management, security,
and governance of your KMS keys, making it easier to organize, track, and
control access to your encrypted data within your AWS environment.\n";
        echo "Let's tag our symmetric key as Environment->Production\n";
        $this->pressEnter();
        $this->kmsService->tagResource($key['KeyId'], [
            [
                'TagKey' => "Environment",
                'TagValue' => "Production",
            ],
        ]);
        echo "The key was successfully tagged!\n";
        $this->pressEnter();

        // 15. Schedule the deletion of the KMS key
        echo "\n";
```

```
        echo "15. Schedule the deletion of the KMS key.\n";
        echo "By default, KMS applies a waiting period of 30 days, but you can
 specify a waiting period of 7-30 days.\n";
        echo "When this operation is successful, the key state of the KMS key
 changes to PendingDeletion and the key can't be used in any cryptographic
 operations.\n";
        echo "It remains in this state for the duration of the waiting period.\n
\n";

        echo "Deleting a KMS key is a destructive and potentially dangerous
 operation. When a KMS key is deleted, all data that was encrypted under the KMS
 key is unrecoverable.\n\n";

        $cleanUp = testable_readline("Would you like to delete the resources
 created during this scenario, including the keys? (y/n): ");
        if($cleanUp == "Y" || $cleanUp == "y"){
            $this->cleanUp();
        }

        echo
 "--------------------------------------------------------------------------------
\n";
        echo "This concludes the AWS Key Management SDK Basics scenario\n";
        echo
 "--------------------------------------------------------------------------------
\n";



namespace Kms;

use Aws\Kms\Exception\KmsException;
use Aws\Kms\KmsClient;
use Aws\Result;
use Aws\ResultPaginator;
use AwsUtilities\AWSServiceClass;

class KmsService extends AWSServiceClass
{

    protected KmsClient $client;
    protected bool $verbose;

    /***
```

```
     * @param KmsClient|null $client
     * @param bool $verbose
     */
    public function __construct(KmsClient $client = null, bool $verbose = false)
    {
        $this->verbose = $verbose;
        if($client){
            $this->client = $client;
            return;
        }
        $this->client = new KmsClient([]);
    }


    /***
     * @param string $keySpec
     * @param string $keyUsage
     * @param string $description
     * @return array
     */
    public function createKey(string $keySpec = "", string $keyUsage = "", string
$description = "Created by the SDK for PHP")
    {
        $parameters = ['Description' => $description];
        if($keySpec && $keyUsage){
            $parameters['KeySpec'] = $keySpec;
            $parameters['KeyUsage'] = $keyUsage;
        }
        try {
            $result = $this->client->createKey($parameters);
            return $result['KeyMetadata'];
        }catch(KmsException $caught){
            // Check for error specific to createKey operations
            if ($caught->getAwsErrorMessage() == "LimitExceededException"){
                echo "The request was rejected because a quota was exceeded. For
more information, see Quotas in the Key Management Service Developer Guide.";
            }
            throw $caught;
        }
    }



    /***
```

```
     * @param string $keyId
     * @param string $ciphertext
     * @param string $algorithm
     * @return Result
     */
    public function decrypt(string $keyId, string $ciphertext, string $algorithm
 = "SYMMETRIC_DEFAULT")
    {
        try{
            return $this->client->decrypt([
                'CiphertextBlob' => $ciphertext,
                'EncryptionAlgorithm' => $algorithm,
                'KeyId' => $keyId,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem decrypting the data: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }



    /***
     * @param string $keyId
     * @param string $text
     * @return Result
     */
    public function encrypt(string $keyId, string $text)
    {
        try {
            return $this->client->encrypt([
                'KeyId' => $keyId,
                'Plaintext' => $text,
            ]);
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "DisabledException"){
                echo "The request was rejected because the specified KMS key is
 not enabled.\n";
            }
            throw $caught;
        }
    }
```

```
    /***
     * @param string $keyId
     * @param int $limit
     * @return ResultPaginator
     */
    public function listAliases(string $keyId = "", int $limit = 0)
    {
        $args = [];
        if($keyId){
            $args['KeyId'] = $keyId;
        }
        if($limit){
            $args['Limit'] = $limit;
        }
        try{
            return $this->client->getPaginator("ListAliases", $args);
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "InvalidMarkerException"){
                echo "The request was rejected because the marker that specifies
where pagination should next begin is not valid.\n";
            }
            throw $caught;
        }
    }



    /***
     * @param string $keyId
     * @param string $alias
     * @return void
     */
    public function createAlias(string $keyId, string $alias)
    {
        try{
            $this->client->createAlias([
                'TargetKeyId' => $keyId,
                'AliasName' => $alias,
            ]);
        }catch (KmsException $caught){
            if($caught->getAwsErrorMessage() == "InvalidAliasNameException"){
```

```
                echo "The request was rejected because the specified alias name
is not valid.";
            }
            throw $caught;
        }
    }



    /***
     * @param string $keyId
     * @param string $granteePrincipal
     * @param array $operations
     * @param array $grantTokens
     * @return Result
     */
    public function createGrant(string $keyId, string $granteePrincipal, array
$operations, array $grantTokens = [])
    {
        $args = [
            'KeyId' => $keyId,
            'GranteePrincipal' => $granteePrincipal,
            'Operations' => $operations,
        ];
        if($grantTokens){
            $args['GrantTokens'] = $grantTokens;
        }
        try{
            return $this->client->createGrant($args);
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "InvalidGrantTokenException"){
                echo "The request was rejected because the specified grant token
is not valid.\n";
            }
            throw $caught;
        }
    }



    /***
     * @param string $keyId
     * @return array
     */
```

```
    public function describeKey(string $keyId)
    {
        try {
            $result = $this->client->describeKey([
                "KeyId" => $keyId,
            ]);
            return $result['KeyMetadata'];
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "NotFoundException"){
                echo "The request was rejected because the specified entity or
 resource could not be found.\n";
            }
            throw $caught;
        }
    }



    /***
     * @param string $keyId
     * @return void
     */
    public function disableKey(string $keyId)
    {
        try {
            $this->client->disableKey([
                'KeyId' => $keyId,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem disabling the key: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }



    /***
     * @param string $keyId
     * @return void
     */
    public function enableKey(string $keyId)
    {
        try {
```

```
            $this->client->enableKey([
                'KeyId' => $keyId,
            ]);
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "NotFoundException"){
                echo "The request was rejected because the specified entity or
 resource could not be found.\n";
            }
            throw $caught;
        }
    }



    /***
     * @return array
     */
    public function listKeys()
    {
        try {
            $contents = [];
            $paginator = $this->client->getPaginator("ListKeys");
            foreach($paginator as $result){
                foreach ($result['Content'] as $object) {
                    $contents[] = $object;
                }
            }
            return $contents;
        }catch(KmsException $caught){
            echo "There was a problem listing the keys: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }



    /***
     * @param string $keyId
     * @return Result
     */
    public function listGrants(string $keyId)
    {
        try{
```

```
            return $this->client->listGrants([
                'KeyId' => $keyId,
            ]);
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "NotFoundException"){
                echo "    The request was rejected because the specified entity
 or resource could not be found.\n";
            }
            throw $caught;
        }
    }


    /***
     * @param string $keyId
     * @return Result
     */
    public function getKeyPolicy(string $keyId)
    {
        try {
            return $this->client->getKeyPolicy([
                'KeyId' => $keyId,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem getting the key policy: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }


    /***
     * @param string $grantId
     * @param string $keyId
     * @return void
     */
    public function revokeGrant(string $grantId, string $keyId)
    {
        try{
            $this->client->revokeGrant([
                'GrantId' => $grantId,
                'KeyId' => $keyId,
            ]);
        }catch(KmsException $caught){
```

```
            echo "There was a problem with revoking the grant: {$caught-
>getAwsErrorMessage()}.\n";
            throw $caught;
        }
    }



    /***
     * @param string $keyId
     * @param int $pendingWindowInDays
     * @return void
     */
    public function scheduleKeyDeletion(string $keyId, int $pendingWindowInDays =
 7)
    {
        try {
            $this->client->scheduleKeyDeletion([
                'KeyId' => $keyId,
                'PendingWindowInDays' => $pendingWindowInDays,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem scheduling the key deletion: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }



    /***
     * @param string $keyId
     * @param array $tags
     * @return void
     */
    public function tagResource(string $keyId, array $tags)
    {
        try {
            $this->client->tagResource([
                'KeyId' => $keyId,
                'Tags' => $tags,
            ]);
        }catch(KmsException $caught){
```

```
            echo "There was a problem applying the tag(s): {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }



    /***
     * @param string $keyId
     * @param string $message
     * @param string $algorithm
     * @return Result
     */
    public function sign(string $keyId, string $message, string $algorithm)
    {
        try {
            return $this->client->sign([
                'KeyId' => $keyId,
                'Message' => $message,
                'SigningAlgorithm' => $algorithm,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem signing the data: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }



    /***
     * @param string $keyId
     * @param int $rotationPeriodInDays
     * @return void
     */
    public function enableKeyRotation(string $keyId, int $rotationPeriodInDays =
 365)
    {
        try{
            $this->client->enableKeyRotation([
                'KeyId' => $keyId,
                'RotationPeriodInDays' => $rotationPeriodInDays,
            ]);
```

```
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "NotFoundException"){
                echo "The request was rejected because the specified entity or
 resource could not be found.\n";
            }
            throw $caught;
        }
    }



    /***
     * @param string $keyId
     * @param string $policy
     * @return void
     */
    public function putKeyPolicy(string $keyId, string $policy)
    {
        try {
            $this->client->putKeyPolicy([
                'KeyId' => $keyId,
                'Policy' => $policy,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem replacing the key policy: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }



    /***
     * @param string $aliasName
     * @return void
     */
    public function deleteAlias(string $aliasName)
    {
        try {
            $this->client->deleteAlias([
                'AliasName' => $aliasName,
            ]);
        }catch(KmsException $caught){
```

```
            echo "There was a problem deleting the alias: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }



    /***
     * @param string $keyId
     * @param string $message
     * @param string $signature
     * @param string $signingAlgorithm
     * @return bool
     */
    public function verify(string $keyId, string $message, string $signature,
 string $signingAlgorithm)
    {
        try {
            $result = $this->client->verify([
                'KeyId' => $keyId,
                'Message' => $message,
                'Signature' => $signature,
                'SigningAlgorithm' => $signingAlgorithm,
            ]);
            return $result['SignatureValid'];
        }catch(KmsException $caught){
            echo "There was a problem verifying the signature: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }


}
```

- 有关 API 详细信息，请参阅《适用于 PHP 的 Amazon SDK API Reference》中的以下主题。
  - CreateAlias
  - CreateGrant
  - CreateKey

- Decrypt
- DescribeKey
- DisableKey
- EnableKey
- Encrypt
- GetKeyPolicy
- ListAliases
- ListGrants
- ListKeys
- RevokeGrant
- ScheduleKeyDeletion
- Sign
- TagResource

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class KMSScenario:
    """Runs an interactive scenario that shows how to get started with KMS."""

    def __init__(
        self,
        key_manager: KeyManager,
        key_encryption: KeyEncrypt,
        alias_manager: AliasManager,
        grant_manager: GrantManager,
        key_policy: KeyPolicy,
    ):
        self.key_manager = key_manager
```

```
        self.key_encryption = key_encryption
        self.alias_manager = alias_manager
        self.grant_manager = grant_manager
        self.key_policy = key_policy
        self.key_id = ""
        self.alias_name = ""
        self.asymmetric_key_id = ""

    def kms_scenario(self):
        key_description = "Created by the AWS KMS API"

        print(DASHES)
        print(
            """
Welcome to the AWS Key Management SDK Basics scenario.

This program demonstrates how to interact with AWS Key Management using the AWS
 SDK for Python (Boto3).
The AWS Key Management Service (KMS) is a secure and highly available service
 that allows you to create
and manage AWS KMS keys and control their use across a wide range of AWS services
 and applications.
KMS provides a centralized and unified approach to managing encryption keys,
 making it easier to meet your
data protection and regulatory compliance requirements.

This Basics scenario creates two key types:

- A symmetric encryption key is used to encrypt and decrypt data.
- An asymmetric key used to digitally sign data.

Let's get started...
        """
        )
        q.ask("Press Enter to continue...")

        print(DASHES)
        print(f"1. Create a symmetric KMS key\n")
        print(
            f"First, the program will creates a symmetric KMS key that you can
 used to encrypt and decrypt data."
        )
        q.ask("Press Enter to continue...")
        self.key_id = self.key_manager.create_key(key_description)["KeyId"]
```

```
        print(f"A symmetric key was successfully created {self.key_id}.")
        q.ask("Press Enter to continue...")
        print(DASHES)
        print(
            """
2. Enable a KMS key

By default, when the SDK creates an AWS key, it is enabled. The next bit of code
 checks to
determine if the key is enabled.
        """
        )
        q.ask("Press Enter to continue...")
        is_enabled = self.is_key_enabled(self.key_id)
        print(f"Is the key enabled? {is_enabled}")
        if not is_enabled:
            self.key_manager.enable_key(self.key_id)
        q.ask("Press Enter to continue...")
        print(DASHES)
        print(f"3. Encrypt data using the symmetric KMS key")
        plain_text = "Hello, AWS KMS!"
        print(
            f"""
One of the main uses of symmetric keys is to encrypt and decrypt data.
Next, the code encrypts the string "{plain_text}" with the SYMMETRIC_DEFAULT
 encryption algorithm.
        """
        )
        q.ask("Press Enter to continue...")
        encrypted_text = self.key_encryption.encrypt(self.key_id, plain_text)
        print(DASHES)
        print(f"4. Create an alias")
        print(
            """
Now, the program will create an alias for the KMS key. An alias is a friendly
 name that you
can associate with a KMS key. The alias name should be prefixed with 'alias/'.
        """
        )
        alias_name = q.ask("Enter an alias name: ", q.non_empty)
        self.alias_manager.create_alias(self.key_id, alias_name)
        print(f"{alias_name} was successfully created.")
        self.alias_name = alias_name
        print(DASHES)
```

```
        print(f"5. List all of your aliases")
        q.ask("Press Enter to continue...")
        self.alias_manager.list_aliases(10)
        q.ask("Press Enter to continue...")
        print(DASHES)
        print(f"6. Enable automatic rotation of the KMS key")
        print(
            """

By default, when the SDK enables automatic rotation of a KMS key,
KMS rotates the key material of the KMS key one year (approximately 365 days)
 from the enable date and every year
thereafter.
        """
        )
        q.ask("Press Enter to continue...")
        self.key_manager.enable_key_rotation(self.key_id)
        print(DASHES)
        print(f"Key rotation has been enabled for key with id {self.key_id}")
        print(
            """
7. Create a grant

A grant is a policy instrument that allows Amazon Web Services principals to use
 KMS keys.
It also can allow them to view a KMS key (DescribeKey) and create and manage
 grants.
When authorizing access to a KMS key, grants are considered along with key
 policies and IAM policies.
        """
        )
        print(
            """
To create a grant you must specify a account_id. To specify the grantee
 account_id, use the Amazon Resource Name (ARN)
of an AWS account_id. Valid principals include AWS accounts, IAM users, IAM
 roles, federated users,
and assumed role users.
        """
        )
        account_id = q.ask(
            "Enter an account_id, or press enter to skip creating a grant... "
        )
        grant = None
```

```
        if account_id != "":
            grant = self.grant_manager.create_grant(
                self.key_id,
                account_id,
                [
                    "Encrypt",
                    "Decrypt",
                    "DescribeKey",
                ],
            )
            print(f"Grant created successfully with ID: {grant['GrantId']}")

        q.ask("Press Enter to continue...")
        print(DASHES)
        print(DASHES)
        print(f"8. List grants for the KMS key")
        q.ask("Press Enter to continue...")
        self.grant_manager.list_grants(self.key_id)
        q.ask("Press Enter to continue...")
        print(DASHES)
        print(f"9. Revoke the grant")
        print(
            """
The revocation of a grant immediately removes the permissions and access that the
 grant had provided.
This means that any account_id (user, role, or service) that was granted access
 to perform specific
KMS operations on a KMS key will no longer be able to perform those operations.
        """
        )
        q.ask("Press Enter to continue...")

        if grant is not None:
            self.grant_manager.revoke_grant(self.key_id, grant["GrantId"])
            print(f"Grant ID: {grant['GrantId']} was successfully revoked!")

        q.ask("Press Enter to continue...")
        print(DASHES)
        print(f"10. Decrypt the data\n")
        print(
            """
Lets decrypt the data that was encrypted in an early step.
The code uses the same key to decrypt the string that we encrypted earlier in the
 program.
```

```
        """
        )
        q.ask("Press Enter to continue...")
        decrypted_data = self.key_encryption.decrypt(self.key_id, encrypted_text)
        print(f"Data decrypted successfully for key ID: {self.key_id}")
        print(f"Decrypted data: {decrypted_data}")

        q.ask("Press Enter to continue...")
        print(DASHES)
        print(f"11. Replace a key policy\n")
        print(
            """
A key policy is a resource policy for a KMS key. Key policies are the primary way
 to control
access to KMS keys. Every KMS key must have exactly one key policy. The
 statements in the key policy
determine who has permission to use the KMS key and how they can use it.
You can also use IAM policies and grants to control access to the KMS key, but
 every KMS key
must have a key policy.

By default, when you create a key by using the SDK, a policy is created that
gives the AWS account that owns the KMS key full access to the KMS key.

Let's try to replace the automatically created policy with the following policy.
{
"Version":"2012-10-17",
"Statement": [{
"Effect": "Allow",
"Principal": {"AWS": "arn:aws:iam::0000000000:root"},
"Action": "kms:*",
"Resource": "*"
}]
}
        """
        )
        account_id = q.ask("Enter your account ID or press enter to skip: ")
        if account_id != "":
            policy = {
                "Version":"2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {"AWS": f"arn:aws:iam::{account_id}:root"},
```

```
                        "Action": "kms:*",
                        "Resource": "*",
                    }
                ],
            }

            self.key_policy.set_new_policy(self.key_id, policy)
            print("Key policy replacement succeeded.")
            q.ask("Press Enter to continue...")
        else:
            print("Skipping replacing the key policy.")

        print(DASHES)
        print(f"12. Get the key policy\n")
        print(
            f"The next bit of code that runs gets the key policy to make sure it
exists."
        )
        q.ask("Press Enter to continue...")
        policy = self.key_policy.get_policy(self.key_id)
        print(f"The key policy is: {policy}")

        q.ask("Press Enter to continue...")
        print(DASHES)
        print(f"13. Create an asymmetric KMS key and sign your data\n")
        print(
            """
        Signing your data with an AWS key can provide several benefits that make
it an attractive option
        for your data signing needs. By using an AWS KMS key, you can leverage
the
        security controls and compliance features provided by AWS,
        which can help you meet various regulatory requirements and enhance the
overall security posture
        of your organization.
        """
        )
        q.ask("Press Enter to continue...")
        print(f"Sign and verify data operation succeeded.")
        self.asymmetric_key_id = self.key_manager.create_asymmetric_key()
        message = "Here is the message that will be digitally signed"
        signature = self.key_encryption.sign(self.asymmetric_key_id, message)
        if self.key_encryption.verify(self.asymmetric_key_id, message,
signature):
```

```
            print("Signature verification succeeded.")
        else:
            print("Signature verification failed.")

        q.ask("Press Enter to continue...")
        print(DASHES)
        print(f"14. Tag your symmetric KMS Key\n")
        print(
            """
        By using tags, you can improve the overall management, security, and
 governance of your
        KMS keys, making it easier to organize, track, and control access to your
 encrypted data within
        your AWS environment
        """
        )
        q.ask("Press Enter to continue...")
        self.key_manager.tag_resource(self.key_id, "Environment", "Production")
        self.clean_up()

    def is_key_enabled(self, key_id: str) -> bool:
        """
        Check if the key is enabled or not.

        :param key_id: The key to check.
        :return: True if the key is enabled, otherwise False.
        """
        response = self.key_manager.describe_key(key_id)
        return response["Enabled"] is True

    def clean_up(self):
        """
        Delete resources created by this scenario.
        """
        if self.alias_name != "":
            print(f"Deleting the alias {self.alias_name}.")
            self.alias_manager.delete_alias(self.alias_name)
        window = 7  # The window in days for a scheduled deletion.
        if self.key_id != "":
            print(
                """
Warning:
Deleting a KMS key is a destructive and potentially dangerous operation. When a
 KMS key is deleted,
```

```
all data that was encrypted under the KMS key is unrecoverable.
                """
            )
            if q.ask(
                f"Do you want to delete the key with ID {self.key_id} (y/n)?",
                q.is_yesno,
            ):
                print(
                    f"The key {self.key_id} will be deleted with a window of
 {window} days. You can cancel the deletion before"
                )
                print("the window expires.")
                self.key_manager.delete_key(self.key_id, window)
                self.key_id = ""

        if self.asymmetric_key_id != "":
            if q.ask(
                f"Do you want to delete the asymmetric key with ID
 {self.asymmetric_key_id} (y/n)?",
                q.is_yesno,
            ):
                print(
                    f"The key {self.asymmetric_key_id} will be deleted with a
 window of {window} days. You can cancel the deletion before"
                )
                print("the window expires.")
                self.key_manager.delete_key(self.asymmetric_key_id, window)
                self.asymmetric_key_id = ""


if __name__ == "__main__":
    kms_scenario = None
    try:
        kms_client = boto3.client("kms")
        a_key_manager = KeyManager(kms_client)
        a_key_encrypt = KeyEncrypt(kms_client)
        an_alias_manager = AliasManager(kms_client)
        a_grant_manager = GrantManager(kms_client)
        a_key_policy = KeyPolicy(kms_client)
        kms_scenario = KMSScenario(
            key_manager=a_key_manager,
            key_encryption=a_key_encrypt,
            alias_manager=an_alias_manager,
            grant_manager=a_grant_manager,
```

```
            key_policy=a_key_policy,
        )
        kms_scenario.kms_scenario()
    except Exception:
        logging.exception("Something went wrong with the demo!")
        if kms_scenario is not None:
            kms_scenario.clean_up()
```

KMS 密钥管理的包装器类和方法。

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def create_key(self, key_description: str) -> dict[str, any]:
        """
        Creates a key with a user-provided description.

        :param key_description: A description for the key.
        :return: The key ID.
        """
        try:
            key = self.kms_client.create_key(Description=key_description)
["KeyMetadata"]
            self.created_keys.append(key)
            return key
        except ClientError as err:
            logging.error(
```

```
                "Couldn't create your key. Here's why: %s",
                err.response["Error"]["Message"],
            )
            raise


    def describe_key(self, key_id: str) -> dict[str, any]:
        """
        Describes a key.

        :param key_id: The ARN or ID of the key to describe.
        :return: Information about the key.
        """

        try:
            key = self.kms_client.describe_key(KeyId=key_id)["KeyMetadata"]
            return key
        except ClientError as err:
            logging.error(
                "Couldn't get key '%s'. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise


    def enable_key_rotation(self, key_id: str) -> None:
        """
        Enables rotation for a key.

        :param key_id: The ARN or ID of the key to enable rotation for.
        """
        try:
            self.kms_client.enable_key_rotation(KeyId=key_id)
        except ClientError as err:
            logging.error(
                "Couldn't enable rotation for key '%s'. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise


    def create_asymmetric_key(self) -> str:
```

```
        """
        Creates an asymmetric key in AWS KMS for signing messages.

        :return: The ID of the created key.
        """
        try:
            key = self.kms_client.create_key(
                KeySpec="RSA_2048", KeyUsage="SIGN_VERIFY", Origin="AWS_KMS"
            )["KeyMetadata"]
            self.created_keys.append(key)
            return key["KeyId"]
        except ClientError as err:
            logger.error(
                "Couldn't create your key. Here's why: %s",
                err.response["Error"]["Message"],
            )
            raise


    def tag_resource(self, key_id: str, tag_key: str, tag_value: str) -> None:
        """
        Add or edit tags on a customer managed key.

        :param key_id: The ARN or ID of the key to enable rotation for.
        :param tag_key: Key for the tag.
        :param tag_value: Value for the tag.
        """
        try:
            self.kms_client.tag_resource(
                KeyId=key_id, Tags=[{"TagKey": tag_key, "TagValue": tag_value}]
            )
        except ClientError as err:
            logging.error(
                "Couldn't add a tag for the key '%s'. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise


    def delete_key(self, key_id: str, window: int) -> None:
        """
        Deletes a list of keys.
```

```
        Warning:
        Deleting a KMS key is a destructive and potentially dangerous operation.
 When a KMS key is deleted,
        all data that was encrypted under the KMS key is unrecoverable.

        :param key_id: The ARN or ID of the key to delete.
        :param window: The waiting period, in days, before the KMS key is
 deleted.
        """

        try:
            self.kms_client.schedule_key_deletion(
                KeyId=key_id, PendingWindowInDays=window
            )
        except ClientError as err:
            logging.error(
                "Couldn't delete key %s. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise
```

KMS 密钥别名的包装器类和方法。

```
class AliasManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_key = None

    @classmethod
    def from_client(cls) -> "AliasManager":
        """
        Creates an AliasManager instance with a default KMS client.

        :return: An instance of AliasManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)
```

```
    def create_alias(self, key_id: str, alias: str) -> None:
        """
        Creates an alias for the specified key.

        :param key_id: The ARN or ID of a key to give an alias.
        :param alias: The alias to assign to the key.
        """
        try:
            self.kms_client.create_alias(AliasName=alias, TargetKeyId=key_id)
        except ClientError as err:
            if err.response["Error"]["Code"] == "AlreadyExistsException":
                logger.error(
                    "Could not create the alias %s because it already exists.",
key_id
                )
            else:
                logger.error(
                    "Couldn't encrypt text. Here's why: %s",
                    err.response["Error"]["Message"],
                )
                raise


    def list_aliases(self, page_size: int) -> None:
        """
        Lists aliases for the current account.
        :param page_size: The number of aliases to list per page.
        """
        try:
            alias_paginator = self.kms_client.get_paginator("list_aliases")
            for alias_page in alias_paginator.paginate(
                PaginationConfig={"PageSize": page_size}
            ):
                print(f"Here are {page_size} aliases:")
                pprint(alias_page["Aliases"])
                if alias_page["Truncated"]:
                    answer = input(
                        f"Do you want to see the next {page_size} aliases (y/n)?
"
                    )
                    if answer.lower() != "y":
                        break
                else:
                    print("That's all your aliases!")
```

```
        except ClientError as err:
            logging.error(
                "Couldn't list your aliases. Here's why: %s",
                err.response["Error"]["Message"],
            )
            raise


    def delete_alias(self, alias: str) -> None:
        """
        Deletes an alias.

        :param alias: The alias to delete.
        """
        try:
            self.kms_client.delete_alias(AliasName=alias)
        except ClientError as err:
            logger.error(
                "Couldn't delete alias %s. Here's why: %s",
                alias,
                err.response["Error"]["Message"],
            )
            raise
```

KMS 密钥加密的包装器类和方法。

```
class KeyEncrypt:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyEncrypt":
        """
        Creates a KeyEncrypt instance with a default KMS client.

        :return: An instance of KeyEncrypt initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)
```

```
    def encrypt(self, key_id: str, text: str) -> bytes:
        """
        Encrypts text by using the specified key.

        :param key_id: The ARN or ID of the key to use for encryption.
        :param text: The text to encrypt.
        :return: The encrypted version of the text.
        """
        try:
            response = self.kms_client.encrypt(KeyId=key_id,
Plaintext=text.encode())
            print(
                f"The string was encrypted with algorithm
{response['EncryptionAlgorithm']}"
            )
            return response["CiphertextBlob"]
        except ClientError as err:
            if err.response["Error"]["Code"] == "DisabledException":
                logger.error(
                    "Could not encrypt because the key %s is disabled.", key_id
                )
            else:
                logger.error(
                    "Couldn't encrypt text. Here's why: %s",
                    err.response["Error"]["Message"],
                )
            raise


    def decrypt(self, key_id: str, cipher_text: bytes) -> str:
        """
        Decrypts text previously encrypted with a key.

        :param key_id: The ARN or ID of the key used to decrypt the data.
        :param cipher_text: The encrypted text to decrypt.
        :return: The decrypted text.
        """
        try:
            return self.kms_client.decrypt(KeyId=key_id,
CiphertextBlob=cipher_text)[
                "Plaintext"
            ].decode()
```

```
        except ClientError as err:
            logger.error(
                "Couldn't decrypt your ciphertext. Here's why: %s",
                err.response["Error"]["Message"],
            )
            raise


    def sign(self, key_id: str, message: str) -> str:
        """
        Signs a message with a key.

        :param key_id: The ARN or ID of the key to use for signing.
        :param message: The message to sign.
        :return: The signature of the message.
        """
        try:
            return self.kms_client.sign(
                KeyId=key_id,
                Message=message.encode(),
                SigningAlgorithm="RSASSA_PSS_SHA_256",
            )["Signature"]
        except ClientError as err:
            logger.error(
                "Couldn't sign your message. Here's why: %s",
                err.response["Error"]["Message"],
            )
            raise


    def verify(self, key_id: str, message: str, signature: str) -> bool:
        """
        Verifies a signature against a message.

        :param key_id: The ARN or ID of the key used to sign the message.
        :param message: The message to verify.
        :param signature: The signature to verify.
        :return: True when the signature matches the message, otherwise False.
        """
        try:
            response = self.kms_client.verify(
                KeyId=key_id,
                Message=message.encode(),
                Signature=signature,
```

```
                SigningAlgorithm="RSASSA_PSS_SHA_256",
            )
            valid = response["SignatureValid"]
            print(f"The signature is {'valid' if valid else 'invalid'}.")
            return valid
        except ClientError as err:
            if err.response["Error"]["Code"] == "SignatureDoesNotMatchException":
                print("The signature is not valid.")
            else:
                logger.error(
                    "Couldn't verify your signature. Here's why: %s",
                    err.response["Error"]["Message"],
                )
            raise
```

KMS 密钥授权的包装器类和方法。

```
class GrantManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "GrantManager":
        """
        Creates a GrantManager instance with a default KMS client.

        :return: An instance of GrantManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def create_grant(
        self, key_id: str, principal: str, operations: [str]
    ) -> dict[str, str]:
        """
        Creates a grant for a key that lets a principal generate a symmetric data
        encryption key.

        :param key_id: The ARN or ID of the key.
```

```
        :param principal: The principal to grant permission to.
        :param operations: The operations to grant permission for.
        :return: The grant that is created.
        """
        try:
            return self.kms_client.create_grant(
                KeyId=key_id,
                GranteePrincipal=principal,
                Operations=operations,
            )
        except ClientError as err:
            logger.error(
                "Couldn't create a grant on key %s. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise


    def list_grants(self, key_id):
        """
        Lists grants for a key.

        :param key_id: The ARN or ID of the key to query.
        :return: The grants for the key.
        """
        try:
            paginator = self.kms_client.get_paginator("list_grants")
            grants = []
            page_iterator = paginator.paginate(KeyId=key_id)
            for page in page_iterator:
                grants.extend(page["Grants"])

            print(f"Grants for key {key_id}:")
            pprint(grants)
            return grants
        except ClientError as err:
            logger.error(
                "Couldn't list grants for key %s. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise
```

```
    def revoke_grant(self, key_id: str, grant_id: str) -> None:
        """
        Revokes a grant so that it can no longer be used.

        :param key_id: The ARN or ID of the key associated with the grant.
        :param grant_id: The ID of the grant to revoke.
        """
        try:
            self.kms_client.revoke_grant(KeyId=key_id, GrantId=grant_id)
        except ClientError as err:
            logger.error(
                "Couldn't revoke grant %s. Here's why: %s",
                grant_id,
                err.response["Error"]["Message"],
            )
            raise
```

KMS 密钥政策的包装器类和方法。

```
class KeyPolicy:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyPolicy":
        """
        Creates a KeyPolicy instance with a default KMS client.

        :return: An instance of KeyPolicy initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def set_new_policy(self, key_id: str, policy: dict[str, any]) -> None:
        """
        Sets the policy of a key. Setting a policy entirely overwrites the
 existing
```

```
        policy, so care is taken to add a statement to the existing list of
statements
        rather than simply writing a new policy.

        :param key_id: The ARN or ID of the key to set the policy to.
        :param policy: A new key policy. The key policy must allow the calling
principal to make a subsequent
                       PutKeyPolicy request on the KMS key. This reduces the risk
that the KMS key becomes unmanageable
        """

        try:
            self.kms_client.put_key_policy(KeyId=key_id,
Policy=json.dumps(policy))
        except ClientError as err:
            logger.error(
                "Couldn't set policy for key %s. Here's why %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise



    def get_policy(self, key_id: str) -> dict[str, str]:
        """
        Gets the policy of a key.

        :param key_id: The ARN or ID of the key to query.
        :return: The key policy as a dict.
        """
        if key_id != "":
            try:
                response = self.kms_client.get_key_policy(
                    KeyId=key_id,
                )
                policy = json.loads(response["Policy"])
            except ClientError as err:
                logger.error(
                    "Couldn't get policy for key %s. Here's why: %s",
                    key_id,
                    err.response["Error"]["Message"],
                )
                raise
```

```
            else:
                pprint(policy)
                return policy
        else:
            print("Skipping get policy demo.")
```

- 有关 API 详细信息，请参阅《Amazon SDK for Python (Boto3) API Reference》中的以下主题。
  - CreateAlias
  - CreateGrant
  - CreateKey
  - Decrypt
  - DescribeKey
  - DisableKey
  - EnableKey
  - Encrypt
  - GetKeyPolicy
  - ListAliases
  - ListGrants
  - ListKeys
  - RevokeGrant
  - ScheduleKeyDeletion
  - Sign
  - TagResource

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

用于 Amazon KMS 使用的操作 Amazon SDKs

以下代码示例演示了如何使用执行单个 Amazon KMS 操作 Amazon SDKs。每个示例都包含一个指向的链接 GitHub，您可以在其中找到有关设置和运行代码的说明。

这些摘录调用 Amazon KMS API，是必须在上下文中运行的大型程序的代码摘录。您可以在Amazon KMS 使用场景 Amazon SDKs 中结合上下文查看操作。

以下示例仅包括最常用的操作。有关完整列表，请参阅 Amazon Key Management Service API 参考。

示例

- CreateAlias与 Amazon SDK 或 CLI 配合使用
- CreateGrant与 Amazon SDK 或 CLI 配合使用
- CreateKey与 Amazon SDK 或 CLI 配合使用
- Decrypt与 Amazon SDK 或 CLI 配合使用
- DeleteAlias与 Amazon SDK 或 CLI 配合使用
- DescribeKey与 Amazon SDK 或 CLI 配合使用
- DisableKey与 Amazon SDK 或 CLI 配合使用
- EnableKey与 Amazon SDK 或 CLI 配合使用
- EnableKeyRotation与 Amazon SDK 或 CLI 配合使用
- Encrypt与 Amazon SDK 或 CLI 配合使用
- GenerateDataKey与 Amazon SDK 或 CLI 配合使用
- GenerateDataKeyWithoutPlaintext与 Amazon SDK 或 CLI 配合使用
- GenerateRandom与 Amazon SDK 或 CLI 配合使用
- GetKeyPolicy与 Amazon SDK 或 CLI 配合使用
- ListAliases与 Amazon SDK 或 CLI 配合使用
- ListGrants与 Amazon SDK 或 CLI 配合使用
- ListKeyPolicies与 Amazon SDK 或 CLI 配合使用
- ListKeys与 Amazon SDK 或 CLI 配合使用
- PutKeyPolicy与 Amazon SDK 或 CLI 配合使用
- ReEncrypt与 Amazon SDK 或 CLI 配合使用
- RetireGrant与 Amazon SDK 或 CLI 配合使用
- RevokeGrant与 Amazon SDK 或 CLI 配合使用
- ScheduleKeyDeletion与 Amazon SDK 或 CLI 配合使用
- Sign与 Amazon SDK 或 CLI 配合使用
- TagResource与 Amazon SDK 或 CLI 配合使用

- UpdateAlias与 Amazon SDK 或 CLI 配合使用
- Verify与 Amazon SDK 或 CLI 配合使用

`CreateAlias`与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 `CreateAlias`。

操作示例是大型程序的代码摘录，必须在上下文中运行。在以下代码示例中，您可以查看此操作的上下文：

- 了解基本功能

.NET

适用于 .NET 的 Amazon SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
using System;
using System.Threading.Tasks;
using Amazon.KeyManagementService;
using Amazon.KeyManagementService.Model;

/// <summary>
/// Creates an alias for an AWS Key Management Service (AWS KMS) key.
/// </summary>
public class CreateAlias
{
    public static async Task Main()
    {
        var client = new AmazonKeyManagementServiceClient();

        // The alias name must start with alias/ and can be
        // up to 256 alphanumeric characters long.
        var aliasName = "alias/ExampleAlias";
```

```
            // The value supplied as the TargetKeyId can be either
            // the key ID or key Amazon Resource Name (ARN) of the
            // AWS KMS key.
            var keyId = "1234abcd-12ab-34cd-56ef-1234567890ab";

            var request = new CreateAliasRequest
            {
                AliasName = aliasName,
                TargetKeyId = keyId,
            };

            var response = await client.CreateAliasAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                Console.WriteLine($"Alias, {aliasName}, successfully created.");
            }
            else
            {
                Console.WriteLine($"Could not create alias.");
            }
        }
    }
```

- 有关 API 的详细信息，请参阅 适用于 .NET 的 Amazon SDK API 参考CreateAlias中的。

CLI

Amazon CLI

为 KMS 密钥创建别名

以下 `create-alias` 命令为由密钥 ID `1234abcd-12ab-34cd-56ef-1234567890ab` 标识的 KMS 密钥创建名为 `example-alias` 的别名。

别名名称必须以 `alias/` 开头。请勿使用以开头的别名`alias/aws`；这些别名是保留给使用的Amazon。

```
aws kms create-alias \
    --alias-name alias/example-alias \
```

```
    --target-key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

此命令不返回任何输出。要查看新别名，请使用 `list-aliases` 命令。

有关更多信息，请参阅《Amazon 密钥管理服务开发人员指南》中的使用别名。

- 有关 API 的详细信息，请参阅Amazon CLI 命令参考CreateAlias中的。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
   /**
    * Creates a custom alias for the specified target key asynchronously.
    *
    * @param targetKeyId the ID of the target key for the alias
    * @param aliasName   the name of the alias to create
    * @return a {@link CompletableFuture} that completes when the alias creation
operation is finished
    */
   public CompletableFuture<Void> createCustomAliasAsync(String targetKeyId,
String aliasName) {
       CreateAliasRequest aliasRequest = CreateAliasRequest.builder()
           .aliasName(aliasName)
           .targetKeyId(targetKeyId)
           .build();

       CompletableFuture<CreateAliasResponse> responseFuture =
getAsyncClient().createAlias(aliasRequest);
       responseFuture.whenComplete((response, exception) -> {
           if (exception == null) {
               logger.info("{} was successfully created.", aliasName);
           } else {
               if (exception instanceof ResourceExistsException) {
                   logger.info("Alias [{}] already exists. Moving on...",
aliasName);
```

```
                } else if (exception instanceof KmsException kmsEx) {
                    throw new RuntimeException("KMS error occurred while creating
 alias: " + kmsEx.getMessage(), kmsEx);
                } else {
                    throw new RuntimeException("An unexpected error occurred
 while creating alias: " + exception.getMessage(), exception);
                }
            }
        });

        return responseFuture.thenApply(response -> null);
    }
```

- 有关 API 的详细信息，请参阅 Amazon SDK for Java 2.x API 参考CreateAlias中的。

Kotlin

适用于 Kotlin 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
suspend fun createCustomAlias(
    targetKeyIdVal: String?,
    aliasNameVal: String?,
) {
    val request =
        CreateAliasRequest {
            aliasName = aliasNameVal
            targetKeyId = targetKeyIdVal
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        kmsClient.createAlias(request)
        println("$aliasNameVal was successfully created")
    }
}
```

- 有关 API 的详细信息，请参阅适用CreateAlias于 K otlin 的Amazon SDK API 参考。

PHP

适用于 PHP 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /***
     * @param string $keyId
     * @param string $alias
     * @return void
     */
    public function createAlias(string $keyId, string $alias)
    {
        try{
            $this->client->createAlias([
                'TargetKeyId' => $keyId,
                'AliasName' => $alias,
            ]);
        }catch (KmsException $caught){
            if($caught->getAwsErrorMessage() == "InvalidAliasNameException"){
                echo "The request was rejected because the specified alias name
is not valid.";
            }
            throw $caught;
        }
    }
```

- 有关 API 的详细信息，请参阅 适用于 PHP 的 Amazon SDK API 参考CreateAlias中的。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class AliasManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_key = None

    @classmethod
    def from_client(cls) -> "AliasManager":
        """
        Creates an AliasManager instance with a default KMS client.

        :return: An instance of AliasManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def create_alias(self, key_id: str, alias: str) -> None:
        """
        Creates an alias for the specified key.

        :param key_id: The ARN or ID of a key to give an alias.
        :param alias: The alias to assign to the key.
        """
        try:
            self.kms_client.create_alias(AliasName=alias, TargetKeyId=key_id)
        except ClientError as err:
            if err.response["Error"]["Code"] == "AlreadyExistsException":
                logger.error(
                    "Could not create the alias %s because it already exists.",
 key_id
                )
```

```
            else:
                logger.error(
                    "Couldn't encrypt text. Here's why: %s",
                    err.response["Error"]["Message"],
                )
                raise
```

- 有关 API 的详细信息，请参阅适用CreateAlias于 Python 的Amazon SDK (Boto3) API 参考。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

`CreateGrant`与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 `CreateGrant`。

操作示例是大型程序的代码摘录，必须在上下文中运行。在以下代码示例中，您可以查看此操作的上下文：

- 了解基本功能

.NET

适用于 .NET 的 Amazon SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
        public static async Task Main()
        {
            var client = new AmazonKeyManagementServiceClient();

            // The identity that is given permission to perform the operations
            // specified in the grant.
            var grantee = "arn:aws:iam::111122223333:role/ExampleRole";
```

```
            // The identifier of the AWS KMS key to which the grant applies. You
            // can use the key ID or the Amazon Resource Name (ARN) of the KMS
key.
            var keyId = "7c9eccc2-38cb-4c4f-9db3-766ee8dd3ad4";

            var request = new CreateGrantRequest
            {
                GranteePrincipal = grantee,
                KeyId = keyId,

                // A list of operations that the grant allows.
                Operations = new List<string>
                {
                    "Encrypt",
                    "Decrypt",
                },
            };

            var response = await client.CreateGrantAsync(request);

            string grantId = response.GrantId; // The unique identifier of the
grant.
            string grantToken = response.GrantToken; // The grant token.

            Console.WriteLine($"Id: {grantId}, Token: {grantToken}");
        }
    }
```

- 有关 API 的详细信息，请参阅 适用于 .NET 的 Amazon SDK API 参考CreateGrant中的。

CLI

Amazon CLI

创建授权

以下 `create-grant` 示例将创建授权，以允许 `exampleUser` 用户对 `1234abcd-12ab-34cd-56ef-1234567890ab` KMS 密钥示例使用 `decrypt` 命令。停用主体是 `adminRole` 角色。该授权使用 `EncryptionContextSubset` 授权约束，以便仅在 `decrypt` 请求中的加密上下文包含 `"Department": "IT"` 键值对时才允许此权限。

```
aws kms create-grant \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --grantee-principal arn:aws:iam::123456789012:user/exampleUser \
    --operations Decrypt \
    --constraints EncryptionContextSubset={Department=IT} \
    --retiring-principal arn:aws:iam::123456789012:role/adminRole
```

输出：

```
{
    "GrantId":
 "1a2b3c4d2f5e69f440bae30eaec9570bb1fb7358824f9ddfa1aa5a0dab1a59b2",
    "GrantToken": "<grant token here>"
}
```

要查看有关授权的详细信息，请使用 `list-grants` 命令。

有关更多信息，请参阅《密Amazon 钥管理服务开发人员指南》中的 Amazon KMS 中的授权。

- 有关 API 的详细信息，请参阅Amazon CLI 命令参考CreateGrant中的。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /**
     * Grants permissions to a specified principal on a customer master key (CMK)
asynchronously.
     *
     * @param keyId             The unique identifier for the customer master key
(CMK) that the grant applies to.
     * @param granteePrincipal  The principal that is given permission to perform
the operations that the grant permits on the CMK.
     * @return A {@link CompletableFuture} that, when completed, contains the ID
of the created grant.
```

```
     * @throws RuntimeException If an error occurs during the grant creation
process.
     */
    public CompletableFuture<String> grantKeyAsync(String keyId, String
granteePrincipal) {
        List<GrantOperation> grantPermissions = List.of(
            GrantOperation.ENCRYPT,
            GrantOperation.DECRYPT,
            GrantOperation.DESCRIBE_KEY
        );

        CreateGrantRequest grantRequest = CreateGrantRequest.builder()
            .keyId(keyId)
            .name("grant1")
            .granteePrincipal(granteePrincipal)
            .operations(grantPermissions)
            .build();

        CompletableFuture<CreateGrantResponse> responseFuture =
getAsyncClient().createGrant(grantRequest);
        responseFuture.whenComplete((response, ex) -> {
            if (ex == null) {
                logger.info("Grant created successfully with ID: " +
response.grantId());
            } else {
                if (ex instanceof KmsException kmsEx) {
                    throw new RuntimeException("Failed to create grant: " +
kmsEx.getMessage(), kmsEx);
                } else {
                    throw new RuntimeException("An unexpected error occurred: " +
ex.getMessage(), ex);
                }
            }
        });

        return responseFuture.thenApply(CreateGrantResponse::grantId);
    }
```

- 有关 API 的详细信息，请参阅 Amazon SDK for Java 2.x API 参考CreateGrant中的。

Kotlin

适用于 Kotlin 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
suspend fun createNewGrant(
    keyIdVal: String?,
    granteePrincipalVal: String?,
    operation: String,
): String? {
    val operationOb = GrantOperation.fromValue(operation)
    val grantOperationList = ArrayList<GrantOperation>()
    grantOperationList.add(operationOb)

    val request =
        CreateGrantRequest {
            keyId = keyIdVal
            granteePrincipal = granteePrincipalVal
            operations = grantOperationList
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.createGrant(request)
        return response.grantId
    }
}
```

- 有关 API 的详细信息，请参阅适用CreateGrant于 K otlin 的Amazon SDK API 参考。

PHP

适用于 PHP 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /***
     * @param string $keyId
     * @param string $granteePrincipal
     * @param array $operations
     * @param array $grantTokens
     * @return Result
     */
    public function createGrant(string $keyId, string $granteePrincipal, array
$operations, array $grantTokens = [])
    {
        $args = [
            'KeyId' => $keyId,
            'GranteePrincipal' => $granteePrincipal,
            'Operations' => $operations,
        ];
        if($grantTokens){
            $args['GrantTokens'] = $grantTokens;
        }
        try{
            return $this->client->createGrant($args);
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "InvalidGrantTokenException"){
                echo "The request was rejected because the specified grant token
is not valid.\n";
            }
            throw $caught;
        }
    }
```

- 有关 API 的详细信息，请参阅 适用于 PHP 的 Amazon SDK API 参考CreateGrant中的。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class GrantManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "GrantManager":
        """
        Creates a GrantManager instance with a default KMS client.

        :return: An instance of GrantManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def create_grant(
        self, key_id: str, principal: str, operations: [str]
    ) -> dict[str, str]:
        """
        Creates a grant for a key that lets a principal generate a symmetric data
        encryption key.

        :param key_id: The ARN or ID of the key.
        :param principal: The principal to grant permission to.
        :param operations: The operations to grant permission for.
        :return: The grant that is created.
        """
        try:
            return self.kms_client.create_grant(
```

```
                KeyId=key_id,
                GranteePrincipal=principal,
                Operations=operations,
            )
        except ClientError as err:
            logger.error(
                "Couldn't create a grant on key %s. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise
```

- 有关 API 的详细信息，请参阅适用CreateGrant于 Python 的Amazon SDK (Boto3) API 参考。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

`CreateKey`与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 `CreateKey`。

操作示例是大型程序的代码摘录，必须在上下文中运行。您可以在以下代码示例中查看此操作的上下文：

- 了解基本功能
- 处理表加密

.NET

适用于 .NET 的 Amazon SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    using System;
    using System.Threading.Tasks;
    using Amazon.KeyManagementService;
    using Amazon.KeyManagementService.Model;

    /// <summary>
    /// Shows how to create a new AWS Key Management Service (AWS KMS)
    /// key.
    /// </summary>
    public class CreateKey
    {
        public static async Task Main()
        {
            // Note that if you need to create a Key in an AWS Region
            // other than the Region defined for the default user, you need to
            // pass the Region to the client constructor.
            var client = new AmazonKeyManagementServiceClient();

            // The call to CreateKeyAsync will create a symmetrical AWS KMS
            // key. For more information about symmetrical and asymmetrical
            // keys, see:
            //
            // https://docs.aws.amazon.com/kms/latest/developerguide/symm-asymm-
choose.html
            var response = await client.CreateKeyAsync(new CreateKeyRequest());

            // The KeyMetadata object contains information about the new AWS KMS
 key.
            KeyMetadata keyMetadata = response.KeyMetadata;

            if (keyMetadata is not null)
            {
                Console.WriteLine($"KMS Key: {keyMetadata.KeyId} was successfully
 created.");
            }
            else
            {
                Console.WriteLine("Could not create KMS Key.");
            }
        }
    }
```

- 有关 API 的详细信息，请参阅 适用于 .NET 的 Amazon SDK API 参考CreateKey中的。

CLI

Amazon CLI

示例 1：在 KMS 中创建客户托管的 KM Amazon S 密钥

以下 create-key 示例创建对称加密 KMS 密钥。

要创建基本 KMS 密钥（对称加密密钥），您无需指定任何参数。这些参数的默认值会创建对称加密密钥。

由于此命令未指定密钥策略，因此，KMS 密钥将获得以编程方式创建的 KMS 密钥的默认密钥策略。要查看密钥策略，请使用 get-key-policy 命令。要更改密钥策略，请使用 put-key-policy 命令。

```
aws kms create-key
```

create-key 命令会返回密钥元数据，包括新 KMS 密钥的密钥 ID 和 ARN。在其他 KMS 操作中，您可以使用这些值来识别 Amazon KMS 密钥。输出不包括标签。要查看 KMS 密钥的标签，请使用 list-resource-tags command。

输出：

```
{
    "KeyMetadata": {
        "AWSAccountId": "111122223333",
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "CreationDate": "2017-07-05T14:04:55-07:00",
        "CurrentKeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "Description": "",
        "Enabled": true,
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "MultiRegion": false,
```

```
        "Origin": "AWS_KMS"
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

注意：create-key 命令不允许您指定别名。要为新 KMS 密钥创建别名，请使用 create-alias 命令。

有关更多信息，请参阅《Amazon 密钥管理服务开发人员指南》中的创建密钥。

示例 2：创建用于加密和解密的非对称 RSA KMS 密钥

以下 create-key 示例创建了一个 KMS 密钥，其中包含用于加密和解密的非对称 RSA 密钥对。创建密钥后，无法更改密钥规格和密钥用法：

```
aws kms create-key \
   --key-spec RSA_4096 \
   --key-usage ENCRYPT_DECRYPT
```

输出：

```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "CreationDate": "2021-04-05T14:04:55-07:00",
        "CustomerMasterKeySpec": "RSA_4096",
        "Description": "",
        "Enabled": true,
        "EncryptionAlgorithms": [
            "RSAES_OAEP_SHA_1",
            "RSAES_OAEP_SHA_256"
        ],
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "RSA_4096",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "MultiRegion": false,
        "Origin": "AWS_KMS"
```

```
    }
}
```

有关更多信息，请参阅《密钥管理服务开发人员指南》中的 Amazon KMS 中的非对称 Amazon密钥。

示例 3：创建用于签名和验证的非对称椭圆曲线 KMS 密钥

创建用于签名和验证的非对称 KMS 密钥，其中包含非对称椭圆曲线（ECC）密钥对。尽管 SIGN_VERIFY 是 ECC KMS 密钥的唯一有效值，但 --key-usage 参数也是必需的。创建密钥后，无法更改密钥规格和密钥用法：

```
aws kms create-key \
    --key-spec ECC_NIST_P521 \
    --key-usage SIGN_VERIFY
```

输出：

```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "CreationDate": "2019-12-02T07:48:55-07:00",
        "CustomerMasterKeySpec": "ECC_NIST_P521",
        "Description": "",
        "Enabled": true,
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "ECC_NIST_P521",
        "KeyState": "Enabled",
        "KeyUsage": "SIGN_VERIFY",
        "MultiRegion": false,
        "Origin": "AWS_KMS",
        "SigningAlgorithms": [
            "ECDSA_SHA_512"
        ]
    }
}
```

有关更多信息，请参阅《密钥管理服务开发人员指南》中的 Amazon KMS 中的非对称 Amazon密钥。

示例 4：创建用于签名和验证的非对称 ML-DSA KMS 密钥

此示例创建了一个用于签名和验证的模格数字签名算法（ML-DSA）密钥。即使 SIGN_VERIFY 是 ML-DSA 密钥的唯一有效值，key-usage 参数也是必需的。

```
aws kms create-key \
    --key-spec ML_DSA_65 \
    --key-usage SIGN_VERIFY
```

输出：

```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "CreationDate": "2019-12-02T07:48:55-07:00",
        "Description": "",
        "Enabled": true,
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "ML_DSA_65",
        "KeyState": "Enabled",
        "KeyUsage": "SIGN_VERIFY",
        "MultiRegion": false,
        "Origin": "AWS_KMS",
        "SigningAlgorithms": [
            "ML_DSA_SHAKE_256"
        ]
    }
}
```

有关更多信息，请参阅《密钥管理服务开发人员指南》中的 Amazon KMS 中的非对称 Amazon密钥。

示例 5：创建 HMAC KMS 密钥

以下 create-key 示例将创建 384 位 HMAC KMS 密钥。尽管 --key-usage 参数的 GENERATE_VERIFY_MAC 值是 HMAC KMS 密钥的唯一有效值，但该值也是必需的。

```
aws kms create-key \
    --key-spec HMAC_384 \
```

```
    --key-usage GENERATE_VERIFY_MAC
```

输出：

```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
        "CreationDate": "2022-04-05T14:04:55-07:00",
        "CustomerMasterKeySpec": "HMAC_384",
        "Description": "",
        "Enabled": true,
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "HMAC_384",
        "KeyState": "Enabled",
        "KeyUsage": "GENERATE_VERIFY_MAC",
        "MacAlgorithms": [
            "HMAC_SHA_384"
        ],
        "MultiRegion": false,
        "Origin": "AWS_KMS"
    }
}
```

有关更多信息，请参阅《密钥管理服务开发人员指南》中的 Amazon KMS 中的 HMA C Amazon 密钥。

示例 6：创建多区域主 KMS 密钥

以下 `create-key` 示例创建多区域主对称加密密钥。由于所有参数的默认值都会创建对称加密密钥，因此，此 KMS 密钥只需要 `--multi-region` 参数。在 Amazon CLI 中，要指示布尔参数为真，只需指定参数名称即可。

```
aws kms create-key \
    --multi-region
```

输出：

```
{
    "KeyMetadata": {
```

```
        "Arn": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef12345678990ab",
        "AWSAccountId": "111122223333",
        "CreationDate": "2021-09-02T016:15:21-09:00",
        "CurrentKeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "Description": "",
        "Enabled": true,
        "EncryptionAlgorithms": [
          "SYMMETRIC_DEFAULT"
        ],
        "KeyId": "mrk-1234abcd12ab34cd56ef12345678990ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "MultiRegion": true,
        "MultiRegionConfiguration": {
            "MultiRegionKeyType": "PRIMARY",
            "PrimaryKey": {
                "Arn": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef12345678990ab",
                "Region": "us-west-2"
            },
            "ReplicaKeys": []
        },
        "Origin": "AWS_KMS"
    }
}
```

有关更多信息，请参阅《密钥管理服务开发人员指南》中的 Amazon KMS 中的非对称 Amazon密钥。

示例 7：为导入的密钥材料创建 KMS 密钥

以下 create-key 示例创建一个不带密钥材料的 KMS 密钥。操作完成后，您可以将自己的密钥材料导入 KMS 密钥。要创建此 KMS 密钥，请将 --origin 参数设置为 EXTERNAL。

```
aws kms create-key \
    --origin EXTERNAL
```

输出：

```
{
     "KeyMetadata": {
         "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
         "AWSAccountId": "111122223333",
         "CreationDate": "2019-12-02T07:48:55-07:00",
         "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
         "Description": "",
         "Enabled": false,
         "EncryptionAlgorithms": [
             "SYMMETRIC_DEFAULT"
         ],
         "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
         "KeyManager": "CUSTOMER",
         "KeySpec": "SYMMETRIC_DEFAULT",
         "KeyState": "PendingImport",
         "KeyUsage": "ENCRYPT_DECRYPT",
         "MultiRegion": false,
         "Origin": "EXTERNAL"
     }
 }
```

有关更多信息，请参阅《密钥管理服务开发人员指南》中的在 Amazon KMS 密Amazon 钥中导入密钥材料。

示例 6：在 Amazon CloudHSM 密钥存储库中创建 KMS 密钥

以下`create-key`示例在指定的 Amazon CloudHSM 密钥存储中创建一个 KMS 密钥。该操作在 KMS 中 Amazon 创建 KMS 密钥及其元数据，并在与自定义密钥存储库关联的 Amazon CloudHSM 集群中创建密钥材料。`--custom-key-store-id` 和 `--origin` 参数是必需的。

```
aws kms create-key \
    --origin AWS_CLOUDHSM \
    --custom-key-store-id cks-1234567890abcdef0
```

输出：

```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
```

```
        "CloudHsmClusterId": "cluster-1a23b4cdefg",
        "CreationDate": "2019-12-02T07:48:55-07:00",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "CustomKeyStoreId": "cks-1234567890abcdef0",
        "Description": "",
        "Enabled": true,
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "MultiRegion": false,
        "Origin": "AWS_CLOUDHSM"
    }
}
```

有关更多信息，请参阅《Amazon 密钥管理服务开发人员指南》中的 Amazon CloudHSM 密钥存储。

示例 8：在外部密钥存储中创建 KMS 密钥

以下 create-key 示例在指定的外部密钥存储中创建一个 KMS 密钥。在此命令中需要使用 --custom-key-store-id、--origin 和 --xks-key-id 参数。

--xks-key-id 参数指定外部密钥管理器中现有对称加密密钥的 ID。此密钥用作 KMS 密钥的外部密钥材料。--origin 参数的值必须为 EXTERNAL_KEY_STORE。custom-key-store-id 参数必须识别连接到其外部密钥存储代理的外部密钥存储。

```
aws kms create-key \
    --origin EXTERNAL_KEY_STORE \
    --custom-key-store-id cks-9876543210fedcba9 \
    --xks-key-id bb8562717f809024
```

输出：

```
{
    "KeyMetadata": {
        "Arn": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "AWSAccountId": "111122223333",
```

```
        "CreationDate": "2022-12-02T07:48:55-07:00",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "CustomKeyStoreId": "cks-9876543210fedcba9",
        "Description": "",
        "Enabled": true,
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeySpec": "SYMMETRIC_DEFAULT",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "MultiRegion": false,
        "Origin": "EXTERNAL_KEY_STORE",
        "XksKeyConfiguration": {
            "Id": "bb8562717f809024"
        }
    }
}
```

有关更多信息，请参阅《Amazon 密钥管理服务开发人员指南》中的外部密钥存储。

- 有关 API 的详细信息，请参阅Amazon CLI 命令参考CreateKey中的。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /**
     * Creates a new symmetric encryption key asynchronously.
     *
     * @param keyDesc the description of the key to be created
     * @return a {@link CompletableFuture} that completes with the ID of the
 newly created key
     * @throws RuntimeException if an error occurs while creating the key
```

```
     */
    public CompletableFuture<String> createKeyAsync(String keyDesc) {
        CreateKeyRequest keyRequest = CreateKeyRequest.builder()
            .description(keyDesc)
            .keySpec(KeySpec.SYMMETRIC_DEFAULT)
            .keyUsage(KeyUsageType.ENCRYPT_DECRYPT)
            .build();

        return getAsyncClient().createKey(keyRequest)
            .thenApply(resp -> resp.keyMetadata().keyId())
            .exceptionally(ex -> {
                throw new RuntimeException("An error occurred while creating the
 key: " + ex.getMessage(), ex);
            });
    }
```

- 有关 API 的详细信息，请参阅 Amazon SDK for Java 2.x API 参考CreateKey中的。

Kotlin

适用于 Kotlin 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
suspend fun createKey(keyDesc: String?): String? {
    val request =
        CreateKeyRequest {
            description = keyDesc
            customerMasterKeySpec = CustomerMasterKeySpec.SymmetricDefault
            keyUsage = KeyUsageType.fromValue("ENCRYPT_DECRYPT")
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val result = kmsClient.createKey(request)
        println("Created a customer key with id " + result.keyMetadata?.arn)
        return result.keyMetadata?.keyId
    }
```

```
}
```

- 有关 API 的详细信息，请参阅适用CreateKey于 K otlin 的Amazon SDK API 参考。

PHP

适用于 PHP 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /***
     * @param string $keySpec
     * @param string $keyUsage
     * @param string $description
     * @return array
     */
    public function createKey(string $keySpec = "", string $keyUsage = "", string
$description = "Created by the SDK for PHP")
    {
        $parameters = ['Description' => $description];
        if($keySpec && $keyUsage){
            $parameters['KeySpec'] = $keySpec;
            $parameters['KeyUsage'] = $keyUsage;
        }
        try {
            $result = $this->client->createKey($parameters);
            return $result['KeyMetadata'];
        }catch(KmsException $caught){
            // Check for error specific to createKey operations
            if ($caught->getAwsErrorMessage() == "LimitExceededException"){
                echo "The request was rejected because a quota was exceeded. For
more information, see Quotas in the Key Management Service Developer Guide.";
            }
            throw $caught;
        }
    }
```

- 有关 API 的详细信息，请参阅 适用于 PHP 的 Amazon SDK API 参考CreateKey中的。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def create_key(self, key_description: str) -> dict[str, any]:
        """
        Creates a key with a user-provided description.

        :param key_description: A description for the key.
        :return: The key ID.
        """
        try:
            key = self.kms_client.create_key(Description=key_description)
["KeyMetadata"]
```

```
            self.created_keys.append(key)
            return key
        except ClientError as err:
            logging.error(
                "Couldn't create your key. Here's why: %s",
                err.response["Error"]["Message"],
            )
            raise
```

- 有关 API 的详细信息，请参阅适用CreateKey于 Python 的Amazon SDK (Boto3) API 参考。

Ruby

适用于 Ruby 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
require 'aws-sdk-kms' # v2: require 'aws-sdk'

# Create a AWS KMS key.
# As long we are only encrypting small amounts of data (4 KiB or less) directly,
# a KMS key is fine for our purposes.
# For larger amounts of data,
# use the KMS key to encrypt a data encryption key (DEK).

client = Aws::KMS::Client.new

resp = client.create_key({
                           tags: [
                             {
                               tag_key: 'CreatedBy',
                               tag_value: 'ExampleUser'
                             }
                           ]
                         })
```

```
puts resp.key_metadata.key_id
```

- 有关 API 的详细信息，请参阅 适用于 Ruby 的 Amazon SDK API 参考CreateKey中的。

Rust

适用于 Rust 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
async fn make_key(client: &Client) -> Result<(), Error> {
    let resp = client.create_key().send().await?;

    let id = resp.key_metadata.as_ref().unwrap().key_id();

    println!("Key: {}", id);

    Ok(())
}
```

- 有关 API 的详细信息，请参阅适用CreateKey于 Rust 的Amazon SDK API 参考。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

`Decrypt`与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 `Decrypt`。

操作示例是大型程序的代码摘录，必须在上下文中运行。在以下代码示例中，您可以查看此操作的上下文：

- 了解基本功能

CLI

Amazon CLI

示例 1：使用对称 KMS 密钥解密加密消息（Linux 和 macOS）

以下decrypt命令示例演示了使用 CLI Amazon 解密数据的推荐方法。此版本演示了如何使用对称 KMS 密钥解密数据。

在文件中提供加密文字。在 --ciphertext-blob 参数的值中，使用 fileb:// 前缀，它将指示 CLI 从二进制文件中读取数据。如果文件不在当前目录中，请键入文件的完整路径。有关从文件读取 Amazon CLI 参数值的更多信息，请参阅Amazon 命令行界面用户指南中的从文件中加载 Amazon CLI 参数 < https://docs.aws.amazon.com/cli/ latest/userguide/cli-usage-parameters-file .html> 和Amazon 命令行工具博客中的本地文件参数最佳实践 < dev https://aws.amazon.com/blogs/ eloper/ best-practices-for-local-file-parameters/>。指定 KMS 密钥来解密密文。使用对称 KMS 密钥解密时不需要参数。--key-id Amazon KMS 可以从密文中的元数据中获取用于加密数据的 KMS 密钥的密钥 ID。但是，指定您正在使用的 KMS 密钥始终是最佳实践。此做法可确保您使用预期的 KMS 密钥，并防止您意外使用不信任的 KMS 密钥解密加密文字。以文本值请求明文输出。--query 参数指示 CLI 仅从输出中获取 Plaintext 字段的值。--output 参数以 text.base64 解码格式返回明文输出并将其保存在文件中。以下示例将 Plaintext 参数的值传送（|）给 Base64 实用程序，该程序负责对其进行解码。然后，它将解码后的输出重定向（>）到 ExamplePlaintext 文件。

在运行此命令之前，请将示例密钥 ID 替换为 Amazon 账户中的有效密钥 ID。

```
aws kms decrypt \
    --ciphertext-blob fileb://ExampleEncryptedFile \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --output text \
    --query Plaintext | base64 \
    --decode > ExamplePlaintextFile
```

此命令不生成任何输出。decrypt 命令的输出经过 base64 解码并保存在文件中。

有关更多信息，请参阅《Amazon 密钥管理服务 API Reference》中的解密。

示例 2：使用对称 KMS 密钥解密加密消息（Windows 命令提示符）

以下示例与前一个示例相同，不同之处在于，它使用 certutil 实用程序对明文数据进行 Base64 解码。此过程需要两个命令，如以下示例所示。

在运行此命令之前，请将示例密钥 ID 替换为 Amazon 账户中的有效密钥 ID。

```
aws kms decrypt ^
    --ciphertext-blob fileb://ExampleEncryptedFile ^
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab ^
    --output text ^
    --query Plaintext > ExamplePlaintextFile.base64
```

运行 certutil 命令。

```
certutil -decode ExamplePlaintextFile.base64 ExamplePlaintextFile
```

输出：

```
Input Length = 18
Output Length = 12
CertUtil: -decode command completed successfully.
```

有关更多信息，请参阅《Amazon 密钥管理服务 API Reference》中的解密。

示例 3：使用非对称 KMS 密钥解密加密消息（Linux 和 macOS）

以下 decrypt 命令示例演示如何解密在 RSA 非对称 KMS 密钥下加密的数据。

使用非对称 KMS 密钥时，需要 encryption-algorithm 参数，以指定用于加密明文的算法。

在运行此命令之前，请将示例密钥 ID 替换为 Amazon 账户中的有效密钥 ID。

```
aws kms decrypt \
    --ciphertext-blob fileb://ExampleEncryptedFile \
    --key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \
    --encryption-algorithm RSAES_OAEP_SHA_256 \
    --output text \
    --query Plaintext | base64 \
    --decode > ExamplePlaintextFile
```

此命令不生成任何输出。decrypt 命令的输出经过 base64 解码并保存在文件中。

有关更多信息，请参阅《密钥管理服务开发人员指南》中的 Amazon KMS 中的非对称 Amazon密钥。

- 有关 API 详细信息，请参阅《Amazon CLI 命令参考》中的解密。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
   /**
    * Asynchronously decrypts the given encrypted data using the specified key
ID.
    *
    * @param encryptedData The encrypted data to be decrypted.
    * @param keyId The ID of the key to be used for decryption.
    * @return A CompletableFuture that, when completed, will contain the
decrypted data as a String.
    *         If an error occurs during the decryption process, the
CompletableFuture will complete
    *         exceptionally with the error, and the method will return an empty
String.
    */
   public CompletableFuture<String> decryptDataAsync(SdkBytes encryptedData,
String keyId) {
       DecryptRequest decryptRequest = DecryptRequest.builder()
           .ciphertextBlob(encryptedData)
           .keyId(keyId)
           .build();

       CompletableFuture<DecryptResponse> responseFuture =
getAsyncClient().decrypt(decryptRequest);
       responseFuture.whenComplete((decryptResponse, exception) -> {
           if (exception == null) {
               logger.info("Data decrypted successfully for key ID: " + keyId);
           } else {
               if (exception instanceof KmsException kmsEx) {
                   throw new RuntimeException("KMS error occurred while
decrypting data: " + kmsEx.getMessage(), kmsEx);
               } else {
```

```
                    throw new RuntimeException("An unexpected error occurred
 while decrypting data: " + exception.getMessage(), exception);
                }
            }
        });

        return responseFuture.thenApply(decryptResponse ->
 decryptResponse.plaintext().asString(StandardCharsets.UTF_8));
    }
```

- 有关 API 详细信息，请参阅《Amazon SDK for Java 2.x API Reference》中的 Decrypt。

Kotlin

适用于 Kotlin 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
suspend fun encryptData(keyIdValue: String): ByteArray? {
    val text = "This is the text to encrypt by using the AWS KMS Service"
    val myBytes: ByteArray = text.toByteArray()

    val encryptRequest =
        EncryptRequest {
            keyId = keyIdValue
            plaintext = myBytes
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.encrypt(encryptRequest)
        val algorithm: String = response.encryptionAlgorithm.toString()
        println("The encryption algorithm is $algorithm")

        // Return the encrypted data.
        return response.ciphertextBlob
    }
}
```

```
suspend fun decryptData(
    encryptedDataVal: ByteArray?,
    keyIdVal: String?,
) {
    val decryptRequest =
        DecryptRequest {
            ciphertextBlob = encryptedDataVal
            keyId = keyIdVal
        }
    KmsClient { region = "us-west-2" }.use { kmsClient ->
        val decryptResponse = kmsClient.decrypt(decryptRequest)
        val myVal = decryptResponse.plaintext

        // Print the decrypted data.
        print(myVal)
    }
}
```

- 有关 API 详细信息，请参阅《Amazon SDK for Kotlin API Reference》中的 Decrypt。

PHP

适用于 PHP 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /**
     * @param string $keyId
     * @param string $ciphertext
     * @param string $algorithm
     * @return Result
     */
    public function decrypt(string $keyId, string $ciphertext, string $algorithm
 = "SYMMETRIC_DEFAULT")
    {
```

```
        try{
            return $this->client->decrypt([
                'CiphertextBlob' => $ciphertext,
                'EncryptionAlgorithm' => $algorithm,
                'KeyId' => $keyId,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem decrypting the data: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }
```

- 有关 API 详细信息，请参阅《适用于 PHP 的 Amazon SDK API Reference》中的 Decrypt。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class KeyEncrypt:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyEncrypt":
        """
        Creates a KeyEncrypt instance with a default KMS client.

        :return: An instance of KeyEncrypt initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)
```

```
    def decrypt(self, key_id: str, cipher_text: bytes) -> str:
        """
        Decrypts text previously encrypted with a key.

        :param key_id: The ARN or ID of the key used to decrypt the data.
        :param cipher_text: The encrypted text to decrypt.
        :return: The decrypted text.
        """
        try:
            return self.kms_client.decrypt(KeyId=key_id,
CiphertextBlob=cipher_text)[
                "Plaintext"
            ].decode()
        except ClientError as err:
            logger.error(
                "Couldn't decrypt your ciphertext. Here's why: %s",
                err.response["Error"]["Message"],
            )
            raise
```

- 有关 API 详细信息，请参阅《Amazon SDK for Python (Boto3) API Reference》中的 Decrypt。

Ruby

适用于 Ruby 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
require 'aws-sdk-kms' # v2: require 'aws-sdk'

# Decrypted blob
```

```
blob =
 '01020200785d68faeec386af1057904926253051eb2919d3c16078badf65b808b26dd057c101747cadf3593
blob_packed = [blob].pack('H*')

client = Aws::KMS::Client.new(region: 'us-west-2')

resp = client.decrypt({
                        ciphertext_blob: blob_packed
                      })

puts 'Raw text: '
puts resp.plaintext
```

- 有关 API 详细信息，请参阅《适用于 Ruby 的 Amazon SDK API Reference》中的 Decrypt。

Rust

适用于 Rust 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
async fn decrypt_key(client: &Client, key: &str, filename: &str) -> Result<(),
 Error> {
    // Open input text file and get contents as a string
    // input is a base-64 encoded string, so decode it:
    let data = fs::read_to_string(filename)
        .map(|input| {
            base64::decode(input).expect("Input file does not contain valid base
 64 characters.")
        })
        .map(Blob::new);

    let resp = client
        .decrypt()
        .key_id(key)
        .ciphertext_blob(data.unwrap())
        .send()
```

```
        .await?;

    let inner = resp.plaintext.unwrap();
    let bytes = inner.as_ref();

    let s = String::from_utf8(bytes.to_vec()).expect("Could not convert to
 UTF-8");

    println!();
    println!("Decoded string:");
    println!("{}", s);

    Ok(())
}
```

- 有关 API 详细信息，请参阅《Amazon SDK for Rust API Reference》中的 Decrypt。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

`DeleteAlias`与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 `DeleteAlias`。

CLI

Amazon CLI

删除 Amazon KMS 别名

以下 `delete-alias` 示例将删除别名 `alias/example-alias`。别名名称必须以 alias/ 开头。

```
aws kms delete-alias \
    --alias-name alias/example-alias
```

此命令不生成任何输出。要查找别名，请使用 `list-aliases` 命令。

有关更多信息，请参阅《Amazon 密钥管理服务开发人员指南》中的删除别名。

- 有关 API 的详细信息，请参阅Amazon CLI 命令参考DeleteAlias中的。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /**
     * Deletes a specific KMS alias asynchronously.
     *
     * @param aliasName the name of the alias to be deleted
     * @return a {@link CompletableFuture} representing the asynchronous
operation of deleting the specified alias
     */
    public CompletableFuture<Void> deleteSpecificAliasAsync(String aliasName) {
        DeleteAliasRequest deleteAliasRequest = DeleteAliasRequest.builder()
            .aliasName(aliasName)
            .build();

        return getAsyncClient().deleteAlias(deleteAliasRequest)
            .thenRun(() -> {
                logger.info("Alias {} has been deleted successfully", aliasName);
            })
            .exceptionally(throwable -> {
                throw new RuntimeException("Failed to delete alias: " +
aliasName, throwable);
            });
    }
```

- 有关 API 的详细信息，请参阅 Amazon SDK for Java 2.x API 参考DeleteAlias中的。

PHP

适用于 PHP 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /***
     * @param string $aliasName
     * @return void
     */
    public function deleteAlias(string $aliasName)
    {
        try {
            $this->client->deleteAlias([
                'AliasName' => $aliasName,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem deleting the alias: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }
```

- 有关 API 的详细信息，请参阅 适用于 PHP 的 Amazon SDK API 参考DeleteAlias中的。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class AliasManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_key = None

    @classmethod
    def from_client(cls) -> "AliasManager":
        """
        Creates an AliasManager instance with a default KMS client.

        :return: An instance of AliasManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def delete_alias(self, alias: str) -> None:
        """
        Deletes an alias.

        :param alias: The alias to delete.
        """
        try:
            self.kms_client.delete_alias(AliasName=alias)
        except ClientError as err:
            logger.error(
                "Couldn't delete alias %s. Here's why: %s",
                alias,
                err.response["Error"]["Message"],
            )
            raise
```

- 有关 API 的详细信息，请参阅适用DeleteAlias于 Python 的Amazon SDK (Boto3) API 参考。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

`DescribeKey`与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 `DescribeKey`。

操作示例是大型程序的代码摘录，必须在上下文中运行。在以下代码示例中，您可以查看此操作的上下文：

- 了解基本功能

.NET

适用于 .NET 的 Amazon SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
using System;
using System.Threading.Tasks;
using Amazon.KeyManagementService;
using Amazon.KeyManagementService.Model;

/// <summary>
/// Retrieve information about an AWS Key Management Service (AWS KMS) key.
/// You can supply either the key Id or the key Amazon Resource Name (ARN)
/// to the DescribeKeyRequest KeyId property.
/// </summary>
public class DescribeKey
{
    public static async Task Main()
    {
        var keyId = "7c9eccc2-38cb-4c4f-9db3-766ee8dd3ad4";
        var request = new DescribeKeyRequest
        {
            KeyId = keyId,
        };

        var client = new AmazonKeyManagementServiceClient();
```

```
            var response = await client.DescribeKeyAsync(request);
            var metadata = response.KeyMetadata;

            Console.WriteLine($"{metadata.KeyId} created on:
 {metadata.CreationDate}");
            Console.WriteLine($"State: {metadata.KeyState}");
            Console.WriteLine($"{metadata.Description}");
        }
    }
```

- 有关 API 的详细信息，请参阅 适用于 .NET 的 Amazon SDK API 参考DescribeKey中的。

CLI

Amazon CLI

示例 1：查找有关 KMS 密钥的详细信息

以下`describe-key`示例在示例账户和区域中获取有关 Amazon S3 Amazon 托管密钥的详细信息。您可以使用此命令来查找有关 Amazon 托管密钥和客户托管密钥的详细信息。

要指定 KMS 密钥，请使用 `key-id` 参数。此示例使用别名名称值，但您可以在此命令中使用密钥 ID、密钥 ARN、别名名称或别名 ARN。

```
aws kms describe-key \
    --key-id alias/aws/s3
```

输出：

```
{
    "KeyMetadata": {
        "AWSAccountId": "846764612917",
        "KeyId": "b8a9477d-836c-491f-857e-07937918959b",
        "Arn": "arn:aws:kms:us-west-2:846764612917:key/
b8a9477d-836c-491f-857e-07937918959b",
        "CurrentKeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
        "CreationDate": 2017-06-30T21:44:32.140000+00:00,
        "Enabled": true,
```

```
        "Description": "Default KMS key that protects my S3 objects when no other
 key is defined",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "KeyState": "Enabled",
        "Origin": "AWS_KMS",
        "KeyManager": "AWS",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ]
    }
}
```

有关更多信息，请参阅《Amazon 密钥管理服务开发人员指南》中的查看密钥。

示例 2：获取有关 RSA 非对称 KMS 密钥的详细信息

以下 describe-key 示例获取有关用于签名和验证的非对称 RSA KMS 密钥的详细信息。

```
aws kms describe-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

输出：

```
{
    "KeyMetadata": {
        "AWSAccountId": "111122223333",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Arn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "CreationDate": "2019-12-02T19:47:14.861000+00:00",
        "CustomerMasterKeySpec": "RSA_2048",
        "Enabled": false,
        "Description": "",
        "KeyState": "Disabled",
        "Origin": "AWS_KMS",
        "MultiRegion": false,
        "KeyManager": "CUSTOMER",
        "KeySpec": "RSA_2048",
        "KeyUsage": "SIGN_VERIFY",
        "SigningAlgorithms": [
            "RSASSA_PKCS1_V1_5_SHA_256",
```

```
            "RSASSA_PKCS1_V1_5_SHA_384",
            "RSASSA_PKCS1_V1_5_SHA_512",
            "RSASSA_PSS_SHA_256",
            "RSASSA_PSS_SHA_384",
            "RSASSA_PSS_SHA_512"
        ]
    }
}
```

示例 3：获取有关多区域副本密钥的详细信息

以下 describe-key 示例获取多区域副本密钥的元数据。此多区域密钥是对称加密密钥。任何多区域密钥的 describe-key 命令输出都会返回有关主密钥及其所有副本的信息。

```
aws kms describe-key \
    --key-id arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab
```

输出：

```
{
    "KeyMetadata": {
        "MultiRegion": true,
        "AWSAccountId": "111122223333",
        "Arn": "arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
        "CreationDate": "2021-06-28T21:09:16.114000+00:00",
        "CurrentKeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
        "Description": "",
        "Enabled": true,
        "KeyId": "mrk-1234abcd12ab34cd56ef1234567890ab",
        "KeyManager": "CUSTOMER",
        "KeyState": "Enabled",
        "KeyUsage": "ENCRYPT_DECRYPT",
        "Origin": "AWS_KMS",
        "CustomerMasterKeySpec": "SYMMETRIC_DEFAULT",
        "EncryptionAlgorithms": [
            "SYMMETRIC_DEFAULT"
        ],
        "MultiRegionConfiguration": {
            "MultiRegionKeyType": "PRIMARY",
```

```
            "PrimaryKey": {
                "Arn": "arn:aws:kms:us-west-2:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                "Region": "us-west-2"
            },
            "ReplicaKeys": [
                {
                    "Arn": "arn:aws:kms:eu-west-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "eu-west-1"
                },
                {
                    "Arn": "arn:aws:kms:ap-northeast-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "ap-northeast-1"
                },
                {
                    "Arn": "arn:aws:kms:sa-east-1:111122223333:key/
mrk-1234abcd12ab34cd56ef1234567890ab",
                    "Region": "sa-east-1"
                }
            ]
        }
    }
}
```

示例 4：获取有关 HMAC KMS 密钥的详细信息

以下 `describe-key` 示例获取有关 HMAC KMS 密钥的详细信息。

```
aws kms describe-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

输出：

```
{
    "KeyMetadata": {
        "AWSAccountId": "123456789012",
        "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "Arn": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "CreationDate": "2022-04-03T22:23:10.194000+00:00",
        "Enabled": true,
```

```
        "Description": "Test key",
        "KeyUsage": "GENERATE_VERIFY_MAC",
        "KeyState": "Enabled",
        "Origin": "AWS_KMS",
        "KeyManager": "CUSTOMER",
        "CustomerMasterKeySpec": "HMAC_256",
        "MacAlgorithms": [
            "HMAC_SHA_256"
        ],
        "MultiRegion": false
    }
}
```

- 有关 API 的详细信息，请参阅Amazon CLI 命令参考DescribeKey中的。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /**
     * Asynchronously checks if a specified key is enabled.
     *
     * @param keyId the ID of the key to check
     * @return a {@link CompletableFuture} that, when completed, indicates
whether the key is enabled or not
     *
     * @throws RuntimeException if an exception occurs while checking the key
state
     */
    public CompletableFuture<Boolean> isKeyEnabledAsync(String keyId) {
        DescribeKeyRequest keyRequest = DescribeKeyRequest.builder()
            .keyId(keyId)
            .build();

        CompletableFuture<DescribeKeyResponse> responseFuture =
getAsyncClient().describeKey(keyRequest);
```

```
        return responseFuture.whenComplete((resp, ex) -> {
            if (resp != null) {
                KeyState keyState = resp.keyMetadata().keyState();
                if (keyState == KeyState.ENABLED) {
                    logger.info("The key is enabled.");
                } else {
                    logger.info("The key is not enabled. Key state: {}",
 keyState);
                }
            } else {
                throw new RuntimeException(ex);
            }
        }).thenApply(resp -> resp.keyMetadata().keyState() == KeyState.ENABLED);
    }
```

- 有关 API 的详细信息，请参阅 Amazon SDK for Java 2.x API 参考DescribeKey中的。

Kotlin

适用于 Kotlin 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
suspend fun describeSpecifcKey(keyIdVal: String?) {
    val request =
        DescribeKeyRequest {
            keyId = keyIdVal
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.describeKey(request)
        println("The key description is ${response.keyMetadata?.description}")
        println("The key ARN is ${response.keyMetadata?.arn}")
    }
}
```

- 有关 API 的详细信息，请参阅适用DescribeKey于 K otlin 的Amazon SDK API 参考。

PHP

适用于 PHP 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /***
     * @param string $keyId
     * @return array
     */
    public function describeKey(string $keyId)
    {
        try {
            $result = $this->client->describeKey([
                "KeyId" => $keyId,
            ]);
            return $result['KeyMetadata'];
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "NotFoundException"){
                echo "The request was rejected because the specified entity or
resource could not be found.\n";
            }
            throw $caught;
        }
    }
```

- 有关 API 的详细信息，请参阅 适用于 PHP 的 Amazon SDK API 参考DescribeKey中的。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def describe_key(self, key_id: str) -> dict[str, any]:
        """
        Describes a key.

        :param key_id: The ARN or ID of the key to describe.
        :return: Information about the key.
        """

        try:
            key = self.kms_client.describe_key(KeyId=key_id)["KeyMetadata"]
            return key
        except ClientError as err:
            logging.error(
                "Couldn't get key '%s'. Here's why: %s",
                key_id,
```

```
                err.response["Error"]["Message"],
            )
            raise
```

- 有关 API 的详细信息，请参阅适用DescribeKey于 Python 的Amazon SDK (Boto3) API 参考。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

`DisableKey`与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 `DisableKey`。

操作示例是大型程序的代码摘录，必须在上下文中运行。在以下代码示例中，您可以查看此操作的上下文：

- 了解基本功能

.NET

适用于 .NET 的 Amazon SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
using System;
using System.Threading.Tasks;
using Amazon.KeyManagementService;
using Amazon.KeyManagementService.Model;

/// <summary>
/// Disable an AWS Key Management Service (AWS KMS) key and then retrieve
/// the key's status to show that it has been disabled.
/// </summary>
```

```
    public class DisableKey
    {
        public static async Task Main()
        {
            var client = new AmazonKeyManagementServiceClient();

            // The identifier of the AWS KMS key to disable. You can use the
            // key Id or the Amazon Resource Name (ARN) of the AWS KMS key.
            var keyId = "1234abcd-12ab-34cd-56ef-1234567890ab";

            var request = new DisableKeyRequest
            {
                KeyId = keyId,
            };

            var response = await client.DisableKeyAsync(request);

            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                // Retrieve information about the key to show that it has now
                // been disabled.
                var describeResponse = await client.DescribeKeyAsync(new
DescribeKeyRequest
                {
                    KeyId = keyId,
                });
                Console.WriteLine($"{describeResponse.KeyMetadata.KeyId} - state:
{describeResponse.KeyMetadata.KeyState}");
            }
        }
    }
```

- 有关 API 的详细信息，请参阅 适用于 .NET 的 Amazon SDK API 参考DisableKey中的。

CLI

Amazon CLI

暂时禁用 KMS 密钥

以下 disable-key 命令禁用客户自主管理型 KMS 密钥。要重新启用 KMS 密钥，请使用 enable-key 命令。

```
aws kms disable-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

此命令不生成任何输出。

有关更多信息，请参阅《Amazon 密钥管理服务开发人员指南》中的启用和禁用密钥。

- 有关 API 的详细信息，请参阅Amazon CLI 命令参考DisableKey中的。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
   /**
    * Asynchronously disables the specified AWS Key Management Service (KMS)
key.
    *
    * @param keyId the ID or Amazon Resource Name (ARN) of the KMS key to be
disabled
    * @return a CompletableFuture that, when completed, indicates that the key
has been disabled successfully
    */
   public CompletableFuture<Void> disableKeyAsync(String keyId) {
       DisableKeyRequest keyRequest = DisableKeyRequest.builder()
           .keyId(keyId)
           .build();

       return getAsyncClient().disableKey(keyRequest)
           .thenRun(() -> {
               logger.info("Key {} has been disabled successfully",keyId);
           })
           .exceptionally(throwable -> {
```

```
                throw new RuntimeException("Failed to disable key: " + keyId,
 throwable);
            });
    }
```

- 有关 API 的详细信息，请参阅 Amazon SDK for Java 2.x API 参考DisableKey中的。

Kotlin

适用于 Kotlin 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
suspend fun disableKey(keyIdVal: String?) {
    val request =
        DisableKeyRequest {
            keyId = keyIdVal
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        kmsClient.disableKey(request)
        println("$keyIdVal was successfully disabled")
    }
}
```

- 有关 API 的详细信息，请参阅适用DisableKey于 K otlin 的Amazon SDK API 参考。

PHP

适用于 PHP 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /***
     * @param string $keyId
     * @return void
     */
    public function disableKey(string $keyId)
    {
        try {
            $this->client->disableKey([
                'KeyId' => $keyId,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem disabling the key: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }
```

- 有关 API 的详细信息，请参阅 适用于 PHP 的 Amazon SDK API 参考DisableKey中的。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def disable_key(self, key_id: str) -> None:
        try:
            self.kms_client.disable_key(KeyId=key_id)
        except ClientError as err:
            logging.error(
                "Couldn't disable key '%s'. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise
```

- 有关 API 的详细信息，请参阅适用DisableKey于 Python 的Amazon SDK (Boto3) API 参考。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

`EnableKey`与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 `EnableKey`。

操作示例是大型程序的代码摘录，必须在上下文中运行。在以下代码示例中，您可以查看此操作的上下文：

- 了解基本功能

.NET

适用于 .NET 的 Amazon SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    using System;
    using System.Threading.Tasks;
    using Amazon.KeyManagementService;
    using Amazon.KeyManagementService.Model;

    /// <summary>
    /// Enable an AWS Key Management Service (AWS KMS) key.
    /// </summary>
    public class EnableKey
    {
        public static async Task Main()
        {
            var client = new AmazonKeyManagementServiceClient();

            // The identifier of the AWS KMS key to enable. You can use the
            // key Id or the Amazon Resource Name (ARN) of the AWS KMS key.
            var keyId = "1234abcd-12ab-34cd-56ef-1234567890ab";

            var request = new EnableKeyRequest
            {
                KeyId = keyId,
            };

            var response = await client.EnableKeyAsync(request);
            if (response.HttpStatusCode == System.Net.HttpStatusCode.OK)
            {
                // Retrieve information about the key to show that it has now
                // been enabled.
                var describeResponse = await client.DescribeKeyAsync(new
DescribeKeyRequest
```

```
                {
                    KeyId = keyId,
                });
                Console.WriteLine($"{describeResponse.KeyMetadata.KeyId} - state:
 {describeResponse.KeyMetadata.KeyState}");
            }
        }
    }
```

- 有关 API 的详细信息，请参阅 适用于 .NET 的 Amazon SDK API 参考EnableKey中的。

CLI

Amazon CLI

启用 KMS 密钥

以下 enable-key 示例启用客户托管密钥。您可以使用类似的命令来启用您通过 disable-key 命令暂时禁用的 KMS 密钥。您还可以使用它来启用已被禁用的 KMS 密钥，因为已计划删除该密钥且删除已取消。

要指定 KMS 密钥，请使用 key-id 参数。此示例使用密钥 ID 值，但您可以在此命令中使用密钥 ID 或密钥 ARN 值。

在运行此命令之前，将密钥 ID 示例替换为有效 ID。

```
aws kms enable-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

此命令不生成任何输出。要验证 KMS 密钥是否已启用，请使用 describe-key 命令。查看 describe-key 输出中 KeyState 和 Enabled 字段的值。

有关更多信息，请参阅《Amazon 密钥管理服务开发人员指南》中的启用和禁用密钥。

- 有关 API 的详细信息，请参阅Amazon CLI 命令参考EnableKey中的。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /**
     * Asynchronously enables the specified key.
     *
     * @param keyId the ID of the key to enable
     * @return a {@link CompletableFuture} that completes when the key has been
enabled
     */
    public CompletableFuture<Void> enableKeyAsync(String keyId) {
        EnableKeyRequest enableKeyRequest = EnableKeyRequest.builder()
            .keyId(keyId)
            .build();

        CompletableFuture<EnableKeyResponse> responseFuture =
getAsyncClient().enableKey(enableKeyRequest);
        responseFuture.whenComplete((response, exception) -> {
            if (exception == null) {
                logger.info("Key with ID [{}] has been enabled.", keyId);
            } else {
                if (exception instanceof KmsException kmsEx) {
                    throw new RuntimeException("KMS error occurred while enabling
key: " + kmsEx.getMessage(), kmsEx);
                } else {
                    throw new RuntimeException("An unexpected error occurred
while enabling key: " + exception.getMessage(), exception);
                }
            }
        });

        return responseFuture.thenApply(response -> null);
    }
```

- 有关 API 的详细信息，请参阅 Amazon SDK for Java 2.x API 参考EnableKey中的。

Kotlin

适用于 Kotlin 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
suspend fun enableKey(keyIdVal: String?) {
    val request =
        EnableKeyRequest {
            keyId = keyIdVal
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        kmsClient.enableKey(request)
        println("$keyIdVal was successfully enabled.")
    }
}
```

- 有关 API 的详细信息，请参阅适用EnableKey于 K otlin 的Amazon SDK API 参考。

PHP

适用于 PHP 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /***
```

```
     * @param string $keyId
     * @return void
     */
    public function enableKey(string $keyId)
    {
        try {
            $this->client->enableKey([
                'KeyId' => $keyId,
            ]);
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "NotFoundException"){
                echo "The request was rejected because the specified entity or
 resource could not be found.\n";
            }
            throw $caught;
        }
    }
```

- 有关 API 的详细信息，请参阅 适用于 PHP 的 Amazon SDK API 参考EnableKey中的。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.
```

```
        :return: An instance of KeyManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def enable_key(self, key_id: str) -> None:
        """
        Enables a key. Gets the key state after each state change.

        :param key_id: The ARN or ID of the key to enable.
        """
        try:
            self.kms_client.enable_key(KeyId=key_id)
        except ClientError as err:
            logging.error(
                "Couldn't enable key '%s'. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise
```

- 有关 API 的详细信息，请参阅适用EnableKey于 Python 的Amazon SDK (Boto3) API 参考。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

EnableKeyRotation与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 EnableKeyRotation。

CLI

Amazon CLI

启用 KMS 密钥自动轮换

以下 enable-key-rotation 示例启用自动轮换客户托管的 KMS 密钥，轮换周期为 180 天。KMS 密钥将自该命令完成之日起一年（大约 365 天）以及此后每年进行轮换。

`--key-id` 参数标识 KMS 密钥。此示例使用密钥 ARN 值，但您可以使用 KMS 密钥的密钥 ID 或 ARN。`--rotation-period-in-days` 参数指定每次轮换日期之间的天数。可指定 90 到 2560 天之间的值。如果未指定值，则默认值为 365 天。

```
aws kms enable-key-rotation \
    --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --rotation-period-in-days 180
```

此命令不生成任何输出。要验证 KMS 密钥是否已启用，请使用 `get-key-rotation-status` 命令。

有关更多信息，请参阅《Amazon 密钥管理服务开发人员指南》中的轮换密钥。

- 有关 API 的详细信息，请参阅Amazon CLI 命令参考EnableKeyRotation中的。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)
```

```
    def enable_key_rotation(self, key_id: str) -> None:
        """
        Enables rotation for a key.

        :param key_id: The ARN or ID of the key to enable rotation for.
        """
        try:
            self.kms_client.enable_key_rotation(KeyId=key_id)
        except ClientError as err:
            logging.error(
                "Couldn't enable rotation for key '%s'. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise
```

- 有关 API 的详细信息，请参阅适用EnableKeyRotation于 Python 的Amazon SDK (Boto3) API 参考。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

`Encrypt`与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 `Encrypt`。

操作示例是大型程序的代码摘录，必须在上下文中运行。在以下代码示例中，您可以查看此操作的上下文：

- 了解基本功能

CLI

Amazon CLI

示例 1：在 Linux 或 macOS 上对文件内容进行加密

以下`encrypt`命令演示了使用 Amazon CLI 加密数据的推荐方法。

```
aws kms encrypt \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --plaintext fileb://ExamplePlaintextFile \
    --output text \
    --query CiphertextBlob | base64 \
    --decode > ExampleEncryptedFile
```

该命令可以执行以下几项操作：

使用 --plaintext 参数来指示要加密的数据。此参数值必须采用 Base64 编码。该plaintext参数的值必须采用 base64 编码，或者必须使用前缀fileb://，它指示 CL Amazon I 从文件中读取二进制数据。如果文件不在当前目录中，请键入文件的完整路径。例如：fileb:///var/tmp/ExamplePlaintextFile 或 fileb://C:\Temp\ExamplePlaintextFile。有关从文件读取 Amazon CLI 参数值的更多信息，请参阅《Amazon 命令行界面用户指南》中的 “从文件加载参数” 和 “ Amazon 命令行工具博客” 上的 “本地文件参数最佳实践”。使用--output和--query参数控制命令的输出。这些参数从命令的输出中提取被称为密文的加密数据。有关控制输出的更多信息，请参阅控制命令Amazon 命令行界面用户指南中的输出。使用该base64实用程序将提取的输出解码为二进制数据。成功命令返回的密文是 base64 编码的文本。encrypt必须先解码此文本，然后才能使用 Amazon CLI 对其进行解密。将二进制密文保存到文件中。命令 (> ExampleEncryptedFile) 的最后一部分将二进制密文保存到文件中，以便于解密。有关使用 Amazon CLI 解密数据的命令示例，请参阅解密示例。

示例 2：使用 Amazon CLI 在 Windows 上加密数据

此示例与前一个示例相同，不同之处在于，它使用 certutil 工具代替 base64。此过程需要两个命令，如以下示例所示。

```
aws kms encrypt \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --plaintext fileb://ExamplePlaintextFile \
    --output text \
    --query CiphertextBlob > C:\Temp\ExampleEncryptedFile.base64

certutil -decode C:\Temp\ExampleEncryptedFile.base64 C:\Temp\ExampleEncryptedFile
```

示例 3：使用非对称 KMS 密钥进行加密

以下 encrypt 命令显示如何使用非对称 KMS 密钥加密明文。--encryption-algorithm 参数是必需的。与所有 encrypt CLI 命令一样，该plaintext参数必须采用 base64 编码，或者必须使用fileb://前缀，它告诉 CL Amazon I 从文件中读取二进制数据。

```
aws kms encrypt \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --encryption-algorithm RSAES_OAEP_SHA_256 \
    --plaintext fileb://ExamplePlaintextFile \
    --output text \
    --query CiphertextBlob | base64 \
    --decode > ExampleEncryptedFile
```

此命令不生成任何输出。

- 有关 API 详细信息，请参阅《Amazon CLI 命令参考》中的 Encrypt。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
   /**
    * Encrypts the given text asynchronously using the specified KMS client and
key ID.
    *
    * @param keyId the ID of the KMS key to use for encryption
    * @param text the text to encrypt
    * @return a CompletableFuture that completes with the encrypted data as an
SdkBytes object
    */
   public CompletableFuture<SdkBytes> encryptDataAsync(String keyId, String
text) {
       SdkBytes myBytes = SdkBytes.fromUtf8String(text);
       EncryptRequest encryptRequest = EncryptRequest.builder()
           .keyId(keyId)
           .plaintext(myBytes)
```

```
            .build();

        CompletableFuture<EncryptResponse> responseFuture =
 getAsyncClient().encrypt(encryptRequest).toCompletableFuture();
        return responseFuture.whenComplete((response, ex) -> {
            if (response != null) {
                String algorithm = response.encryptionAlgorithm().toString();
                logger.info("The string was encrypted with algorithm {}.",
 algorithm);
            } else {
                throw new RuntimeException(ex);
            }
        }).thenApply(EncryptResponse::ciphertextBlob);
    }
```

- 有关 API 详细信息，请参阅《Amazon SDK for Java 2.x API Reference》中的 Encrypt。

Kotlin

适用于 Kotlin 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
suspend fun encryptData(keyIdValue: String): ByteArray? {
    val text = "This is the text to encrypt by using the AWS KMS Service"
    val myBytes: ByteArray = text.toByteArray()

    val encryptRequest =
        EncryptRequest {
            keyId = keyIdValue
            plaintext = myBytes
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.encrypt(encryptRequest)
        val algorithm: String = response.encryptionAlgorithm.toString()
        println("The encryption algorithm is $algorithm")
```

```
        // Return the encrypted data.
        return response.ciphertextBlob
    }
}

suspend fun decryptData(
    encryptedDataVal: ByteArray?,
    keyIdVal: String?,
) {
    val decryptRequest =
        DecryptRequest {
            ciphertextBlob = encryptedDataVal
            keyId = keyIdVal
        }
    KmsClient { region = "us-west-2" }.use { kmsClient ->
        val decryptResponse = kmsClient.decrypt(decryptRequest)
        val myVal = decryptResponse.plaintext

        // Print the decrypted data.
        print(myVal)
    }
}
```

- 有关 API 详细信息，请参阅《Amazon SDK for Kotlin API Reference》中的 Encrypt。

PHP

适用于 PHP 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /**
     * @param string $keyId
     * @param string $text
     * @return Result
```

```
     */
    public function encrypt(string $keyId, string $text)
    {
        try {
            return $this->client->encrypt([
                'KeyId' => $keyId,
                'Plaintext' => $text,
            ]);
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "DisabledException"){
                echo "The request was rejected because the specified KMS key is
 not enabled.\n";
            }
            throw $caught;
        }
    }
```

- 有关 API 详细信息，请参阅《适用于 PHP 的 Amazon SDK API Reference》中的 Encrypt。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class KeyEncrypt:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyEncrypt":
        """
        Creates a KeyEncrypt instance with a default KMS client.

        :return: An instance of KeyEncrypt initialized with the default KMS
 client.
```

```
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def encrypt(self, key_id: str, text: str) -> bytes:
        """
        Encrypts text by using the specified key.

        :param key_id: The ARN or ID of the key to use for encryption.
        :param text: The text to encrypt.
        :return: The encrypted version of the text.
        """
        try:
            response = self.kms_client.encrypt(KeyId=key_id,
Plaintext=text.encode())
            print(
                f"The string was encrypted with algorithm
{response['EncryptionAlgorithm']}"
            )
            return response["CiphertextBlob"]
        except ClientError as err:
            if err.response["Error"]["Code"] == "DisabledException":
                logger.error(
                    "Could not encrypt because the key %s is disabled.", key_id
                )
            else:
                logger.error(
                    "Couldn't encrypt text. Here's why: %s",
                    err.response["Error"]["Message"],
                )
            raise
```

- 有关 API 详细信息，请参阅《Amazon SDK for Python (Boto3) API Reference》中的 Encrypt。

Ruby

适用于 Ruby 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
require 'aws-sdk-kms' # v2: require 'aws-sdk'

# ARN of the AWS KMS key.
#
# Replace the fictitious key ARN with a valid key ID

keyId = 'arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab'

text = '1234567890'

client = Aws::KMS::Client.new(region: 'us-west-2')

resp = client.encrypt({
                        key_id: keyId,
                        plaintext: text
                      })

# Display a readable version of the resulting encrypted blob.
puts 'Blob:'
puts resp.ciphertext_blob.unpack('H*')
```

- 有关 API 详细信息，请参阅《适用于 Ruby 的 Amazon SDK API Reference》中的 Encrypt。

Rust

适用于 Rust 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
async fn encrypt_string(
    verbose: bool,
    client: &Client,
    text: &str,
    key: &str,
    out_file: &str,
) -> Result<(), Error> {
    let blob = Blob::new(text.as_bytes());

    let resp = client.encrypt().key_id(key).plaintext(blob).send().await?;

    // Did we get an encrypted blob?
    let blob = resp.ciphertext_blob.expect("Could not get encrypted text");
    let bytes = blob.as_ref();

    let s = base64::encode(bytes);

    let mut ofile = File::create(out_file).expect("unable to create file");
    ofile.write_all(s.as_bytes()).expect("unable to write");

    if verbose {
        println!("Wrote the following to {:?}", out_file);
        println!("{}", s);
    }

    Ok(())
}
```

- 有关 API 详细信息，请参阅《Amazon SDK for Rust API Reference》中的 Encrypt。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

`GenerateDataKey`与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 `GenerateDataKey`。

CLI

Amazon CLI

示例 1：生成 256 位对称数据密钥

以下`generate-data-key`示例请求一个 256 位的对称数据密钥以供外部使用。 Amazon该命令返回一个明文数据密钥以供立即使用和删除，以及以指定 KMS 密钥加密的该数据密钥的副本。您可以安全地将加密的数据密钥与加密的数据一起存储。

要请求 256 位数据密钥，请使用值为 `AES_256` 的 `key-spec` 参数。要请求 128 位数据密钥，请使用值为 `AES_128` 的 `key-spec` 参数。对于所有其他数据密钥长度，请使用 `number-of-bytes` 参数。

您指定的 KMS 密钥必须是对称加密 KMS 密钥，即 KeySpec 值为 SYMMETRIC_DEFAULT 的 KMS 密钥。

```
aws kms generate-data-key \
    --key-id alias/ExampleAlias \
    --key-spec AES_256
```

输出：

```
{
    "Plaintext": "VdzKNHGzUAzJeRBVY+uUmofUGGiDzyB3+i9fVkh3piw=",
    "KeyId": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6",
    "CiphertextBlob":
 "AQEDAHjRYf5WytIc0C857tFSnBaPn2F8DgfmThbJlGfR8P3WlwAAAH4wfAYJKoZIhvcNAQcGoG8wbQIBADBoBgk
+YdhV8MrkBQPeac0ReRVNDt9qleAt+SHgIRF8P0H+7U="
}
```

Plaintext（明文数据密钥）和 CiphertextBlob（加密数据密钥）均以 base64 编码的格式返回。

有关更多信息，请参阅《Amazon 密钥管理服务开发人员指南》中的数据密钥。示例 2：生成 512 位对称数据密钥

以下 generate-data-key 示例请求用于加密和解密的 512 位对称数据密钥。该命令返回一个明文数据密钥以供立即使用和删除，以及以指定 KMS 密钥加密的该数据密钥的副本。您可以安全地将加密的数据密钥与加密的数据一起存储。

要请求 128 或 256 位以外的密钥长度，请使用 number-of-bytes 参数。为了请求 512 位数据密钥，以下示例使用值为 64（字节）的 number-of-bytes 参数。

您指定的 KMS 密钥必须是对称加密 KMS 密钥，即密钥规格值为 SYMMETRIC_DEFAULT 的 KMS 密钥。

注意：本示例输出中的值被截断，便于显示。

```
aws kms generate-data-key \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --number-of-bytes 64
```

输出：

```
{
    "CiphertextBlob": "AQIBAHi6LtupRpdKl2aJTzkK6FbhOtQkMlQJJH3PdtHvS/y+hAEnX/
QQNmMwDfg2korNMEc8AAACaDCCAmQGCSqGSIb3DQEHBqCCAlUwggJRAgEAMIICSgYJKoZ...",
    "Plaintext": "ty8Lr0Bk6OF07M2BWt6qbFdNB
+G00ZLtf5MSEb4al3R2UKWGOp06njAwy2n72VRm2m7z/
Pm9Wpbvttz6a4lSo9hgPvKhZ5y6RTm4OovEXiVfBveyX3DQxDzRSwbKDPk/...",
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6"
}
```

Plaintext（明文数据密钥）和 CiphertextBlob（加密数据密钥）均以 base64 编码的格式返回。

有关更多信息，请参阅《Amazon 密钥管理服务开发人员指南》中的数据密钥。

- 有关 API 的详细信息，请参阅Amazon CLI 命令参考GenerateDataKey中的。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def generate_data_key(self, key_id):
        """
        Generates a symmetric data key that can be used for client-side
 encryption.
        """
        answer = input(
            f"Do you want to generate a symmetric data key from key {key_id} (y/
n)? "
        )
        if answer.lower() == "y":
            try:
                data_key = self.kms_client.generate_data_key(
                    KeyId=key_id, KeySpec="AES_256"
```

```
            )
        except ClientError as err:
            logger.error(
                "Couldn't generate a data key for key %s. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
        else:
            pprint(data_key)
```

- 有关 API 的详细信息，请参阅适用GenerateDataKey于 Python 的Amazon SDK (Boto3) API 参考。

Rust

适用于 Rust 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
async fn make_key(client: &Client, key: &str) -> Result<(), Error> {
    let resp = client
        .generate_data_key()
        .key_id(key)
        .key_spec(DataKeySpec::Aes256)
        .send()
        .await?;

    // Did we get an encrypted blob?
    let blob = resp.ciphertext_blob.expect("Could not get encrypted text");
    let bytes = blob.as_ref();

    let s = base64::encode(bytes);

    println!();
    println!("Data key:");
```

```
    println!("{}", s);

    Ok(())
}
```

- 有关 API 的详细信息，请参阅适用GenerateDataKey于 Rust 的Amazon SDK API 参考。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

`GenerateDataKeyWithoutPlaintext`与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 `GenerateDataKeyWithoutPlaintext`。

CLI

Amazon CLI

生成不带明文密钥的 256 位对称数据密钥

以下 `generate-data-key-without-plaintext` 示例请求 256 位对称数据密钥的加密副本以供在 Amazon外部使用。准备好使用数据密钥时，可以调用 Amazon KMS 对其进行解密。

要请求 256 位数据密钥，请使用值为 `AES_256` 的 `key-spec` 参数。要请求 128 位数据密钥，请使用值为 `AES_128` 的 `key-spec` 参数。对于所有其他数据密钥长度，请使用 `number-of-bytes` 参数。

您指定的 KMS 密钥必须是对称加密 KMS 密钥，即 KeySpec 值为 SYMMETRIC_DEFAULT 的 KMS 密钥。

```
aws kms generate-data-key-without-plaintext \
    --key-id "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab" \
    --key-spec AES_256
```

输出：

```
{
    "CiphertextBlob":
 "AQEDAHjRYf5WytIc0C857tFSnBaPn2F8DgfmThbJlGfR8P3WlwAAAH4wfAYJKoZIhvcNAQcGoG8wbQIBADBoBgk
```

```
    "KeyId": "arn:aws:kms:us-
east-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "KeyMaterialId":
 "0b7fd7ddbac6eef27907413567cad8c810e2883dc8a7534067a82ee1142fc1e6"
}
```

`CiphertextBlob`（加密数据密钥）以 base64 编码的格式返回。

有关更多信息，请参阅《Amazon 密钥管理服务开发人员指南》中的数据密钥。

- 有关 API 的详细信息，请参阅Amazon CLI 命令参考GenerateDataKeyWithoutPlaintext中的。

Rust

适用于 Rust 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
async fn make_key(client: &Client, key: &str) -> Result<(), Error> {
    let resp = client
        .generate_data_key_without_plaintext()
        .key_id(key)
        .key_spec(DataKeySpec::Aes256)
        .send()
        .await?;

    // Did we get an encrypted blob?
    let blob = resp.ciphertext_blob.expect("Could not get encrypted text");
    let bytes = blob.as_ref();

    let s = base64::encode(bytes);

    println!();
    println!("Data key:");
    println!("{}", s);

    Ok(())
```

```
}
```

- 有关 API 的详细信息，请参阅适用GenerateDataKeyWithoutPlaintext于 Rust 的Amazon SDK API 参考。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

GenerateRandom与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 GenerateRandom。

CLI

Amazon CLI

示例 1：生成 256 位随机字节字符串（Linux 或 macOS）

以下 generate-random 示例生成 256 位（32 字节）、以 base64 编码的随机字节字符串。该示例对字节字符串进行解码并将其保存在随机文件中。

运行此命令时，您必须使用 number-of-bytes 参数指定随机值的长度（以字节为单位）。

在运行此命令时，无需指定 KMS 密钥。随机字节字符串与任何 KMS 密钥无关。

默认情况下， Amazon KMS 会生成随机数。但是，如果您指定自定义密钥存储库，则随机字节字符串将在与该自定义密钥库关联的 Amazon CloudHSM 集群中生成。

本示例使用以下参数和值：

它使用值为的必需--number-of-bytes参数32来请求一个 32 字节（256 位）的字符串。它使用值为的--output参数指示 CL Amazon I 将输出作为文本返回，而不是 JSON。它使用从响应中提取Plaintext属性的值。它使用重定--query parameter向运算符 (>) 从响应中提取的字符串中提取出来。它使用重定向运算符 (>) 来保存解码后的字节字符串到文件。它使用重定向运算符 (>) text base64 ExampleRandom将二进制密文保存到文件中。

```
aws kms generate-random \
    --number-of-bytes 32 \
    --output text \
```

```
    --query Plaintext | base64 --decode > ExampleRandom
```

此命令不生成任何输出。

有关更多信息，请参阅《Amazon 密钥管理服务 API 参考》GenerateRandom中的。

示例 2：生成 256 位随机数（Windows 命令提示符）

以下示例使用 generate-random 命令生成 256 位（32 字节）、以 base64 编码的随机字节字符串。该示例对字节字符串进行解码并将其保存在随机文件中。此示例与前面的示例相同，不同之处在于：它在将随机字节字符串保存到文件之前，使用 Windows 中的 certutil 实用工具对随机字节字符串进行 base64 解码。

首先，生成一个 base64 编码的随机字节字符串并将其保存在临时文件 ExampleRandom.base64 中。

```
aws kms generate-random \
    --number-of-bytes 32 \
    --output text \
    --query Plaintext > ExampleRandom.base64
```

由于 generate-random 命令的输出保存在文件中，因此，此示例不生成任何输出。

现在，使用 certutil -decode 命令解码 ExampleRandom.base64 文件中以 base64 编码的字节字符串。然后，它将解码后的字节字符串保存在 ExampleRandom 文件中。

```
certutil -decode ExampleRandom.base64 ExampleRandom
```

输出：

```
Input Length = 18
Output Length = 12
CertUtil: -decode command completed successfully.
```

有关更多信息，请参阅《Amazon 密钥管理服务 API 参考》GenerateRandom中的。

- 有关 API 的详细信息，请参阅Amazon CLI 命令参考GenerateRandom中的。

Rust

适用于 Rust 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
async fn make_string(client: &Client, length: i32) -> Result<(), Error> {
    let resp = client
        .generate_random()
        .number_of_bytes(length)
        .send()
        .await?;

    // Did we get an encrypted blob?
    let blob = resp.plaintext.expect("Could not get encrypted text");
    let bytes = blob.as_ref();

    let s = base64::encode(bytes);

    println!();
    println!("Data key:");
    println!("{}", s);

    Ok(())
}
```

- 有关 API 的详细信息，请参阅适用GenerateRandom于 Rust 的Amazon SDK API 参考。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

`GetKeyPolicy`与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 `GetKeyPolicy`。

操作示例是大型程序的代码摘录，必须在上下文中运行。在以下代码示例中，您可以查看此操作的上下文：

- 了解基本功能

CLI

Amazon CLI

将密钥策略从一个 KMS 密钥复制到另一个 KMS 密钥

以下 `get-key-policy` 示例从一个 KMS 密钥获取密钥策略并将其保存在文本文件中。然后，它使用文本文件作为策略输入替换其他 KMS 密钥的策略。

由于 `put-key-policy` 的 `--policy` 参数需要字符串，因此，您必须使用 `--output text` 选项将输出作为文本字符串（而不是 JSON）返回。

```
aws kms get-key-policy \
    --policy-name default \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --query Policy \
    --output text > policy.txt

aws kms put-key-policy \
    --policy-name default \
    --key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \
    --policy file://policy.txt
```

此命令不生成任何输出。

有关更多信息，请参阅 Amazon KMS API 参考PutKeyPolicy中的。

- 有关 API 的详细信息，请参阅Amazon CLI 命令参考GetKeyPolicy中的。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class KeyPolicy:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyPolicy":
        """
        Creates a KeyPolicy instance with a default KMS client.

        :return: An instance of KeyPolicy initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def get_policy(self, key_id: str) -> dict[str, str]:
        """
        Gets the policy of a key.

        :param key_id: The ARN or ID of the key to query.
        :return: The key policy as a dict.
        """
        if key_id != "":
            try:
                response = self.kms_client.get_key_policy(
                    KeyId=key_id,
                )
                policy = json.loads(response["Policy"])
            except ClientError as err:
                logger.error(
                    "Couldn't get policy for key %s. Here's why: %s",
```

```
                    key_id,
                    err.response["Error"]["Message"],
                )
                raise
            else:
                pprint(policy)
                return policy
        else:
            print("Skipping get policy demo.")
```

- 有关 API 的详细信息，请参阅适用GetKeyPolicy于 Python 的Amazon SDK (Boto3) API 参考。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

`ListAliases`与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 `ListAliases`。

操作示例是大型程序的代码摘录，必须在上下文中运行。在以下代码示例中，您可以查看此操作的上下文：

- 了解基本功能

.NET

适用于 .NET 的 Amazon SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
using System;
using System.Threading.Tasks;
using Amazon.KeyManagementService;
```

```
    using Amazon.KeyManagementService.Model;

    /// <summary>
    /// List the AWS Key Management Service (AWS KMS) aliases that have been
defined for
    /// the keys in the same AWS Region as the default user. If you want to list
    /// the aliases in a different Region, pass the Region to the client
    /// constructor.
    /// </summary>
    public class ListAliases
    {
        public static async Task Main()
        {
            var client = new AmazonKeyManagementServiceClient();
            var request = new ListAliasesRequest();
            var response = new ListAliasesResponse();

            do
            {
                response = await client.ListAliasesAsync(request);

                response.Aliases.ForEach(alias =>
                {
                    Console.WriteLine($"Created: {alias.CreationDate} Last
Update: {alias.LastUpdatedDate} Name: {alias.AliasName}");
                });

                request.Marker = response.NextMarker;
            }
            while (response.Truncated);
        }
    }
```

- 有关 API 的详细信息，请参阅 适用于 .NET 的 Amazon SDK API 参考ListAliases中的。

CLI

Amazon CLI

示例 1：列出 Amazon 账户和区域中的所有别名

以下示例使用list-aliases命令列出 Amazon 账户默认区域中的所有别名。输出包括与 Amazon 托管 KMS 密钥和客户托管的 KMS 密钥关联的别名。

```
aws kms list-aliases
```

输出：

```
{
    "Aliases": [
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/testKey",
            "AliasName": "alias/testKey",
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
        },
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/FinanceDept",
            "AliasName": "alias/FinanceDept",
            "TargetKeyId": "0987dcba-09fe-87dc-65ba-ab0987654321"
        },
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/dynamodb",
            "AliasName": "alias/aws/dynamodb",
            "TargetKeyId": "1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d"
        },
        {
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/aws/ebs",
            "AliasName": "alias/aws/ebs",
            "TargetKeyId": "0987ab65-43cd-21ef-09ab-87654321cdef"
        },
        ...
    ]
}
```

示例 2：列出特定 KMS 密钥的所有别名

以下示例使用 list-aliases 命令及其 key-id 参数列出与特定 KMS 密钥关联的所有别名。

每个别名都仅与一个 KMS 密钥关联，但一个 KMS 密钥可以有多个别名。此命令非常有用，因为 Amazon KMS 控制台仅列出每个 KMS 密钥的一个别名。要查找 KMS 密钥的所有别名，您必须使用 list-aliases 命令。

此示例使用 KMS 密钥的密钥 ID 作为 `--key-id` 参数，但您可以在此命令中使用密钥 ID、密钥 ARN、别名名称或别名 ARN。

```
aws kms list-aliases --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

输出：

```
{
    "Aliases": [
        {
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/oregon-test-
key",
            "AliasName": "alias/oregon-test-key"
        },
        {
            "TargetKeyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
            "AliasArn": "arn:aws:kms:us-west-2:111122223333:alias/project121-
test",
            "AliasName": "alias/project121-test"
        }
    ]
}
```

有关更多信息，请参阅《Amazon 密钥管理服务开发人员指南》中的使用别名。

- 有关 API 的详细信息，请参阅Amazon CLI 命令参考ListAliases中的。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /**
     * Asynchronously lists all the aliases in the current AWS account.
     *
```

```
     * @return a {@link CompletableFuture} that completes when the list of
aliases has been processed
     */
    public CompletableFuture<Object> listAllAliasesAsync() {
        ListAliasesRequest aliasesRequest = ListAliasesRequest.builder()
            .limit(15)
            .build();

        ListAliasesPublisher paginator =
getAsyncClient().listAliasesPaginator(aliasesRequest);
        return paginator.subscribe(response -> {
                response.aliases().forEach(alias ->
                    logger.info("The alias name is: " + alias.aliasName())
                );
            })
            .thenApply(v -> null)
            .exceptionally(ex -> {
                if (ex.getCause() instanceof KmsException) {
                    KmsException e = (KmsException) ex.getCause();
                    throw new RuntimeException("A KMS exception occurred: " +
e.getMessage());
                } else {
                    throw new RuntimeException("An unexpected error occurred: " +
ex.getMessage());
                }
            });
    }
```

- 有关 API 的详细信息，请参阅 Amazon SDK for Java 2.x API 参考ListAliases中的。

Kotlin

适用于 Kotlin 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
suspend fun listAllAliases() {
```

```
    val request =
        ListAliasesRequest {
            limit = 15
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.listAliases(request)
        response.aliases?.forEach { alias ->
            println("The alias name is ${alias.aliasName}")
        }
    }
}
```

- 有关 API 的详细信息，请参阅适用ListAliases于 K otlin 的Amazon SDK API 参考。

PHP

适用于 PHP 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /***
     * @param string $keyId
     * @param int $limit
     * @return ResultPaginator
     */
    public function listAliases(string $keyId = "", int $limit = 0)
    {
        $args = [];
        if($keyId){
            $args['KeyId'] = $keyId;
        }
        if($limit){
            $args['Limit'] = $limit;
        }
        try{
```

```
            return $this->client->getPaginator("ListAliases", $args);
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "InvalidMarkerException"){
                echo "The request was rejected because the marker that specifies
 where pagination should next begin is not valid.\n";
            }
            throw $caught;
        }
    }
```

- 有关 API 的详细信息，请参阅 适用于 PHP 的 Amazon SDK API 参考ListAliases中的。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class AliasManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_key = None

    @classmethod
    def from_client(cls) -> "AliasManager":
        """
        Creates an AliasManager instance with a default KMS client.

        :return: An instance of AliasManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def list_aliases(self, page_size: int) -> None:
```

```
        """
        Lists aliases for the current account.
        :param page_size: The number of aliases to list per page.
        """
        try:
            alias_paginator = self.kms_client.get_paginator("list_aliases")
            for alias_page in alias_paginator.paginate(
                PaginationConfig={"PageSize": page_size}
            ):
                print(f"Here are {page_size} aliases:")
                pprint(alias_page["Aliases"])
                if alias_page["Truncated"]:
                    answer = input(
                        f"Do you want to see the next {page_size} aliases (y/n)?
"
                    )
                    if answer.lower() != "y":
                        break
                else:
                    print("That's all your aliases!")
        except ClientError as err:
            logging.error(
                "Couldn't list your aliases. Here's why: %s",
                err.response["Error"]["Message"],
            )
            raise
```

- 有关 API 的详细信息，请参阅适用ListAliases于 Python 的Amazon SDK (Boto3) API 参考。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

`ListGrants`与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 `ListGrants`。

操作示例是大型程序的代码摘录，必须在上下文中运行。在以下代码示例中，您可以查看此操作的上下文：

- 了解基本功能

.NET

适用于 .NET 的 Amazon SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    using System;
    using System.Threading.Tasks;
    using Amazon.KeyManagementService;
    using Amazon.KeyManagementService.Model;

    /// <summary>
    /// List the AWS Key Management Service (AWS KMS) grants that are associated
with
    /// a specific key.
    /// </summary>
    public class ListGrants
    {
        public static async Task Main()
        {
            // The identifier of the AWS KMS key to disable. You can use the
            // key Id or the Amazon Resource Name (ARN) of the AWS KMS key.
            var keyId = "1234abcd-12ab-34cd-56ef-1234567890ab";
            var client = new AmazonKeyManagementServiceClient();
            var request = new ListGrantsRequest
            {
                KeyId = keyId,
            };

            var response = new ListGrantsResponse();

            do
            {
                response = await client.ListGrantsAsync(request);

                response.Grants.ForEach(grant =>
                {
                    Console.WriteLine($"{grant.GrantId}");
```

```
                });

                request.Marker = response.NextMarker;
            }
            while (response.Truncated);
        }
    }
```

- 有关 API 的详细信息，请参阅 适用于 .NET 的 Amazon SDK API 参考ListGrants中的。

CLI

Amazon CLI

查看 Amazon KMS 密钥的授权

以下`list-grants`示例显示了您的账户中针对亚马逊 DynamoDB 的指定 Amazon 托管 KMS 密钥的所有授权。该授权允许 DynamoDB 在将 DynamoDB 表写入磁盘之前代表您使用 KMS 密钥对其进行加密。您可以使用这样的命令来查看 Amazon 账户和区域中 Amazon 托管 KMS 密钥和客户托管 KMS 密钥的授权。

此命令使用带有密钥 ID 的 `key-id` 参数来识别 KMS 密钥。您可以使用密钥 ID 或密钥 ARN 来识别 KMS 密钥。要获取 Amazon 托管 KMS 密钥的密钥 ID 或密钥 ARN，请使用`list-keys`或`list-aliases`命令。

```
aws kms list-grants \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

输出显示，该授权向 Amazon DynamoDB 授予了使用 KMS 密钥进行加密操作的权限，并向其授予了查看有关 KMS 密钥的详细信息（`DescribeKey`）和停用授权（`RetireGrant`）的权限。`EncryptionContextSubset` 约束将这些权限限制为包含指定加密上下文对的请求。因此，授权中的权限仅对指定账户和 DynamoDB 表有效。

```
{
    "Grants": [
        {
            "Constraints": {
                "EncryptionContextSubset": {
                    "aws:dynamodb:subscriberId": "123456789012",
```

```
                    "aws:dynamodb:tableName": "Services"
                }
            },
            "IssuingAccount": "arn:aws:iam::123456789012:root",
            "Name": "8276b9a6-6cf0-46f1-b2f0-7993a7f8c89a",
            "Operations": [
                "Decrypt",
                "Encrypt",
                "GenerateDataKey",
                "ReEncryptFrom",
                "ReEncryptTo",
                "RetireGrant",
                "DescribeKey"
            ],
            "GrantId":
 "1667b97d27cf748cf05b487217dd4179526c949d14fb3903858e25193253fe59",
            "KeyId": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "RetiringPrincipal": "dynamodb.us-west-2.amazonaws.com",
            "GranteePrincipal": "dynamodb.us-west-2.amazonaws.com",
            "CreationDate": "2021-05-13T18:32:45.144000+00:00"
        }
    ]
}
```

有关更多信息，请参阅《密Amazon 钥管理服务开发人员指南》中的 Amazon KMS 中的授权。

- 有关 API 的详细信息，请参阅Amazon CLI 命令参考ListGrants中的。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /**
     * Asynchronously displays the grant IDs for the specified key ID.
     *
```

```
    * @param keyId the ID of the AWS KMS key for which to list the grants
    * @return a {@link CompletableFuture} that, when completed, will be null
if the operation succeeded, or will throw a {@link RuntimeException} if the
operation failed
    * @throws RuntimeException if there was an error listing the grants, either
due to an {@link KmsException} or an unexpected error
    */
   public CompletableFuture<Object> displayGrantIdsAsync(String keyId) {
       ListGrantsRequest grantsRequest = ListGrantsRequest.builder()
           .keyId(keyId)
           .limit(15)
           .build();

       ListGrantsPublisher paginator =
getAsyncClient().listGrantsPaginator(grantsRequest);
       return paginator.subscribe(response -> {
               response.grants().forEach(grant -> {
                   logger.info("The grant Id is: " + grant.grantId());
               });
           })
           .thenApply(v -> null)
           .exceptionally(ex -> {
               Throwable cause = ex.getCause();
               if (cause instanceof KmsException) {
                   throw new RuntimeException("Failed to list grants: " +
cause.getMessage(), cause);
               } else {
                   throw new RuntimeException("An unexpected error occurred: " +
cause.getMessage(), cause);
               }
           });
   }
```

- 有关 API 的详细信息，请参阅 Amazon SDK for Java 2.x API 参考ListGrants中的。

Kotlin

适用于 Kotlin 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
suspend fun displayGrantIds(keyIdVal: String?) {
    val request =
        ListGrantsRequest {
            keyId = keyIdVal
            limit = 15
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.listGrants(request)
        response.grants?.forEach { grant ->
            println("The grant Id is ${grant.grantId}")
        }
    }
}
```

- 有关 API 的详细信息，请参阅适用ListGrants于 K otlin 的Amazon SDK API 参考。

PHP

适用于 PHP 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /***
```

```
     * @param string $keyId
     * @return Result
     */
    public function listGrants(string $keyId)
    {
        try{
            return $this->client->listGrants([
                'KeyId' => $keyId,
            ]);
        }catch(KmsException $caught){
            if($caught->getAwsErrorMessage() == "NotFoundException"){
                echo "    The request was rejected because the specified entity
 or resource could not be found.\n";
            }
            throw $caught;
        }
    }
```

- 有关 API 的详细信息，请参阅 适用于 PHP 的 Amazon SDK API 参考ListGrants中的。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class GrantManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "GrantManager":
        """
        Creates a GrantManager instance with a default KMS client.
```

```
        :return: An instance of GrantManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def list_grants(self, key_id):
        """
        Lists grants for a key.

        :param key_id: The ARN or ID of the key to query.
        :return: The grants for the key.
        """
        try:
            paginator = self.kms_client.get_paginator("list_grants")
            grants = []
            page_iterator = paginator.paginate(KeyId=key_id)
            for page in page_iterator:
                grants.extend(page["Grants"])

            print(f"Grants for key {key_id}:")
            pprint(grants)
            return grants
        except ClientError as err:
            logger.error(
                "Couldn't list grants for key %s. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise
```

- 有关 API 的详细信息，请参阅适用ListGrants于 Python 的Amazon SDK (Boto3) API 参考。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

`ListKeyPolicies`与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 `ListKeyPolicies`。

CLI

Amazon CLI

获取 KMS 密钥的密钥策略名称

以下 `list-key-policies` 示例获取示例账户和区域中客户托管密钥的密钥策略名称。您可以使用此命令查找托管密钥和客户 Amazon 托管密钥的密钥策略名称。

由于唯一有效的密钥策略名称是 `default`，因此，此命令没有用。

要指定 KMS 密钥，请使用 `key-id` 参数。此示例使用密钥 ID 值，但您可以在此命令中使用密钥 ID 或密钥 ARN。

```
aws kms list-key-policies \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

输出：

```
{
    "PolicyNames": [
    "default"
    ]
}
```

有关 Amazon KMS 密钥策略的更多信息，请参阅《密钥管理服务开发人员指南》中的在 Amazon KMS 中使用Amazon 密钥策略。

- 有关 API 的详细信息，请参阅Amazon CLI 命令参考ListKeyPolicies中的。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /**
```

```
    * Asynchronously retrieves the key policy for the specified key ID and
policy name.
    *
    * @param keyId       the ID of the AWS KMS key for which to retrieve the
policy
    * @param policyName the name of the key policy to retrieve
    * @return a {@link CompletableFuture} that, when completed, contains the key
policy as a {@link String}
    */
   public CompletableFuture<String> getKeyPolicyAsync(String keyId, String
policyName) {
       GetKeyPolicyRequest policyRequest = GetKeyPolicyRequest.builder()
           .keyId(keyId)
           .policyName(policyName)
           .build();

       return getAsyncClient().getKeyPolicy(policyRequest)
           .thenApply(response -> {
               String policy = response.policy();
               logger.info("The response is: " + policy);
               return policy;
           })
           .exceptionally(ex -> {
               throw new RuntimeException("Failed to get key policy", ex);
           });
   }
```

- 有关 API 的详细信息，请参阅 Amazon SDK for Java 2.x API 参考ListKeyPolicies中的。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class KeyPolicy:
    def __init__(self, kms_client):
```

```
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyPolicy":
        """
        Creates a KeyPolicy instance with a default KMS client.

        :return: An instance of KeyPolicy initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def list_policies(self, key_id):
        """
        Lists the names of the policies for a key.

        :param key_id: The ARN or ID of the key to query.
        """
        try:
            policy_names = self.kms_client.list_key_policies(KeyId=key_id)[
                "PolicyNames"
            ]
        except ClientError as err:
            logging.error(
                "Couldn't list your policies. Here's why: %s",
                err.response["Error"]["Message"],
            )
            raise
        else:
            print(f"The policies for key {key_id} are:")
            pprint(policy_names)
```

- 有关 API 的详细信息，请参阅适用ListKeyPolicies于 Python 的Amazon SDK (Boto3) API 参考。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

`ListKeys`与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 `ListKeys`。

操作示例是大型程序的代码摘录，必须在上下文中运行。在以下代码示例中，您可以查看此操作的上下文：

- 了解基本功能

.NET

适用于 .NET 的 Amazon SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
   using System;
   using System.Threading.Tasks;
   using Amazon.KeyManagementService;
   using Amazon.KeyManagementService.Model;

   /// <summary>
   /// List the AWS Key Managements Service (AWS KMS) keys for the AWS Region
   /// of the default user. To list keys in another AWS Region, supply the
Region
   /// as a parameter to the client constructor.
   /// </summary>
   public class ListKeys
   {
       public static async Task Main()
       {
           var client = new AmazonKeyManagementServiceClient();
           var request = new ListKeysRequest();
           var response = new ListKeysResponse();

           do
           {
               response = await client.ListKeysAsync(request);
```

```
                response.Keys.ForEach(key =>
                {
                    Console.WriteLine($"ID: {key.KeyId}, {key.KeyArn}");
                });

                // Set the Marker property when response.Truncated is true
                // in order to get the next keys.
                request.Marker = response.NextMarker;
            }
            while (response.Truncated);
        }
    }
```

- 有关 API 的详细信息，请参阅 适用于 .NET 的 Amazon SDK API 参考ListKeys中的。

CLI

Amazon CLI

获取账户和区域中的 KMS 密钥

以下 `list-keys` 示例获取账户和区域中的 KMS 密钥。此命令同时返回 Amazon 托管密钥和客户托管密钥。

```
aws kms list-keys
```

输出：

```
{
    "Keys": [
        {
            "KeyArn": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
            "KeyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
        },
        {
            "KeyArn": "arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321",
            "KeyId": "0987dcba-09fe-87dc-65ba-ab0987654321"
        },
```

```
        {
            "KeyArn": "arn:aws:kms:us-
east-2:111122223333:key/1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d",
            "KeyId": "1a2b3c4d-5e6f-1a2b-3c4d-5e6f1a2b3c4d"
        }
    ]
}
```

有关更多信息，请参阅《Amazon 密钥管理服务开发人员指南》中的查看密钥。

- 有关 API 的详细信息，请参阅Amazon CLI 命令参考ListKeys中的。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
import software.amazon.awssdk.services.kms.KmsAsyncClient;
import software.amazon.awssdk.services.kms.model.ListKeysRequest;
import software.amazon.awssdk.services.kms.paginators.ListKeysPublisher;
import java.util.concurrent.CompletableFuture;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
started.html
 */
public class HelloKMS {
    public static void main(String[] args) {
        listAllKeys();
    }

    public static void listAllKeys() {
```

```
        KmsAsyncClient kmsAsyncClient = KmsAsyncClient.builder()
            .build();
        ListKeysRequest listKeysRequest = ListKeysRequest.builder()
            .limit(15)
            .build();

        /*
         * The `subscribe` method is required when using paginator methods in the
AWS SDK
         * because paginator methods return an instance of a `ListKeysPublisher`,
which is
         * based on a reactive stream. This allows asynchronous retrieval of
paginated
         * results as they become available. By subscribing to the stream, we can
process
         * each page of results as they are emitted.
         */
        ListKeysPublisher keysPublisher =
kmsAsyncClient.listKeysPaginator(listKeysRequest);
        CompletableFuture<Void> future = keysPublisher
            .subscribe(r -> r.keys().forEach(key ->
                System.out.println("The key ARN is: " + key.keyArn() + ". The key
Id is: " + key.keyId())))
            .whenComplete((result, exception) -> {
                if (exception != null) {
                    System.err.println("Error occurred: " +
exception.getMessage());
                } else {
                    System.out.println("Successfully listed all keys.");
                }
            });

        try {
            future.join();
        } catch (Exception e) {
            System.err.println("Failed to list keys: " + e.getMessage());
        }
    }
}
```

- 有关 API 的详细信息，请参阅 Amazon SDK for Java 2.x API 参考ListKeys中的。

Kotlin

适用于 Kotlin 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
suspend fun listAllKeys() {
    val request =
        ListKeysRequest {
            limit = 15
        }

    KmsClient.fromEnvironment { region = "us-west-2" }.use { kmsClient ->
        val response = kmsClient.listKeys(request)
        response.keys?.forEach { key ->
            println("The key ARN is ${key.keyArn}")
            println("The key Id is ${key.keyId}")
        }
    }
}
```

- 有关 API 的详细信息，请参阅适用ListKeys于 K otlin 的Amazon SDK API 参考。

PHP

适用于 PHP 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /***
```

```
     * @return array
     */
    public function listKeys()
    {
        try {
            $contents = [];
            $paginator = $this->client->getPaginator("ListKeys");
            foreach($paginator as $result){
                foreach ($result['Content'] as $object) {
                    $contents[] = $object;
                }
            }
            return $contents;
        }catch(KmsException $caught){
            echo "There was a problem listing the keys: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }
```

- 有关 API 的详细信息，请参阅 适用于 PHP 的 Amazon SDK API 参考ListKeys中的。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
```

```
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def list_keys(self):
        """
        Lists the keys for the current account by using a paginator.
        """
        try:
            page_size = 10
            print("\nLet's list your keys.")
            key_paginator = self.kms_client.get_paginator("list_keys")
            for key_page in key_paginator.paginate(PaginationConfig={"PageSize":
10}):
                print(f"Here are {len(key_page['Keys'])} keys:")
                pprint(key_page["Keys"])
                if key_page["Truncated"]:
                    answer = input(
                        f"Do you want to see the next {page_size} keys (y/n)? "
                    )
                    if answer.lower() != "y":
                        break
                else:
                    print("That's all your keys!")
        except ClientError as err:
            logging.error(
                "Couldn't list your keys. Here's why: %s",
                err.response["Error"]["Message"],
            )
```

- 有关 API 的详细信息，请参阅适用ListKeys于 Python 的Amazon SDK (Boto3) API 参考。

Rust

适用于 Rust 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
async fn show_keys(client: &Client) -> Result<(), Error> {
    let resp = client.list_keys().send().await?;

    let keys = resp.keys.unwrap_or_default();

    let len = keys.len();

    for key in keys {
        println!("Key ARN: {}", key.key_arn.as_deref().unwrap_or_default());
    }

    println!();
    println!("Found {} keys", len);

    Ok(())
}
```

- 有关 API 的详细信息，请参阅适用ListKeys于 Rust 的Amazon SDK API 参考。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

`PutKeyPolicy`与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 `PutKeyPolicy`。

CLI

Amazon CLI

更改 KMS 密钥的密钥策略

以下 put-key-policy 示例更改客户托管密钥的密钥策略。

首先，创建密钥策略并将其保存在本地 JSON 文件中。在本示例中，该文件为 key_policy.json。您也可以将密钥策略指定为 policy 参数的字符串值。

此密钥策略中的第一条语句允许 Amazon 账户使用 IAM 策略来控制对 KMS 密钥的访问。第二条语句向 test-user 用户授予针对 KMS 密钥运行 describe-key 和 list-keys 命令的权限。

key_policy.json 的内容：

```
{
    "Version":"2012-10-17",
    "Id" : "key-default-1",
    "Statement" : [
        {
            "Sid" : "Enable IAM User Permissions",
            "Effect" : "Allow",
            "Principal" : {
                "AWS" : "arn:aws:iam::111122223333:root"
            },
            "Action" : "kms:*",
            "Resource" : "*"
        },
        {
            "Sid" : "Allow Use of Key",
            "Effect" : "Allow",
            "Principal" : {
                "AWS" : "arn:aws:iam::111122223333:user/test-user"
            },
            "Action" : [
                "kms:DescribeKey",
                "kms:ListKeys"
            ],
            "Resource" : "*"
        }
    ]
```

```
}
```

为了标识 KMS 密钥，此示例使用了密钥 ID，但您也可以使用密钥 ARN。为了指定密钥策略，该命令使用 policy 参数。为了表示策略位于文件中，它使用所需的 file:// 前缀。需要使用此前缀来识别所有受支持操作系统上的文件。最后，该命令使用值为 default 的 policy-name 参数。如果未指定策略名称，则默认值为 default。唯一有效值为 default。

```
aws kms put-key-policy \
    --policy-name default \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --policy file://key_policy.json
```

此命令不生成任何输出。要验证命令是否有效，请使用 get-key-policy 命令。以下示例命令获取相同 KMS 密钥的密钥策略。值为 text 的 output 参数返回一种易于读取的文本格式。

```
aws kms get-key-policy \
    --policy-name default \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --output text
```

输出：

```
{
    "Version":"2012-10-17",
    "Id" : "key-default-1",
    "Statement" : [
        {
            "Sid" : "Enable IAM User Permissions",
            "Effect" : "Allow",
            "Principal" : {
                "AWS" : "arn:aws:iam::111122223333:root"
            },
            "Action" : "kms:*",
            "Resource" : "*"
            },
            {
            "Sid" : "Allow Use of Key",
            "Effect" : "Allow",
            "Principal" : {
                "AWS" : "arn:aws:iam::111122223333:user/test-user"
            },
            "Action" : [ "kms:Describe", "kms:List" ],
```

```
            "Resource" : "*"
        }
    ]
}
```

有关更多信息，请参阅《Amazon 密钥管理服务开发人员指南》中的更改密钥策略。

- 有关 API 的详细信息，请参阅Amazon CLI 命令参考PutKeyPolicy中的。

PHP

适用于 PHP 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /***
     * @param string $keyId
     * @param string $policy
     * @return void
     */
    public function putKeyPolicy(string $keyId, string $policy)
    {
        try {
            $this->client->putKeyPolicy([
                'KeyId' => $keyId,
                'Policy' => $policy,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem replacing the key policy: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }
```

- 有关 API 的详细信息，请参阅 适用于 PHP 的 Amazon SDK API 参考PutKeyPolicy中的。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class KeyPolicy:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyPolicy":
        """
        Creates a KeyPolicy instance with a default KMS client.

        :return: An instance of KeyPolicy initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def set_policy(self, key_id: str, policy: dict[str, any]) -> None:
        """
        Sets the policy of a key. Setting a policy entirely overwrites the
 existing
        policy, so care is taken to add a statement to the existing list of
 statements
        rather than simply writing a new policy.

        :param key_id: The ARN or ID of the key to set the policy to.
        :param policy: The existing policy of the key.
        :return: None
        """
        principal = input(
            "Enter the ARN of an IAM role to set as the principal on the policy:
 "
        )
```

```
        if key_id != "" and principal != "":
            # The updated policy replaces the existing policy. Add a new
statement to
            # the list along with the original policy statements.
            policy["Statement"].append(
                {
                    "Sid": "Allow access for ExampleRole",
                    "Effect": "Allow",
                    "Principal": {"AWS": principal},
                    "Action": [
                        "kms:Encrypt",
                        "kms:GenerateDataKey*",
                        "kms:Decrypt",
                        "kms:DescribeKey",
                        "kms:ReEncrypt*",
                    ],
                    "Resource": "*",
                }
            )
            try:
                self.kms_client.put_key_policy(KeyId=key_id,
Policy=json.dumps(policy))
            except ClientError as err:
                logger.error(
                    "Couldn't set policy for key %s. Here's why %s",
                    key_id,
                    err.response["Error"]["Message"],
                )
                raise
            else:
                print(f"Set policy for key {key_id}.")
        else:
            print("Skipping set policy demo.")
```

- 有关 API 的详细信息，请参阅适用PutKeyPolicy于 Python 的Amazon SDK (Boto3) API 参考。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

ReEncrypt与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 ReEncrypt。

CLI

Amazon CLI

示例 1：使用不同的对称 KMS 密钥重新加密加密消息（Linux 和 macOS）。

以下re-encrypt命令示例演示了使用 Amazon CLI 重新加密数据的推荐方法。

在文件中提供加密文字。在 --ciphertext-blob 参数的值中，使用 fileb:// 前缀，它将指示 CLI 从二进制文件中读取数据。如果文件不在当前目录中，请键入文件的完整路径。有关从文件读取 Amazon CLI 参数值的更多信息，请参阅《Amazon 命令行界面用户指南》中的从文件加载 Amazon CLI 参数和Amazon 命令行工具博客中的本地文件参数最佳实践。指定源 KMS 密钥，用于解密密文。使用对称加密 KMS 密钥进行解密时不需要--source-key-id参数。 Amazon KMS 可以从密文 blob 中的元数据中获取用于加密数据的 KMS 密钥。但是，指定您正在使用的 KMS 密钥始终是最佳实践。此做法可确保您使用预期的 KMS 密钥，并防止您意外使用不信任的 KMS 密钥解密加密文字。指定目标 KMS 密钥来重新加密数据。--destination-key-id 参数始终为必需项。此示例使用密钥 ARN，但您可以使用任何有效的密钥标识符。将明文输出请求为文本值。--query 参数指示 CLI 仅从输出中获取 Plaintext 字段的值。--output 参数以 text.base64 解码格式返回明文输出并将其保存在文件中。以下示例将 Plaintext 参数的值传送（|）给 Base64 实用工具，该程序负责对其进行解码。然后，它将解码后的输出重定向（>）到 ExamplePlaintext 文件。

在运行此命令之前，请将示例密钥 IDs 替换为 Amazon 账户中的有效密钥标识符。

```
aws kms re-encrypt \
    --ciphertext-blob fileb://ExampleEncryptedFile \
    --source-key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --destination-key-id 0987dcba-09fe-87dc-65ba-ab0987654321 \
    --query CiphertextBlob \
    --output text | base64 --decode > ExampleReEncryptedFile
```

此命令不生成任何输出。re-encrypt 命令的输出经过 base64 解码并保存在文件中。

有关更多信息，请参阅《Amazon 密钥管理服务 API 参考》ReEncrypt中的。

示例 2：使用不同的对称 KMS 密钥重新加密加密消息（Windows 命令提示符）。

以下 re-encrypt 命令示例与前一个示例相同，不同之处在于，它使用 certutil 实用工具对明文数据进行 Base64 解码。此过程需要两个命令，如以下示例所示。

在运行此命令之前，请将示例密钥 ID 替换为 Amazon 账户中的有效密钥 ID。

```
aws kms re-encrypt ^
    --ciphertext-blob fileb://ExampleEncryptedFile ^
    --source-key-id 1234abcd-12ab-34cd-56ef-1234567890ab ^
    --destination-key-id 0987dcba-09fe-87dc-65ba-ab0987654321 ^
    --query CiphertextBlob ^
    --output text > ExampleReEncryptedFile.base64
```

然后使用 certutil 实用工具

```
certutil -decode ExamplePlaintextFile.base64 ExamplePlaintextFile
```

输出：

```
Input Length = 18
Output Length = 12
CertUtil: -decode command completed successfully.
```

有关更多信息，请参阅《Amazon 密钥管理服务 API 参考》ReEncrypt中的。

- 有关 API 的详细信息，请参阅Amazon CLI 命令参考ReEncrypt中的。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class KeyEncrypt:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
```

```
    def from_client(cls) -> "KeyEncrypt":
        """
        Creates a KeyEncrypt instance with a default KMS client.

        :return: An instance of KeyEncrypt initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def re_encrypt(self, source_key_id, cipher_text):
        """
        Takes ciphertext previously encrypted with one key and reencrypt it by
using
        another key.

        :param source_key_id: The ARN or ID of the original key used to encrypt
the
                              ciphertext.
        :param cipher_text: The encrypted ciphertext.
        :return: The ciphertext encrypted by the second key.
        """
        destination_key_id = input(
            f"Your ciphertext is currently encrypted with key {source_key_id}. "
            f"Enter another key ID or ARN to reencrypt it: "
        )
        if destination_key_id != "":
            try:
                cipher_text = self.kms_client.re_encrypt(
                    SourceKeyId=source_key_id,
                    DestinationKeyId=destination_key_id,
                    CiphertextBlob=cipher_text,
                )["CiphertextBlob"]
            except ClientError as err:
                logger.error(
                    "Couldn't reencrypt your ciphertext. Here's why: %s",
                    err.response["Error"]["Message"],
                )
            else:
                print(f"Reencrypted your ciphertext as: {cipher_text}")
                return cipher_text
        else:
            print("Skipping reencryption demo.")
```

- 有关 API 的详细信息，请参阅适用ReEncrypt于 Python 的Amazon SDK (Boto3) API 参考。

Ruby

适用于 Ruby 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
require 'aws-sdk-kms' # v2: require 'aws-sdk'

# Human-readable version of the ciphertext of the data to reencrypt.

blob =
 '01020200785d68faeec386af1057904926253051eb2919d3c16078badf65b808b26dd057c101747cadf3593
sourceCiphertextBlob = [blob].pack('H*')

# Replace the fictitious key ARN with a valid key ID

destinationKeyId = 'arn:aws:kms:us-
west-2:111122223333:key/0987dcba-09fe-87dc-65ba-ab0987654321'

client = Aws::KMS::Client.new(region: 'us-west-2')

resp = client.re_encrypt({
                           ciphertext_blob: sourceCiphertextBlob,
                           destination_key_id: destinationKeyId
                         })

# Display a readable version of the resulting re-encrypted blob.
puts 'Blob:'
puts resp.ciphertext_blob.unpack('H*')
```

- 有关 API 的详细信息，请参阅 适用于 Ruby 的 Amazon SDK API 参考ReEncrypt中的。

Rust

适用于 Rust 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
async fn reencrypt_string(
    verbose: bool,
    client: &Client,
    input_file: &str,
    output_file: &str,
    first_key: &str,
    new_key: &str,
) -> Result<(), Error> {
    // Get blob from input file
    // Open input text file and get contents as a string
    // input is a base-64 encoded string, so decode it:
    let data = fs::read_to_string(input_file)
        .map(|input_file| base64::decode(input_file).expect("invalid base 64"))
        .map(Blob::new);

    let resp = client
        .re_encrypt()
        .ciphertext_blob(data.unwrap())
        .source_key_id(first_key)
        .destination_key_id(new_key)
        .send()
        .await?;

    // Did we get an encrypted blob?
    let blob = resp.ciphertext_blob.expect("Could not get encrypted text");
    let bytes = blob.as_ref();

    let s = base64::encode(bytes);
    let o = &output_file;
```

```
    let mut ofile = File::create(o).expect("unable to create file");
    ofile.write_all(s.as_bytes()).expect("unable to write");

    if verbose {
        println!("Wrote the following to {}:", output_file);
        println!("{}", s);
    } else {
        println!("Wrote base64-encoded output to {}", output_file);
    }

    Ok(())
}
```

- 有关 API 的详细信息，请参阅适用ReEncrypt于 Rust 的Amazon SDK API 参考。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

`RetireGrant`与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 `RetireGrant`。

CLI

Amazon CLI

停用对客户主密钥的授权

以下 `retire-grant` 示例从 KMS 密钥中删除授权。

以下示例命令指定 `grant-id` 和 `key-id` 参数。`key-id` 参数值必须为 KMS 密钥的密钥 ARN。

```
aws kms retire-grant \
    --grant-id 1234a2345b8a4e350500d432bccf8ecd6506710e1391880c4f7f7140160c9af3 \
    --key-id arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab
```

此命令不生成任何输出。要确认授权是否已停用，请使用 `list-grants` 命令。

有关更多信息，请参阅《Amazon 密钥管理服务开发人员指南》中的停用和撤销授权。

- 有关 API 的详细信息，请参阅Amazon CLI 命令参考RetireGrant中的。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class GrantManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "GrantManager":
        """
        Creates a GrantManager instance with a default KMS client.

        :return: An instance of GrantManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def retire_grant(self, grant):
        """
        Retires a grant so that it can no longer be used.

        :param grant: The grant to retire.
        """
        try:
            self.kms_client.retire_grant(GrantToken=grant["GrantToken"])
        except ClientError as err:
            logger.error(
                "Couldn't retire grant %s. Here's why: %s",
                grant["GrantId"],
                err.response["Error"]["Message"],
            )
```

```
        else:
            print(f"Grant {grant['GrantId']} retired.")
```

- 有关 API 的详细信息，请参阅适用RetireGrant于 Python 的Amazon SDK (Boto3) API 参考。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

`RevokeGrant`与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 `RevokeGrant`。

操作示例是大型程序的代码摘录，必须在上下文中运行。在以下代码示例中，您可以查看此操作的上下文：

- 了解基本功能

CLI

Amazon CLI

撤销对客户主密钥的授权

以下 `revoke-grant` 示例从 KMS 密钥中删除授权。以下示例命令指定 `grant-id` 和 `key-id` 参数。`key-id` 参数的值可以是 KMS 密钥的密钥 ID 或密钥 ARN。

```
aws kms revoke-grant \
    --grant-id 1234a2345b8a4e350500d432bccf8ecd6506710e1391880c4f7f7140160c9af3 \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

此命令不生成任何输出。要确认授权是否已撤销，请使用 `list-grants` 命令。

有关更多信息，请参阅《Amazon 密钥管理服务开发人员指南》中的停用和撤销授权。

- 有关 API 的详细信息，请参阅Amazon CLI 命令参考RevokeGrant中的。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
   /**
    * Revokes a grant for the specified AWS KMS key asynchronously.
    *
    * @param keyId   The ID or key ARN of the AWS KMS key.
    * @param grantId The identifier of the grant to be revoked.
    * @return A {@link CompletableFuture} representing the asynchronous
operation of revoking the grant.
    *         The {@link CompletableFuture} will complete with a {@link
RevokeGrantResponse} object
    *         if the operation is successful, or with a {@code null} value if an
error occurs.
    */
   public CompletableFuture<RevokeGrantResponse> revokeKeyGrantAsync(String
keyId, String grantId) {
       RevokeGrantRequest grantRequest = RevokeGrantRequest.builder()
           .keyId(keyId)
           .grantId(grantId)
           .build();

       CompletableFuture<RevokeGrantResponse> responseFuture =
getAsyncClient().revokeGrant(grantRequest);
       responseFuture.whenComplete((response, exception) -> {
           if (exception == null) {
               logger.info("Grant ID: [" + grantId + "] was successfully
revoked!");
           } else {
               if (exception instanceof KmsException kmsEx) {
                   if (kmsEx.getMessage().contains("Grant does not exist")) {
                       logger.info("The grant ID '" + grantId + "' does not
exist. Moving on...");
                   } else {
```

```
                        throw new RuntimeException("KMS error occurred: " +
 kmsEx.getMessage(), kmsEx);
                    }
                } else {
                    throw new RuntimeException("An unexpected error occurred: " +
 exception.getMessage(), exception);
                }
            }
        });

        return responseFuture;
    }
```

- 有关 API 的详细信息，请参阅 Amazon SDK for Java 2.x API 参考RevokeGrant中的。

PHP

适用于 PHP 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /***
     * @param string $grantId
     * @param string $keyId
     * @return void
     */
    public function revokeGrant(string $grantId, string $keyId)
    {
        try{
            $this->client->revokeGrant([
                'GrantId' => $grantId,
                'KeyId' => $keyId,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem with revoking the grant: {$caught-
>getAwsErrorMessage()}.\n";
```

```
            throw $caught;
        }
    }
```

- 有关 API 的详细信息，请参阅 适用于 PHP 的 Amazon SDK API 参考RevokeGrant中的。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class GrantManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "GrantManager":
        """
        Creates a GrantManager instance with a default KMS client.

        :return: An instance of GrantManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def revoke_grant(self, key_id: str, grant_id: str) -> None:
        """
        Revokes a grant so that it can no longer be used.

        :param key_id: The ARN or ID of the key associated with the grant.
        :param grant_id: The ID of the grant to revoke.
        """
        try:
```

```
            self.kms_client.revoke_grant(KeyId=key_id, GrantId=grant_id)
        except ClientError as err:
            logger.error(
                "Couldn't revoke grant %s. Here's why: %s",
                grant_id,
                err.response["Error"]["Message"],
            )
            raise
```

- 有关 API 的详细信息，请参阅适用RevokeGrant于 Python 的Amazon SDK (Boto3) API 参考。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

ScheduleKeyDeletion与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 `ScheduleKeyDeletion`。

操作示例是大型程序的代码摘录，必须在上下文中运行。在以下代码示例中，您可以查看此操作的上下文：

- 了解基本功能

CLI

Amazon CLI

计划删除客户托管的 KMS 密钥。

以下 `schedule-key-deletion` 示例计划在 15 天后删除指定的客户托管 KMS 密钥。

`--key-id` 参数识别 KMS 密钥。此示例使用密钥 ARN 值，但您可以使用 KMS 密钥的密钥 ID 或 ARN。`--pending-window-in-days` 参数指定 7-30 天的等待期限。默认的等待期限为 30 天。此示例指定的值为 15，表示在命令完成 15 天后永久删除 KMS 密钥。 Amazon

```
aws kms schedule-key-deletion \
```

```
    --key-id arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab \
    --pending-window-in-days 15
```

响应包括密钥 ARN、密钥状态、等待期（PendingWindowInDays）和删除日期（以 Unix 时间表示）。要以当地时间查看删除日期，请使用 Amazon KMS 控制台。无法在加密操作中使用密钥状态为 PendingDeletion 的 KMS 密钥。

```
{
    "KeyId": "arn:aws:kms:us-
west-2:123456789012:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "DeletionDate": "2022-06-18T23:43:51.272000+00:00",
    "KeyState": "PendingDeletion",
    "PendingWindowInDays": 15
}
```

有关更多信息，请参阅《Amazon 密钥管理服务开发人员指南》中的删除密钥。

- 有关 API 的详细信息，请参阅Amazon CLI 命令参考ScheduleKeyDeletion中的。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /**
     * Deletes a KMS key asynchronously.
     *
     * <p><strong>Warning:</strong> Deleting a KMS key is a destructive and
potentially dangerous operation.
     * When a KMS key is deleted, all data that was encrypted under the KMS key
becomes unrecoverable.
     * This means that any files, databases, or other data that were encrypted
using the deleted KMS key
     * will become permanently inaccessible. Exercise extreme caution when
deleting KMS keys.</p>
```

```
     *
     * @param keyId the ID of the KMS key to delete
     * @return a {@link CompletableFuture} that completes when the key deletion
is scheduled
     */
    public CompletableFuture<Void> deleteKeyAsync(String keyId) {
        ScheduleKeyDeletionRequest deletionRequest =
ScheduleKeyDeletionRequest.builder()
            .keyId(keyId)
            .pendingWindowInDays(7)
            .build();

        return getAsyncClient().scheduleKeyDeletion(deletionRequest)
            .thenRun(() -> {
                logger.info("Key {} will be deleted in 7 days", keyId);
            })
            .exceptionally(throwable -> {
                throw new RuntimeException("Failed to schedule key deletion for
key ID: " + keyId, throwable);
            });
    }
```

- 有关 API 的详细信息，请参阅 Amazon SDK for Java 2.x API 参考[ScheduleKeyDeletion](#)中的。

PHP

适用于 PHP 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 [Amazon 代码示例存储库](#)中查找完整示例，了解如何进行设置和运行。

```
    /**
     * @param string $keyId
     * @param int $pendingWindowInDays
     * @return void
```

```
     */
    public function scheduleKeyDeletion(string $keyId, int $pendingWindowInDays =
 7)
    {
        try {
            $this->client->scheduleKeyDeletion([
                'KeyId' => $keyId,
                'PendingWindowInDays' => $pendingWindowInDays,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem scheduling the key deletion: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }
```

- 有关 API 的详细信息，请参阅 适用于 PHP 的 Amazon SDK API 参考[ScheduleKeyDeletion](#)中的。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 还有更多相关信息 GitHub。在 [Amazon 代码示例存储库](#)中查找完整示例，了解如何进行设置和运行。

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.
```

```
        :return: An instance of KeyManager initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def delete_key(self, key_id: str, window: int) -> None:
        """
        Deletes a list of keys.

        Warning:
        Deleting a KMS key is a destructive and potentially dangerous operation.
When a KMS key is deleted,
        all data that was encrypted under the KMS key is unrecoverable.

        :param key_id: The ARN or ID of the key to delete.
        :param window: The waiting period, in days, before the KMS key is
deleted.
        """

        try:
            self.kms_client.schedule_key_deletion(
                KeyId=key_id, PendingWindowInDays=window
            )
        except ClientError as err:
            logging.error(
                "Couldn't delete key %s. Here's why: %s",
                key_id,
                err.response["Error"]["Message"],
            )
            raise
```

- 有关 API 的详细信息，请参阅适用ScheduleKeyDeletion于 Python 的Amazon SDK (Boto3) API 参考。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

Sign与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 Sign。

操作示例是大型程序的代码摘录，必须在上下文中运行。在以下代码示例中，您可以查看此操作的上下文：

- 了解基本功能

CLI

Amazon CLI

示例 1：为消息生成数字签名

以下 sign 示例为一条短消息生成加密签名。该命令的输出包括一个 base-64 编码的 Signature 字段，您可以使用 verify 命令对其进行验证。

您必须指定要签名的消息以及您的非对称 KMS 密钥支持的签名算法。要获取 KMS 密钥的签名算法，请使用 describe-key 命令。

在 Amazon CLI v2 中，message参数的值必须采用 Base64 编码。或者，您可以将消息保存在文件中并使用fileb://前缀，它告诉 Amazon CLI 从文件中读取二进制数据。

在运行此命令之前，请将示例密钥 ID 替换为 Amazon 账户中的有效密钥 ID。密钥 ID 必须表示密钥用法为 SIGN_VERIFY 的非对称 KMS 密钥。

```
msg=(echo 'Hello World' | base64)

aws kms sign \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --message fileb://UnsignedMessage \
    --message-type RAW \
    --signing-algorithm RSASSA_PKCS1_V1_5_SHA_256
```

输出：

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
```

```
    "Signature": "ABCDEFhpyVYyTxbafE74ccSvEJLJr3zuoV1Hfymz4qv+/
fxmxNLA7SE1SiF8lHw80fKZZ3bJ...",
    "SigningAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
}
```

有关在 Amazon KMS 中使用非对称 KMS 密钥的更多信息，请参阅《密钥管理服务开发人员指南》中的 Amazon KMS 中的非对称Amazon密钥。

示例 2：将数字签名保存在文件中（Linux 和 macOS）

以下 sign 示例为一条存储在本地文件中的短消息生成加密签名。该命令还会从响应中获取Signature属性，Base64 对其进行解码并将其保存在文件中。 ExampleSignature 您可以在验证签名的 verify 命令中使用该签名文件。

sign 命令需要一条 Base64 编码的消息和您的非对称 KMS 密钥支持的签名算法。要获取 KMS 密钥支持的签名算法，请使用 describe-key 命令。

在运行此命令之前，请将示例密钥 ID 替换为 Amazon 账户中的有效密钥 ID。密钥 ID 必须表示密钥用法为 SIGN_VERIFY 的非对称 KMS 密钥。

```
echo 'hello world' | base64 > EncodedMessage

aws kms sign \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --message fileb://EncodedMessage \
    --message-type RAW \
    --signing-algorithm RSASSA_PKCS1_V1_5_SHA_256 \
    --output text \
    --query Signature | base64 --decode > ExampleSignature
```

此命令不生成任何输出。此示例提取输出的 Signature 属性并将其保存在文件中。

有关在 Amazon KMS 中使用非对称 KMS 密钥的更多信息，请参阅《密钥管理服务开发人员指南》中的 Amazon KMS 中的非对称Amazon密钥。

- 有关 API 详细信息，请参阅《Amazon CLI 命令参考》中的 Sign。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /**
     * Asynchronously signs and verifies data using AWS KMS.
     *
     * <p>The method performs the following steps:
     * <ol>
     *     <li>Creates an AWS KMS key with the specified key spec, key usage, and
origin.</li>
     *     <li>Signs the provided message using the created KMS key and the
RSASSA-PSS-SHA-256 algorithm.</li>
     *     <li>Verifies the signature of the message using the created KMS key
and the RSASSA-PSS-SHA-256 algorithm.</li>
     * </ol>
     *
     * @return a {@link CompletableFuture} that completes with the result of the
signature verification,
     *         {@code true} if the signature is valid, {@code false} otherwise.
     * @throws KmsException if any error occurs during the KMS operations.
     * @throws RuntimeException if an unexpected error occurs.
     */
    public CompletableFuture<Boolean> signVerifyDataAsync() {
        String signMessage = "Here is the message that will be digitally signed";

        // Create an AWS KMS key used to digitally sign data.
        CreateKeyRequest createKeyRequest = CreateKeyRequest.builder()
            .keySpec(KeySpec.RSA_2048)
            .keyUsage(KeyUsageType.SIGN_VERIFY)
            .origin(OriginType.AWS_KMS)
            .build();

        return getAsyncClient().createKey(createKeyRequest)
            .thenCompose(createKeyResponse -> {
                String keyId = createKeyResponse.keyMetadata().keyId();
```

```
                SdkBytes messageBytes = SdkBytes.fromString(signMessage,
Charset.defaultCharset());
                SignRequest signRequest = SignRequest.builder()
                    .keyId(keyId)
                    .message(messageBytes)
                    .signingAlgorithm(SigningAlgorithmSpec.RSASSA_PSS_SHA_256)
                    .build();

                return getAsyncClient().sign(signRequest)
                    .thenCompose(signResponse -> {
                        byte[] signedBytes =
signResponse.signature().asByteArray();

                        VerifyRequest verifyRequest = VerifyRequest.builder()
                            .keyId(keyId)

.message(SdkBytes.fromByteArray(signMessage.getBytes(Charset.defaultCharset())))

.signature(SdkBytes.fromByteBuffer(ByteBuffer.wrap(signedBytes)))

.signingAlgorithm(SigningAlgorithmSpec.RSASSA_PSS_SHA_256)
                            .build();

                        return getAsyncClient().verify(verifyRequest)
                            .thenApply(verifyResponse -> {
                                return (boolean) verifyResponse.signatureValid();
                            });
                    });
            })
            .exceptionally(throwable -> {
               throw new RuntimeException("Failed to sign or verify data",
throwable);
            });
    }
```

- 有关 API 详细信息，请参阅《Amazon SDK for Java 2.x API Reference》中的 Sign。

PHP

适用于 PHP 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /**
     * @param string $keyId
     * @param string $message
     * @param string $algorithm
     * @return Result
     */
    public function sign(string $keyId, string $message, string $algorithm)
    {
        try {
            return $this->client->sign([
                'KeyId' => $keyId,
                'Message' => $message,
                'SigningAlgorithm' => $algorithm,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem signing the data: {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }
```

- 有关 API 详细信息，请参阅《适用于 PHP 的 Amazon SDK API Reference》中的 Sign。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class KeyEncrypt:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyEncrypt":
        """
        Creates a KeyEncrypt instance with a default KMS client.

        :return: An instance of KeyEncrypt initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def sign(self, key_id: str, message: str) -> str:
        """
        Signs a message with a key.

        :param key_id: The ARN or ID of the key to use for signing.
        :param message: The message to sign.
        :return: The signature of the message.
        """
        try:
            return self.kms_client.sign(
                KeyId=key_id,
                Message=message.encode(),
                SigningAlgorithm="RSASSA_PSS_SHA_256",
            )["Signature"]
        except ClientError as err:
            logger.error(
```

```
                "Couldn't sign your message. Here's why: %s",
                err.response["Error"]["Message"],
            )
            raise
```

- 有关 API 详细信息，请参阅《Amazon SDK for Python（Boto3）API Reference》中的 Sign。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

`TagResource`与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 `TagResource`。

操作示例是大型程序的代码摘录，必须在上下文中运行。在以下代码示例中，您可以查看此操作的上下文：

- 了解基本功能

CLI

Amazon CLI

向 KMS 密钥添加标签

以下 `tag-resource` 示例为客户托管的 KMS 密钥添加 `"Purpose":"Test"` 和 `"Dept":"IT"` 标签。您可以使用此类标签来标记 KMS 密钥并创建 KMS 密钥类别以进行权限管理和审计。

要指定 KMS 密钥，请使用 `key-id` 参数。此示例使用密钥 ID 值，但您可以在此命令中使用密钥 ID 或密钥 ARN。

```
aws kms tag-resource \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --tags TagKey='Purpose',TagValue='Test' TagKey='Dept',TagValue='IT'
```

此命令不生成任何输出。要查看 Amazon KMS KMS 密钥上的标签，请使用`list-resource-tags`命令。

有关在 Amazon KMS 中使用标签的更多信息，请参阅《密钥管理服务开发人员指南》中的为密 Amazon 钥添加标签。

- 有关 API 的详细信息，请参阅Amazon CLI 命令参考TagResource中的。

Java

适用于 Java 的 SDK 2.x

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /**
     * Asynchronously tags a KMS key with a specific tag.
     *
     * @param keyId the ID of the KMS key to be tagged
     * @return a {@link CompletableFuture} that completes when the tagging
operation is finished
     */
    public CompletableFuture<Void> tagKMSKeyAsync(String keyId) {
        Tag tag = Tag.builder()
            .tagKey("Environment")
            .tagValue("Production")
            .build();

        TagResourceRequest tagResourceRequest = TagResourceRequest.builder()
            .keyId(keyId)
            .tags(tag)
            .build();

        return getAsyncClient().tagResource(tagResourceRequest)
            .thenRun(() -> {
                logger.info("{} key was tagged", keyId);
            })
            .exceptionally(throwable -> {
                throw new RuntimeException("Failed to tag the KMS key",
throwable);
            });
    }
```

- 有关 API 的详细信息，请参阅 Amazon SDK for Java 2.x API 参考TagResource中的。

PHP

适用于 PHP 的 SDK

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
    /***
     * @param string $keyId
     * @param array $tags
     * @return void
     */
    public function tagResource(string $keyId, array $tags)
    {
        try {
            $this->client->tagResource([
                'KeyId' => $keyId,
                'Tags' => $tags,
            ]);
        }catch(KmsException $caught){
            echo "There was a problem applying the tag(s): {$caught-
>getAwsErrorMessage()}\n";
            throw $caught;
        }
    }
```

- 有关 API 的详细信息，请参阅 适用于 PHP 的 Amazon SDK API 参考TagResource中的。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class KeyManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_keys = []

    @classmethod
    def from_client(cls) -> "KeyManager":
        """
        Creates a KeyManager instance with a default KMS client.

        :return: An instance of KeyManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def tag_resource(self, key_id: str, tag_key: str, tag_value: str) -> None:
        """
        Add or edit tags on a customer managed key.

        :param key_id: The ARN or ID of the key to enable rotation for.
        :param tag_key: Key for the tag.
        :param tag_value: Value for the tag.
        """
        try:
            self.kms_client.tag_resource(
                KeyId=key_id, Tags=[{"TagKey": tag_key, "TagValue": tag_value}]
            )
        except ClientError as err:
            logging.error(
                "Couldn't add a tag for the key '%s'. Here's why: %s",
```

```
                key_id,
                err.response["Error"]["Message"],
            )
            raise
```

- 有关 API 的详细信息，请参阅适用TagResource于 Python 的Amazon SDK (Boto3) API 参考。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

`UpdateAlias`与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 `UpdateAlias`。

CLI

Amazon CLI

将别名与其他 KMS 密钥关联

以下 `update-alias` 示例将别名 `alias/test-key` 与其他 KMS 密钥关联起来。

`--alias-name` 参数指定别名。别名名称值必须以 `alias/` 开头。`--target-key-id` 参数指定要与别名关联的 KMS 密钥。您不需要为别名指定当前 KMS 密钥。

```
aws kms update-alias \
    --alias-name alias/test-key \
    --target-key-id 1234abcd-12ab-34cd-56ef-1234567890ab
```

此命令不生成任何输出。要查找别名，请使用 `list-aliases` 命令。

有关更多信息，请参阅《Amazon 密钥管理服务开发人员指南》中的更新别名。

- 有关 API 的详细信息，请参阅Amazon CLI 命令参考UpdateAlias中的。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class AliasManager:
    def __init__(self, kms_client):
        self.kms_client = kms_client
        self.created_key = None

    @classmethod
    def from_client(cls) -> "AliasManager":
        """
        Creates an AliasManager instance with a default KMS client.

        :return: An instance of AliasManager initialized with the default KMS
 client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def update_alias(self, alias, current_key_id):
        """
        Updates an alias by assigning it to another key.

        :param alias: The alias to reassign.
        :param current_key_id: The ARN or ID of the key currently associated with
 the alias.
        """
        new_key_id = input(
            f"Alias {alias} is currently associated with {current_key_id}. "
            f"Enter another key ID or ARN that you want to associate with
 {alias}: "
        )
        if new_key_id != "":
            try:
```

```
                self.kms_client.update_alias(AliasName=alias,
TargetKeyId=new_key_id)
            except ClientError as err:
                logger.error(
                    "Couldn't associate alias %s with key %s. Here's why: %s",
                    alias,
                    new_key_id,
                    err.response["Error"]["Message"],
                )
            else:
                print(f"Alias {alias} is now associated with key {new_key_id}.")
        else:
            print("Skipping alias update.")
```

- 有关 API 的详细信息，请参阅适用UpdateAlias于 Python 的Amazon SDK (Boto3) API 参考。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

`Verify`与 Amazon SDK 或 CLI 配合使用

以下代码示例演示如何使用 `Verify`。

CLI

Amazon CLI

验证数字签名

以下 `verify` 命令验证一条 Base64 编码短消息的加密签名。密钥 ID、消息、消息类型和签名算法必须与用于签名该消息的相同。

在 Amazon CLI v2 中，`message`参数的值必须采用 Base64 编码。或者，您可以将消息保存在文件中并使用`fileb://`前缀，它告诉 Amazon CLI 从文件中读取二进制数据。

您指定的签名不得采用 base64 编码。如需解码 `sign` 命令返回的签名的帮助，请参阅 `sign` 命令示例。

该命令的输出包括一个布尔 `SignatureValid` 字段，表示签名已通过验证。如果签名验证失败，`verify` 命令也会失败。

在运行此命令之前，请将示例密钥 ID 替换为 Amazon 账户中的有效密钥 ID。

```
aws kms verify \
    --key-id 1234abcd-12ab-34cd-56ef-1234567890ab \
    --message fileb://EncodedMessage \
    --message-type RAW \
    --signing-algorithm RSASSA_PKCS1_V1_5_SHA_256 \
    --signature fileb://ExampleSignature
```

输出：

```
{
    "KeyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
    "SignatureValid": true,
    "SigningAlgorithm": "RSASSA_PKCS1_V1_5_SHA_256"
}
```

有关在 KMS 中 Amazon 使用非对称 KMS 密钥的更多信息，请参阅密钥管理服务开发人员指南中的Amazon 使用非对称密钥。

- 有关 API 详细信息，请参阅《Amazon CLI 命令参考》中的 Verify。

Python

适用于 Python 的 SDK（Boto3）

> **Note**
>
> 还有更多相关信息 GitHub。在 Amazon 代码示例存储库中查找完整示例，了解如何进行设置和运行。

```
class KeyEncrypt:
    def __init__(self, kms_client):
        self.kms_client = kms_client

    @classmethod
    def from_client(cls) -> "KeyEncrypt":
        """
```

```
        Creates a KeyEncrypt instance with a default KMS client.

        :return: An instance of KeyEncrypt initialized with the default KMS
client.
        """
        kms_client = boto3.client("kms")
        return cls(kms_client)


    def verify(self, key_id: str, message: str, signature: str) -> bool:
        """
        Verifies a signature against a message.

        :param key_id: The ARN or ID of the key used to sign the message.
        :param message: The message to verify.
        :param signature: The signature to verify.
        :return: True when the signature matches the message, otherwise False.
        """
        try:
            response = self.kms_client.verify(
                KeyId=key_id,
                Message=message.encode(),
                Signature=signature,
                SigningAlgorithm="RSASSA_PSS_SHA_256",
            )
            valid = response["SignatureValid"]
            print(f"The signature is {'valid' if valid else 'invalid'}.")
            return valid
        except ClientError as err:
            if err.response["Error"]["Code"] == "SignatureDoesNotMatchException":
                print("The signature is not valid.")
            else:
                logger.error(
                    "Couldn't verify your signature. Here's why: %s",
                    err.response["Error"]["Message"],
                )
            raise
```

- 有关 API 详细信息，请参阅《Amazon SDK for Python（Boto3）API Reference》中的 Verify。

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

Amazon KMS 使用场景 Amazon SDKs

以下代码示例向您展示了如何在中 Amazon KMS 实现常见场景 Amazon SDKs。这些场景向您展示了如何通过在其中调用多个函数 Amazon KMS 或与其他函数组合来完成特定任务 Amazon Web Services 服务。每个场景都包含完整源代码的链接，您可以在其中找到有关如何设置和运行代码的说明。

场景以中等水平的经验为目标，可帮助您结合具体环境了解服务操作。

示例

- 使用 v2 使用 DynamoDB 表加密 Amazon Command Line Interface

使用 v2 使用 DynamoDB 表加密 Amazon Command Line Interface

以下代码示例演示了如何管理 DynamoDB 表的加密选项。

- 创建具有默认加密的表。
- 创建具有客户托管式 CMK 的表。
- 更新表加密设置。
- 描述表加密。

Bash

Amazon CLI 使用 Bash 脚本

创建具有默认加密的表。

```
# Create a table with default encryption (AWS owned key)
aws dynamodb create-table \
    --table-name CustomerData \
    --attribute-definitions \
        AttributeName=CustomerID,AttributeType=S \
    --key-schema \
        AttributeName=CustomerID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
```

```
    --sse-specification Enabled=true,SSEType=KMS
```

创建具有客户托管式 CMK 的表。

```
# Step 1: Create a customer managed key in KMS
aws kms create-key \
    --description "Key for DynamoDB table encryption" \
    --key-usage ENCRYPT_DECRYPT \
    --customer-master-key-spec SYMMETRIC_DEFAULT

# Store the key ID for later use
KEY_ID=$(aws kms list-keys --query "Keys[?contains(KeyArn, 'Key for
 DynamoDB')].KeyId" --output text)

# Step 2: Create a table with the customer managed key
aws dynamodb create-table \
    --table-name SensitiveData \
    --attribute-definitions \
        AttributeName=RecordID,AttributeType=S \
    --key-schema \
        AttributeName=RecordID,KeyType=HASH \
    --billing-mode PAY_PER_REQUEST \
    --sse-specification Enabled=true,SSEType=KMS,KMSMasterKeyId=$KEY_ID
```

更新表加密。

```
# Update a table to use a different KMS key
aws dynamodb update-table \
    --table-name CustomerData \
    --sse-specification Enabled=true,SSEType=KMS,KMSMasterKeyId=$KEY_ID
```

描述表加密。

```
# Describe the table to see encryption settings
aws dynamodb describe-table \
    --table-name CustomerData \
    --query "Table.SSEDescription"
```

- 有关 API 详细信息，请参阅《Amazon CLI 命令参考》中的以下主题。
  - CreateKey
  - CreateTable
  - DescribeTable
  - UpdateTable

有关 S Amazon DK 开发者指南和代码示例的完整列表，请参阅将此服务与 Amazon SDK 配合使用。本主题还包括有关入门的信息以及有关先前的 SDK 版本的详细信息。

Amazon KMS 中的密码认证支持

Amazon KMS 支持将密码认证用于 Amazon Nitro Enclaves 和 Amazon NitroTPM。支持这些认证方法的应用程序使用已签名的认证文档调用以下 Amazon KMS 密码操作。Amazon KMS 会验证认证文档是否来自有效来源（某个 Nitro 飞地或 NitroTPM）。然后，这些 API 不会在响应中返回明文数据，而是使用认证文档中的公有密钥对明文进行加密，并返回只能通过该飞地或 EC2 实例中的对应私有密钥解密的加密文字。

- Decrypt
- DeriveSharedSecret
- GenerateDataKey
- GenerateDataKeyPair
- GenerateRandom

下表展示了 API 操作对认证请求的响应与其标准响应的差异。

Amazon KMS 操作	标准响应	对认证请求的响应
`Decrypt`	返回明文数据	返回证明文档中由公有密钥加密的明文数据
`DeriveSharedSecret`	返回原始共享密钥	返回证明文档中由公有密钥加密的原始共享密钥
`GenerateDataKey`	返回数据密钥的明文副本 （还会返回由 KMS 密钥加密的数据密钥副本）	返回证明文档中由公有密钥加密的数据密钥副本 （还会返回由 KMS 密钥加密的数据密钥副本）
`GenerateDataKeyPair`	返回私有密钥的明文副本 （还会返回公有密钥和由 KMS 密钥加密的私有密钥副本）	返回证明文档中由公有密钥加密的私有密钥副本 （还会返回公有密钥和由 KMS 密钥加密的私有密钥副本）

Amazon KMS 操作	标准响应	对认证请求的响应
GenerateRandom	返回一个随机字节字符串	返回证明文档中由公有密钥加密的随机字节字符串

Amazon KMS 支持策略条件键，您可以使用这些条件键根据认证文件的内容允许或拒绝对 Amazon KMS 密钥的认证操作。您还可以通过 Amazon CloudTrail 日志监控对 Amazon KMS 的认证请求。

了解更多

- 加密证明
- Amazon KMS 经过验证的平台的条件密钥
- 如何向 Amazon KMS 发出认证调用
- 监控认证请求

如何向 Amazon KMS 发出认证调用

要向 Amazon KMS 发出认证调用，请在请求中使用 Recipient 参数来提供已签名的认证文档以及用于认证文档中的公有密钥的加密算法。当请求中包含带有已签名证明文档的 Recipient 参数时，响应将包含一个具有由公有密钥加密的加密文字的 CiphertextForRecipient 字段。明文字段为空。

Recipient 参数必须指定一个来自 Amazon Nitro Enclaves 或 Amazon NitroTPM 的已签名认证文档。Amazon KMS 依赖该认证文档的数字签名来证明请求中的公有密钥来自有效的来源。您不能提供自己的证书来对证明文档进行数字签名。

Amazon Nitro Enclaves 开发工具包仅在 Nitro Enclave 内受支持，会自动将 Recipient 参数及其值添加到每个 Amazon KMS 请求中。

要在 Amazon SDK 中发出认证请求，必须指定 Recipient 参数及其值。该认证文档可使用 nitro-tpm-attest 实用程序从 NitroTPM 中检索，也可使用 NSM API 从 Nitro 安全模块（NSM）中检索。

Amazon KMS 支持策略条件键，您可以使用这些条件键根据认证文件的内容允许或拒绝对 Amazon KMS 密钥的认证操作。您还可以通过 Amazon CloudTrail 日志监控对 Amazon KMS 的认证请求。

有关 Recipient 参数和 AWS CiphertextForRecipient 响应字段的详细信息，请参阅《Amazon Key Management Service API 参考》、Amazon Nitro Enclaves SDK 或任何 Amazon SDK 中的 Decrypt、DeriveSharedSecret、GenerateDataKey、GenerateDataKeyPair 和 GenerateRandom 主题。有关为加密设置数据和数据密钥的信息，请参阅将加密证明与 Amazon KMS 结合使用。

监控认证请求

您可以使用 Amazon CloudTrail 日志来监控使用认证的 Decrypt、DeriveSharedSecret、GenerateDataKey、GenerateDataKeyPair 和 GenerateRandom 操作。在这些日志条目中，additionalEventData 字段中的 recipient 字段包含来自请求中认证文档的信息。仅当请求中的 Recipient 参数指定了某个已签名认证文档时，才会包含这些字段。

CloudTrail 日志中包含的具体信息取决于所使用的认证方法。

监控 Nitro Enclave 的请求

对于 Nitro Enclaves 认证，CloudTrail 日志将来自认证文档的模块 ID (attestationDocumentModuleId)、映像摘要 (attestationDocumentEnclaveImageDigest) 和平台配置寄存器 (PCR)。

模块 ID 是 Nitro Enclave 的 Enclave ID。映像摘要是 Enclave 映像的 SHA384 哈希值。您可以在密钥政策和 IAM policy 的条件中使用映像摘要和 PCR 值。有关 PCR 的信息，请参阅《Amazon Nitro Enclave User Guide》中的 Where to get an enclave's measurements。

本节显示了对 Amazon KMS 的每个受支持的 Nitro Enclave 请求的一个 CloudTrail 日志条目示例。

解密（适用于 Enclave）

以下示例显示了 Amazon Nitro enclave 的 Decrypt 操作的一个 Amazon CloudTrail 日志条目。


```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-07-27T22:58:24Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
```

```
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "additionalEventData": {
        "recipient": {
            "attestationDocumentModuleId": "i-123456789abcde123-enc123456789abcde12",
            "attestationDocumentEnclaveImageDigest": "<AttestationDocument.PCR0>",
            "attestationDocumentEnclavePCR1": "<AttestationDocument.PCR1>",
            "attestationDocumentEnclavePCR2": "<AttestationDocument.PCR2>",
            "attestationDocumentEnclavePCR3": "<AttestationDocument.PCR3>",
            "attestationDocumentEnclavePCR4": "<AttestationDocument.PCR4>",
            "attestationDocumentEnclavePCR8": "<AttestationDocument.PCR8>"
        }
    },
    "requestID": "b4a65126-30d5-4b28-98b9-9153da559963",
    "eventID": "e5a2f202-ba1a-467c-b4ba-f729d45ae521",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GenerateDataKey（适用于 Enclave）

以下示例显示了 Amazon Nitro Enclave 的 GenerateDataKey 操作的一个 Amazon CloudTrail 日志条目。

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
```

```
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:40Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "numberOfBytes": 32
    },
    "responseElements": null,
    "additionalEventData": {
        "recipient": {
            "attestationDocumentModuleId": "i-123456789abcde123-enc123456789abcde12",
            "attestationDocumentEnclaveImageDigest": "<AttestationDocument.PCR0>",
            "attestationDocumentEnclavePCR1": "<AttestationDocument.PCR1>",
            "attestationDocumentEnclavePCR2": "<AttestationDocument.PCR2>",
            "attestationDocumentEnclavePCR3": "<AttestationDocument.PCR3>",
            "attestationDocumentEnclavePCR4": "<AttestationDocument.PCR4>",
            "attestationDocumentEnclavePCR8": "<AttestationDocument.PCR8>"
        }
    },
    "requestID": "e0eb83e3-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "a9dea4f9-8395-46c0-942c-f509c02c2b71",
    "readOnly": true,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GenerateDataKeyPair（适用于 Enclave）

以下示例显示了 Amazon Nitro enclave 的 GenerateDataKeyPair 操作的一个 Amazon CloudTrail 日志条目。

```
{
```

```
    "eventVersion": "1.05",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2020-07-27T18:57:57Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKeyPair",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "keyPairSpec": "RSA_3072",
        "encryptionContext": {
            "Project": "Alpha"
        },
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "additionalEventData": {
        "recipient": {
            "attestationDocumentModuleId": "i-123456789abcde123-enc123456789abcde12",
            "attestationDocumentEnclaveImageDigest": "<AttestationDocument.PCR0>",
            "attestationDocumentEnclavePCR1": "<AttestationDocument.PCR1>",
            "attestationDocumentEnclavePCR2": "<AttestationDocument.PCR2>",
            "attestationDocumentEnclavePCR3": "<AttestationDocument.PCR3>",
            "attestationDocumentEnclavePCR4": "<AttestationDocument.PCR4>",
            "attestationDocumentEnclavePCR8": "<AttestationDocument.PCR8>"
        }
    },
    "requestID": "52fb127b-0fe5-42bb-8e5e-f560febde6b0",
    "eventID": "9b6bd6d2-529d-4890-a949-593b13800ad7",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
```

```
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GenerateRandom（适用于 Enclave）

以下示例显示了 Amazon Nitro Enclave 的 GenerateRandom 操作的一个 Amazon CloudTrail 日志条目。

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:37Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateRandom",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData": {
        "recipient": {
            "attestationDocumentModuleId": "i-123456789abcde123-enc123456789abcde12",
            "attestationDocumentEnclaveImageDigest": "<AttestationDocument.PCR0>",
            "attestationDocumentEnclavePCR1": "<AttestationDocument.PCR1>",
            "attestationDocumentEnclavePCR2": "<AttestationDocument.PCR2>",
            "attestationDocumentEnclavePCR3": "<AttestationDocument.PCR3>",
            "attestationDocumentEnclavePCR4": "<AttestationDocument.PCR4>",
            "attestationDocumentEnclavePCR8": "<AttestationDocument.PCR8>"
        }
    },
    "requestID": "df1e3de6-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "239cb9f7-ae05-4c94-9221-6ea30eef0442",
    "readOnly": true,
    "resources": [],
```

```
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

监控 NitroTPM 的请求

对于 NitroTPM 认证，CloudTrail 日志将来自认证文档的模块 ID (`attestationDocumentModuleId`) 和平台配置寄存器 (PCR)。

模块 ID 是使用 NitrotPM 的 EC2 实例 ID，带有 TPM 标识符。您可以在密钥策略和 IAM 策略的条件中使用 PCR 值。

本节展示了向 Amazon KMS 发出的每个受支持 NitroTPM 请求的一个 CloudTrail 日志条目示例。

Decrypt（适用于 NitroTPM）

以下示例展示了 NitroTPM 的 Decrypt 操作的一个 Amazon CloudTrail 日志条目。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2020-07-27T22:58:24Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "Decrypt",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "encryptionAlgorithm": "SYMMETRIC_DEFAULT",
        "keyId": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "additionalEventData": {
        "recipient": {
           "attestationDocumentModuleId": "i-123456789abcde123-tpm0000000000000000",
```

```
            "attestationDocumentNitroTPMPCR4": "<AttestationDocument.PCR4>",
            "attestationDocumentNitroTPMPCR7": "<AttestationDocument.PCR7>",
            "attestationDocumentNitroTPMPCR8": "<AttestationDocument.PCR8>",
            "attestationDocumentNitroTPMPCR9": "<AttestationDocument.PCR9>",
            "attestationDocumentNitroTPMPCR16": "<AttestationDocument.PCR16>",
            "attestationDocumentNitroTPMPCR23": "<AttestationDocument.PCR23>"
        }
    },
    "requestID": "b4a65126-30d5-4b28-98b9-9153da559963",
    "eventID": "e5a2f202-ba1a-467c-b4ba-f729d45ae521",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GenerateDataKey（适用于 NitroTPM）

以下示例展示了 NitroTPM 的 GenerateDataKey 操作的一个 Amazon CloudTrail 日志条目。

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:40Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKey",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
```

```
    "requestParameters": {
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab",
        "numberOfBytes": 32
    },
    "responseElements": null,
    "additionalEventData": {
        "recipient": {
           "attestationDocumentModuleId": "i-123456789abcde123-tpm0000000000000000",
           "attestationDocumentNitroTPMPCR4": "<AttestationDocument.PCR4>",
           "attestationDocumentNitroTPMPCR7": "<AttestationDocument.PCR7>",
           "attestationDocumentNitroTPMPCR8": "<AttestationDocument.PCR8>",
           "attestationDocumentNitroTPMPCR9": "<AttestationDocument.PCR9>",
           "attestationDocumentNitroTPMPCR16": "<AttestationDocument.PCR16>",
           "attestationDocumentNitroTPMPCR23": "<AttestationDocument.PCR23>"
        }
    },
    "requestID": "e0eb83e3-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "a9dea4f9-8395-46c0-942c-f509c02c2b71",
    "readOnly": true,
    "resources": [{
        "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab",
        "accountId": "111122223333"
    }],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GenerateDataKeyPair（适用于 NitroTPM）

以下示例展示了 NitroTPM 的 GenerateDataKeyPair 操作的一个 Amazon CloudTrail 日志条目。

```
{
    "eventVersion": "1.05",
    "userIdentity": {
            "type": "IAMUser",
            "principalId": "EX_PRINCIPAL_ID",
            "arn": "arn:aws:iam::111122223333:user/Alice",
            "accountId": "111122223333",
            "accessKeyId": "EXAMPLE_KEY_ID",
            "userName": "Alice"
    },
    "eventTime": "2020-07-27T18:57:57Z",
```

```
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateDataKeyPair",
    "awsRegion": "us-west-2",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": {
        "keyPairSpec": "RSA_3072",
        "encryptionContext": {
            "Project": "Alpha"
        },
        "keyId": "1234abcd-12ab-34cd-56ef-1234567890ab"
    },
    "responseElements": null,
    "additionalEventData": {
        "recipient": {
           "attestationDocumentModuleId": "i-123456789abcde123-tpm0000000000000000",
           "attestationDocumentNitroTPMPCR4": "<AttestationDocument.PCR4>",
           "attestationDocumentNitroTPMPCR7": "<AttestationDocument.PCR7>",
           "attestationDocumentNitroTPMPCR8": "<AttestationDocument.PCR8>",
           "attestationDocumentNitroTPMPCR9": "<AttestationDocument.PCR9>",
           "attestationDocumentNitroTPMPCR16": "<AttestationDocument.PCR16>",
           "attestationDocumentNitroTPMPCR23": "<AttestationDocument.PCR23>"
        }
    },
    "requestID": "52fb127b-0fe5-42bb-8e5e-f560febde6b0",
    "eventID": "9b6bd6d2-529d-4890-a949-593b13800ad7",
    "readOnly": true,
    "resources": [
        {
            "accountId": "111122223333",
            "type": "AWS::KMS::Key",
            "ARN": "arn:aws:kms:us-
west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab"
        }
    ],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

GenerateRandom（适用于 NitroTPM）

以下示例展示了 NitroTPM 的 GenerateRandom 操作的一个 Amazon CloudTrail 日志条目。

```
{
    "eventVersion": "1.02",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "EX_PRINCIPAL_ID",
        "arn": "arn:aws:iam::111122223333:user/Alice",
        "accountId": "111122223333",
        "accessKeyId": "EXAMPLE_KEY_ID",
        "userName": "Alice"
    },
    "eventTime": "2014-11-04T00:52:37Z",
    "eventSource": "kms.amazonaws.com",
    "eventName": "GenerateRandom",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "192.0.2.0",
    "userAgent": "Amazon Internal",
    "requestParameters": null,
    "responseElements": null,
    "additionalEventData": {
        "recipient": {
           "attestationDocumentModuleId": "i-123456789abcde123-tpm0000000000000000",
           "attestationDocumentNitroTPMPCR4": "<AttestationDocument.PCR4>",
           "attestationDocumentNitroTPMPCR7": "<AttestationDocument.PCR7>",
           "attestationDocumentNitroTPMPCR8": "<AttestationDocument.PCR8>",
           "attestationDocumentNitroTPMPCR9": "<AttestationDocument.PCR9>",
           "attestationDocumentNitroTPMPCR16": "<AttestationDocument.PCR16>",
           "attestationDocumentNitroTPMPCR23": "<AttestationDocument.PCR23>"
        }
    },
    "requestID": "df1e3de6-63bc-11e4-bc2b-4198b6150d5c",
    "eventID": "239cb9f7-ae05-4c94-9221-6ea30eef0442",
    "readOnly": true,
    "resources": [],
    "eventType": "AwsApiCall",
    "recipientAccountId": "111122223333"
}
```

对 Amazon 服务使用 Amazon KMS 加密

使用 Amazon Key Management Service，您可以提供加密密钥以保护其他 Amazon 服务中的数据。对 Amazon 服务使用 Amazon KMS 加密是指将 Amazon KMS 与其他 Amazon 服务集成，从而对静态数据或传输中数据进行加密和解密的过程。开发人员、系统管理员和安全专业人员可能会对本主题感兴趣，阅读本主题，以便保护通过 Amazon 服务存储或传输的敏感数据，满足监管合规性要求或实施加密最佳实践。常见用例包括加密 Amazon EBS 卷、Amazon S3 存储桶和 Amazon RDS 数据库。以下各节将介绍配置和管理特定 Amazon 服务的 Amazon KMS 加密密钥的步骤，确保整个 Amazon 环境中的数据机密性和完整性。有关与 Amazon KMS 集成的 Amazon 服务的完整列表，请参阅 Amazon 服务集成。

以下主题详细讨论了特定服务如何使用 Amazon KMS（包括其支持的 KMS 密钥）、这些服务如何管理数据密钥、所需的权限、以及如何跟踪您账户中每项服务使用 KMS 密钥的情况。

> **Important**
>
> 与 Amazon KMS 集成的 Amazon 服务仅使用对称加密 KMS 密钥加密您的数据。这些服务不支持使用非对称 KMS 密钥进行加密。要获取确定 KMS 密钥是对称还是非对称的帮助，请参阅识别不同的密钥类型。

主题

- 亚马逊 Elastic Block Store (Amazon EBS) 的使用方式 Amazon KMS
- Amazon EMR 如何使用 Amazon KMS
- Amazon Redshift 如何使用 Amazon KMS

亚马逊 Elastic Block Store (Amazon EBS) 的使用方式 Amazon KMS

本主题详细讨论了 Amazon Elastic Block Store (Amazon EBS) 如何 Amazon KMS 使用加密卷和快照。有关加密 Amazon EBS 卷的基本说明，请参阅 Amazon EBS 加密。

主题

- Amazon EBS 加密
- 使用 KMS 密钥和数据密钥

- Amazon EBS 加密上下文
- 检测 Amazon EBS 故障
- Amazon CloudFormation 用于创建加密的 Amazon EBS 卷

Amazon EBS 加密

当您将加密的 Amazon EBS 卷连接到支持的亚马逊弹性计算云 (Amazon EC2) 实例类型时，该卷上的静态数据、磁盘 I/O 以及从该卷创建的快照都会被加密。加密发生在托管 Amazon EC2 实例的服务器上。

所有 Amazon EBS 卷类型都支持此功能。您可以像访问其他卷一样访问加密卷；加密和解密是透明处理的，无需您、您的 EC2 实例或应用程序采取任何其他操作。加密 卷的快照会自动加密，通过加密快照创建的卷也会自动加密。

EBS 卷的加密状态在您创建该卷时就已经确定了。您不能更改现有卷的加密状态。但是，您可以在加密卷和未加密卷之间迁移数据，并在复制快照时应用新的加密状态。

默认情况下，Amazon EBS 支持可选加密。您可以在您和地区的所有新 EBS 卷和快照副本上自动启用加密。 Amazon Web Services 账户 此配置设置不会影响现有卷或快照。有关详细信息，请参阅《Amazon EBS 用户指南》中的 Amazon EBS encryption。

使用 KMS 密钥和数据密钥

当您创建一个加密的 Amazon EBS 卷时，您可以指定 Amazon KMS key。默认情况下，Amazon EBS 将在您的账户 (aws/ebs) 中将 Amazon 托管式密钥 Amazon EBS。不过，您可以指定自己创建和管理的客户托管的密钥。

要使用客户托管的密钥，您必须向 Amazon EBS 授予代表您使用 KMS 密钥的权限。有关更多信息，请参阅《Amazon EBS 用户指南》中的 How Amazon EBS encryption works。

> **Important**
>
> Amazon EBS 仅支持对称 KMS 密钥。不能使用非对称 KMS 密钥来加密 Amazon EBS 卷。要获取确定 KMS 密钥是对称还是非对称的帮助，请参阅 识别不同的密钥类型。

对于每个卷，Amazon EBS 会要求 Amazon KMS 生成一个使用您指定的 KMS 密钥加密的唯一数据密钥。Amazon EBS 使用该卷存储加密数据密钥。然后，当您将卷连接到 Amazon EC2 实例

时，Amazon EBS 会调 Amazon KMS 用解密数据密钥。Amazon EBS 使用虚拟机管理程序内存中的纯文本数据密钥将所有磁盘加密到该卷。 I/O 有关详情，请参阅《亚马逊 EC2 用户指南》或《亚马逊用户指南》中的 EBS 加密的工作原理。 EC2

Amazon EBS 加密上下文

在对的请求GenerateDataKeyWithoutPlaintext和解密请求中， Amazon KMS Amazon EBS 使用带有名称/值对的加密上下文，用于标识请求中的卷或快照。加密上下文中的名称不会发生变化。

加密上下文 是一组包含任意非机密数据的键值对。当您在加密数据的请求中包含加密上下文时，会以加密 Amazon KMS 方式将加密上下文绑定到加密数据。要解密数据，您必须传入相同的加密上下文。

对于所有卷以及通过 Amazon EBS CreateSnapshot操作创建的加密快照，Amazon EBS 使用卷 ID 作为加密上下文值。在 CloudTrail 日志条目的 `requestParameters` 字段中，加密上下文类似于以下内容：

```
"encryptionContext": {
  "aws:ebs:id": "vol-0cfb133e847d28be9"
}
```

对于通过亚马逊 EC2 CopySnapshot操作创建的加密快照，Amazon EBS 使用快照 ID 作为加密上下文值。在 CloudTrail 日志条目的 `requestParameters` 字段中，加密上下文类似于以下内容：

```
"encryptionContext": {
  "aws:ebs:id": "snap-069a655b568de654f"
}
```

检测 Amazon EBS 故障

要创建加密的 EBS 卷或将该卷附加到 EC2 实例，Amazon EBS 和亚马逊 EC2 基础设施必须能够使用您为 EBS 卷加密指定的 KMS 密钥。当 KMS 密钥不可用时—例如，当其密钥状态处于 `Enabled` 时—卷创建或卷附加操作将失败。

在这种情况下，Amazon EBS 会向亚马逊发送一个事件 EventBridge （以前称为“ CloudWatch 事件”），以通知您有关失败的信息。在中 EventBridge，您可以建立触发自动操作以响应这些事件的规则。有关更多信息，请参阅《亚马逊 EBS 用户指南》中的 Amazon Ebs CloudWatch 活动，尤其是以下部分：

- 附加或重新附加卷时加密密钥无效

- 创建卷时加密密钥无效

要修复这些故障，请确保您为 EBS 卷加密指定的 KMS 密钥处于启用状态。为此，请先查看 KMS 密钥以确定其当前密钥状态（中的状态列 Amazon Web Services 管理控制台）。然后，请参阅以下任一链接中的信息：

- 如果 KMS 密钥的密钥状态为已禁用，则启用它。
- 如果 KMS 密钥的密钥状态为待导入，则导入密钥材料。
- 如果 KMS 密钥的密钥状态为待删除，则取消密钥删除。

Amazon CloudFormation 用于创建加密的 Amazon EBS 卷

您可以使用 Amazon CloudFormation 创建加密的 Amazon EBS 卷。有关更多信息，请参阅《Amazon CloudFormation 用户指南》中的 AWS::EC2::Volume。

Amazon EMR 如何使用 Amazon KMS

当您使用 Amazon EMR 集群时，您可以将集群配置为静态加密数据，然后将其保存到持久性存储位置。您可以在 EMR 文件系统 (EMRFS) 上、在群集节点的存储卷上，或同时在这两者上对静态数据进行加密。要加密静态数据，您可以使用 Amazon KMS key。以下主题介绍 Amazon EMR 集群如何使用 KMS 密钥来加密静态数据。

> **Important**
>
> Amazon EMR 仅支持对称 KMS 密钥。不能使用非对称 KMS 密钥来加密 Amazon EMR 集群中的静态数据。要获取确定 KMS 密钥是对称还是非对称的帮助，请参阅 识别不同的密钥类型。

Amazon EMR 集群也可以加密传输中的数据，这意味着集群会先加密数据，然后将其通过网络发送。您不能使用 KMS 密钥加密传输中的数据。有关更多信息，请参阅 Amazon EMR 管理指南中的传输中的数据加密。

有关 Amazon EMR 中所有可用加密选项的更多信息，请参阅 Amazon EMR 管理指南中的加密选项。

主题

- 在 EMR 文件系统 (EMRFS) 上加密数据

- 在集群节点的存储卷上加密数据
- 加密上下文

在 EMR 文件系统 (EMRFS) 上加密数据

Amazon EMR 集群使用两个分布式文件系统：

- Hadoop Distributed File System（HDFS）。HDFS 加密不会使用 Amazon KMS 中的 KMS 密钥。
- EMR 文件系统 (EMRFS)。EMRFS 是一种 HDFS 实施，使 Amazon EMR 集群能够在 Amazon Simple Storage Service (Amazon S3) 中存储数据。EMRFS 支持四种加密选项，其中两种会使用 Amazon KMS 中的 KMS 密钥。有关全部四种 EMRFS 加密选项的更多信息，请参阅 Amazon EMR 管理指南中的加密选项。

使用 KMS 密钥的两个 EMRFS 加密选项使用 Amazon S3 提供的以下加密功能：

- 使用具有 Amazon Key Management Service 的服务器端加密 (SSE-KMS) 保护数据。Amazon EMR 集群会将数据发送到 Amazon S3。Amazon S3 使用 KMS 密钥加密数据，然后将其保存到 S3 存储桶。有关其工作方式的更多信息，请参阅使用 SSE-KMS 在 EMRFS 上加密数据的过程。
- 使用客户端加密保护数据 (CSE-KMS)。Amazon EMR 中的数据通过 Amazon KMS key 进行加密，然后发送到 Amazon S3 进行存储。有关其工作方式的更多信息，请参阅使用 CSE-KMS 在 EMRFS 上加密数据的过程。

当您将 Amazon EMR 集群配置为使用 KMS 密钥在 EMRFS 上加密数据时，请选择您希望 Amazon S3 或 Amazon EMR 集群使用的 KMS 密钥。借助 SSE-KMS，您可以为 Amazon S3 选择 Amazon 托管式密钥（具有别名 aws/s3）或您创建的对称客户托管密钥。使用客户端加密时，必须选择您创建的对称客户托管式密钥。当您选择客户托管密钥时，您必须确保 Amazon EMR 集群有权使用该 KMS 密钥。有关更多信息，请参阅 Amazon EMR 管理指南中的将 Amazon KMS keys 用于加密。

对于服务器端和客户端加密这两者而言，您选择的 KMS 密钥就是信封加密工作流程中的根密钥。数据使用唯一的数据密钥加密，该密钥通过 Amazon KMS 中的 KMS 密钥进行加密。已加密的数据及其数据密钥的加密副本将作为单个加密对象共同存储在 S3 存储桶中。有关其工作方式的更多信息，请参阅以下主题。

主题

- 使用 SSE-KMS 在 EMRFS 上加密数据的过程
- 使用 CSE-KMS 在 EMRFS 上加密数据的过程

使用 SSE-KMS 在 EMRFS 上加密数据的过程

当您将 Amazon EMR 集群配置为使用 SSE-KMS 时，加密过程的工作方式如下所示：

1. 集群将数据发送到 Amazon S3，以存储在 S3 存储桶中。
2. Amazon S3 向 Amazon KMS 发送 GenerateDataKey 请求，同时指定您在将集群配置为使用 SSE-KMS 时所选的 KMS 密钥的密钥 ID。该请求包含加密上下文；有关更多信息，请参阅加密上下文。
3. Amazon KMS 生成一个唯一数据加密密钥（数据密钥），然后将此数据密钥的两个副本发送到 Amazon S3。一个副本未加密（明文），另一个副本使用 KMS 密钥加密。
4. Amazon S3 使用明文数据密钥加密它在步骤 1 中收到的数据，并在使用后尽快从内存中删除该明文数据密钥。
5. Amazon S3 将已加密的数据及数据密钥的加密副本作为单个加密对象共同存储在 S3 存储桶中。

解密过程的工作方式如下所示：

1. 群集从 S3 存储桶请求加密的数据对象。
2. Amazon S3 从 S3 对象提取加密的数据密钥，然后使用 Decrypt 请求将已加密的数据密钥发送给 Amazon KMS。该请求包括一个加密上下文。
3. Amazon KMS 借助加密数据密钥时使用的同一 KMS 密钥解密该加密数据密钥，然后将已解密的（明文）数据密钥发送到 Amazon S3。
4. Amazon S3 使用明文数据密钥解密已加密的数据，并在使用后尽快从内存中删除该明文数据密钥。
5. Amazon S3 将解密数据发送给集群。

使用 CSE-KMS 在 EMRFS 上加密数据的过程

当您将 Amazon EMR 集群配置为使用 CSE-KMS 时，加密过程的工作方式如下所示：

1. 在准备好将数据存储在 Amazon S3 中后，集群向 Amazon KMS 发送 GenerateDataKey 请求，同时指定您在将集群配置为使用 CSE-KMS 时所选的 KMS 密钥的密钥 ID。该请求包含加密上下文；有关更多信息，请参阅加密上下文。
2. Amazon KMS 生成一个唯一数据加密密钥（数据密钥），然后将此数据密钥的两个副本发送到群集。一个副本未加密（明文），另一个副本使用 KMS 密钥加密。
3. 群集使用明文数据密钥加密数据，并在使用后尽快从内存中删除该明文数据密钥。
4. 群集将已加密的数据及数据密钥的加密副本组合为单个加密对象。

5. 集群将加密的对象发送给 Amazon S3 进行存储。

解密过程的工作方式如下所示：

1. 群集从 S3 存储桶请求加密的数据对象。
2. Amazon S3 将加密的对象发送给集群。
3. 群集从加密的对象提取加密的数据密钥，然后使用 Amazon KMSDecrypt 请求将已加密的数据密钥发送给 。该请求包括加密上下文。
4. Amazon KMS 借助加密数据密钥时使用的同一 KMS 密钥解密该加密数据密钥，然后将已解密的（明文）数据密钥发送到集群。
5. 群集使用明文数据密钥解密已加密的数据，并在使用后尽快从内存中删除该明文数据密钥。

在集群节点的存储卷上加密数据

Amazon EMR 集群是 Amazon Elastic Compute Cloud（Amazon EC2）实例的集合。群集中的每个实例称作群集节点或节点。每个节点都可以有两类存储卷：实例存储卷和 Amazon Elastic Block Store (Amazon EBS) 卷。您可以将群集配置为使用 Linux Unified Key Setup (LUKS) 来加密节点上的两类存储卷 (但不包括每个节点的启动卷)。这称为本地磁盘加密。

在为集群启用本地磁盘加密后，您可以选择使用 Amazon KMS 中的 KMS 密钥加密 LUKS 密钥。您必须选择您创建的客户托管密钥；不能使用 Amazon 托管式密钥。如果您选择客户托管密钥，您必须确保 Amazon EMR 集群有权使用该 KMS 密钥。有关更多信息，请参阅 Amazon EMR 管理指南中的将 Amazon KMS keys 用于加密。

当您使用 KMS 密钥启用本地磁盘加密时，加密过程的工作方式如下所示：

1. 在每个集群节点启动后，它将向 Amazon KMS 发送 GenerateDataKey 请求，同时指定您在为集群启用本地磁盘加密时所选的 KMS 密钥的密钥 ID。
2. Amazon KMS 生成一个唯一数据加密密钥（数据密钥），然后将此数据密钥的两个副本发送到节点。一个副本未加密（明文），另一个副本使用 KMS 密钥加密。
3. 该节点将明文数据密钥的 base64 编码版本作为保护 LUKS 密钥的密码。节点会将加密的数据密钥副本保存在启动卷上。
4. 如果节点重启，重启节点会使用 Amazon KMSDecrypt 请求将已加密的数据密钥发送到 。
5. Amazon KMS 借助加密数据密钥时使用的同一 KMS 密钥解密该加密数据密钥，然后将已解密的（明文）数据密钥发送到节点。
6. 该节点将明文数据密钥的 base64 编码版本作为解锁 LUKS 密钥的密码。

加密上下文

与 Amazon KMS 集成的每项 Amazon 服务在使用 Amazon KMS 生成数据密钥或者加密或解密数据时都可以指定加密上下文。加密上下文是 Amazon KMS 检查数据完整性时使用的额外的身份验证信息。当服务为加密操作指定加密上下文时，它还必须为对应的解密操作指定同一加密上下文，否则解密会失败。加密上下文也将写入 Amazon CloudTrail 日志文件中，这可以帮助您了解为什么要使用指定的 KMS 密钥。

以下部分介绍每个使用 KMS 密钥的 Amazon EMR 加密场景中使用的加密上下文。

使用 SSE-KMS 的 EMRFS 加密的加密上下文

借助 SSE-KMS，Amazon EMR 集群将数据发送到 Amazon S3，然后，Amazon S3 使用 KMS 密钥加密数据，然后将其存储到 S3 存储桶中。在这种情况下，Amazon S3 使用它发送给 Amazon KMS 的每个 GenerateDataKey 和 Decrypt 请求将 S3 对象的 Amazon Resource Name (ARN) 作为加密上下文。以下示例显示了 Amazon S3 使用的加密上下文的 JSON 表示形式。

```
{ "aws:s3:arn" : "arn:aws:s3:::S3_bucket_name/S3_object_key" }
```

使用 CSE-KMS 的 EMRFS 加密的加密上下文

借助 CSE-KMS，Amazon EMR 集群使用 KMS 密钥加密数据，然后将其发送到 Amazon S3 进行存储。在这种情况下，集群使用它发送给 Amazon KMS 的每个 GenerateDataKey 和 Decrypt 请求将 KMS 密钥的 Amazon Resource Name (ARN) 作为加密上下文。以下示例显示了群集使用的加密上下文的 JSON 表示形式。

```
{ "kms_cmk_id" : "arn:aws:kms:us-
east-2:111122223333:key/0987ab65-43cd-21ef-09ab-87654321cdef" }
```

使用 LUKS 的本地磁盘加密的加密上下文

当 Amazon EMR 集群使用 LUKS 进行本地磁盘加密时，集群节点不使用它们发送给 Amazon KMS 的 GenerateDataKey 和 Decrypt 请求指定加密上下文。

Amazon Redshift 如何使用 Amazon KMS

本主题讨论 Amazon Redshift 如何使用 Amazon KMS 加密数据。

主题

Amazon Redshift 加密

Amazon Redshift 数据仓库是一个由称作节点的各种计算资源构成的集合，这些节点已整理到名为集群的组中。每个集群运行一个 Amazon Redshift 引擎并包含一个或多个数据库。

Amazon Redshift 使用基于密钥的四层架构来进行加密。此架构包括数据加密密钥、数据库密钥、群集密钥和根密钥。您可以使用 Amazon KMS key 作为根密钥。

数据加密密钥对群集中的数据块进行加密。每个数据块分配有一个随机生成的 AES-256 密钥。这些密钥使用群集的数据库密钥进行加密。

数据库密钥对群集中的数据加密密钥进行加密。数据库密钥是随机生成的 AES-256 密钥。它存储在独立于 Amazon Redshift 集群的网络中的磁盘上，并跨安全通道传递到集群中。

群集密钥对 Amazon Redshift 群集的数据库密钥进行加密。您可以使用 Amazon KMS、Amazon CloudHSM 或外部硬件安全模块 (HSM) 来管理群集密钥。有关更多详细信息，请参阅 Amazon Redshift 数据库加密文档。

您可以通过选中 Amazon Redshift 控制台中相应的框请求加密。您可以从加密框下方显示的列表中选择，指定一个客户托管密钥。如果您不指定客户托管的密钥，则 Amazon Redshift 将 Amazon 托管式密钥 用于您账户下的 Amazon Redshift。

> **Important**
>
> Amazon Redshift 仅支持对称加密 KMS 密钥。您不能在 Amazon Redshift 加密工作流中使用非对称 KMS 密钥。要获取确定 KMS 密钥是对称还是非对称的帮助，请参阅 识别不同的密钥类型。

加密上下文

与 Amazon KMS 集成的每项服务在请求数据密钥、加密和解密时会指定一个加密上下文。加密上下文是 Amazon KMS 用于检查数据完整性而使用的额外的身份验证数据 (AAD)。也就是说，在为加密操作指定加密上下文时，该服务还要为解密操作指定同一加密上下文，否则解密会失败。Amazon Redshift 使用加密上下文的集群 ID 和创建时间。在 CloudTrail 日志文件的 `requestParameters` 字段中，加密上下文与以下示例类似。

```
"encryptionContext": {
    "aws:redshift:arn": "arn:aws:redshift:region:account_ID:cluster:cluster_name",
    "aws:redshift:createtime": "20150206T1832Z"
},
```

您可以使用 Amazon KMS key（KMS 密钥）在 CloudTrail 日志中搜索集群名称，以了解具体执行了哪些操作。这些操作包括集群加密、集群解密以及生成数据密钥。

Amazon KMS 参考

以下引用材料提供了有关使用和管理 KMS 密钥的有用信息。

- 密钥类型引用。列出支持每个 Amazon KMS API 操作的 KMS 密钥类型。

  查找：我能否启用和禁用 RSA 签名 KMS 密钥？
- 密钥状态表。显示 KMS 密钥的密钥状态如何影响其在 Amazon KMS API 操作中的使用。

  查找：我能否更改待删除的 KMS 密钥的别名？
- Amazon KMS API 权限引用。提供有关各个 Amazon KMS API 操作所需权限的信息。

  查找：我能否以不同 Amazon 账户的密钥运行 GetKeyPolicy？ 我能否在 IAM policy 中允许 `kms:Decrypt` 权限？
  - ViaService 引用。列出支持 `kms:ViaService` 条件键的 Amazon 服务。

    查找：我能否仅在其来自 Amazon ElastiCache 时，使用 `kms:ViaService` 条件键允许权限？ Amazon Neptune 怎么样？
- Amazon KMS 定价。列出并解释 KMS 密钥的价格。

  查找：使用我的非对称密钥如何收费？
- Amazon KMS 请求配额。列出每个账户和区域中 Amazon KMS API 请求的每秒配额。

  查找：我每秒可以运行多少 Decrypt 请求？ 在自定义密钥存储中的 KMS 密钥上，我可以运行多少个 Decrypt 请求？
- Amazon KMS 资源配额。列出 Amazon KMS 资源上的配额。

  查找：我可以在账户的各个区域拥有多少 KMS 密钥？ 在每个 KMS 密钥上我可以拥有多少个别名？
- 与 Amazon 集成的 Amazon KMS 服务。列出使用 KMS 密钥来保护其所创建、存储和管理的资源的 Amazon 服务。

  查找：Amazon Connect 是否使用 KMS 密钥来保护我的 Connect 资源？

Amazon KMS 密钥的密钥状态

Amazon KMS key 始终具有密钥状态。KMS 密钥及其环境上的操作可能会改变密钥状态。密钥状态的改变可能是暂时的，也可能持续到其他操作更改其密钥状态为止。这些操作可以异步完成，也可以通过 API 调用完成。

本节中的表格显示了密钥状态如何影响对 Amazon KMS API 操作的调用。由于其密钥状态，对 KMS 密钥执行的操作预计会成功 (#)，失败 (X)，或者仅在某些条件下成功 (?)。对于已导入密钥材料的 KMS 密钥，结果通常不同。

此表仅包括使用现有 KMS 密钥的 API 操作。其他操作，例如 CreateKey 和 ListKeys，则会省略。

主题

密钥状态和 KMS 密钥类型

KMS 密钥的类型决定了它可以具有的密钥状态。

- 所有 KMS 密钥都可以处于 `Enabled`、`Disabled` 和 `PendingDeletion` 状态。
- 大多数 KMS 密钥都在 `Enabled` 状态下创建。带导入的密钥材料的密钥在 `PendingImport` 状态下创建。
- `PendingImport` 状态仅适用于具有导入的密钥材料的 KMS 密钥。当导入的密钥中有任何密钥材料被删除或过期时，其状态将从 `Enabled` 变为 `PendingImport`。
- `Unavailable` 状态仅适用于自定义密钥存储中的 KMS 密钥。当自定义密钥存储有意与其 Amazon CloudHSM 集群断开连接时，Amazon CloudHSM 密钥存储中的 KMS 密钥为 `Unavailable`。当自定义密钥存储有意与其外部密钥存储代理断开连接时，外部密钥存储中的 KMS 密钥为 `Unavailable`。您可查看和管理不可用的 KMS 密钥，但无法在加密操作中使用它们。

  自定义密钥中的 KMS 密钥的密钥状态不受其备用密钥的更改的影响。Amazon CloudHSM 密钥存储中的 KMS 密钥不受 Amazon CloudHSM 集群中其关联密钥材料的更改的影响。外部密钥存储中的 KMS 密钥不受外部密钥管理器中其外部密钥的更改的影响。如果禁用或删除备用密钥，KMS 密钥状态不会改变，但使用 KMS 密钥的加密操作会失败。
- `Creating`、`Updating` 和 `PendingReplicaDeletion` 密钥状态仅适用于多区域密钥。
  - 多区域副本密钥在创建时处于临时 `Creating` 密钥状态。当 ReplicateKey 操作完成时，此过程可能仍在进行中。复制过程完成后，副本密钥处于 `Enabled` 或 `PendingImport` 状态。

- 多区域密钥在主区域正在更新时处于临时的 `Updating` 密钥状态。当 UpdatePrimaryRegion 操作完成时，此过程可能仍在进行中。更新过程完成后，主密钥和副本密钥将恢复 `Enabled` 密钥状态。
- 当您计划删除具有副本密钥的多区域主键时，主键处于 `PendingReplicaDeletion` 状态，直到其所有的副本密钥都被删除。然后，它的密钥状态更改为 `PendingDeletion`。有关更多信息，请参阅 Deleting multi-Region keys。

密钥状态表

下表显示 KMS 密钥的密钥状态如何影响 Amazon KMS 操作。

编号脚注的说明 ([n]) 处于本主题的末尾。

> **Note**
>
> 您可能需要水平或垂直滚动才能查看此表中的所有数据。

API	已启用	已禁用	待删除 待删除副本	待导入	不可用	创建	Updating
CancelKeyDeletion	✗ [4]	✗ [4]	✓	✗ [4]	✗ [4]、[13]	✗ [4]	✗ [4]
CreateAlias	✓	✓	✗ [3]	✓	✓	✓	✓
CreateGrant	✓	✗ [1]	✗ [2] 或 [3]	✗ [5]	✓	✗ [14]	✓
Decrypt	✓	✗	✗	✗	✗	✗	✓

API	已启用	已禁用	待删除 待删除副本	待导入	不可用	创建	Updating
		[1]	[2] 或 [3]	[5]	[11]	[14]	
DeleteAlias	✓	✓	✓	✓	✓	✓	✓
DeleteImportedKeyMaterial	✓ [9]	✓ [9]	✓ [9]	✓	不适用	⊗ [14]	⊗ [15]
DeriveSharedSecret	✓	⊗ [1]	⊗ [2] 或 [3]	⊗ [5]	不适用	⊗ [14]	✓
DescribeKey	✓	✓	✓	✓	✓	✓	✓
DisableKey	✓	✓	⊗ [3]	⊗ [5]	✓ [12]	⊗ [14]	⊗ [15]
DisableKeyRotation	? [7]	⊗ [1] 或 [7]	⊗ [3] 或 [7]	⊗ [6]	⊗ [7]	⊗ [14]	? [7]
EnableKey	✓	✓	⊗ [3]	⊗ [5]	✓ [12]	⊗ [14]	⊗ [15]
EnableKeyRotation	? [7]	⊗ [1] 或 [7]	⊗ [3] 或 [7]	⊗ [6]	⊗ [7]	⊗ [14]	? [7]

API	已启用	已禁用	待删除 待删除副本	待导入	不可用	创建	Updating
Encrypt	✓	⊗ [1]	⊗ [2] 或 [3]	⊗ [5]	⊗ [11]	⊗ [14]	✓
GenerateDataKey	✓	⊗ [1]	⊗ [2] 或 [3]	⊗ [5]	⊗ [11]	⊗ [14]	✓
GenerateDataKeyPair	✓	⊗ [1]	⊗ [2] 或 [3]	⊗ [5]	⊗ [7]	⊗ [14]	✓
GenerateDataKeyPairWithoutPlaintext	✓	⊗ [1]	⊗ [2] 或 [3]	⊗ [5]	⊗ [7]	⊗ [14]	✓
GenerateDataKeyWithoutPlaintext	✓	⊗ [1]	⊗ [2] 或 [3]	⊗ [5]	⊗ [11]	⊗ [14]	✓
GenerateMac	✓	⊗ [1]	⊗ [2] 或 [3]	⊗ [5]	不适用	⊗ [14]	✓
GetKeyPolicy	✓	✓	✓	✓	✓	✓	✓

API	已启用	已禁用	待删除 待删除副本	待导入	不可用	创建	Updating
GetKeyRotationStatus	❓ [7]	❓ [7]	❓ [7]	🚫 [6]	🚫 [7]	❓ [7]	❓ [7]
GetParametersForImport	❓ [9]	❓ [9]	🚫 [8] 或 [9]	✓	🚫 [9]	🚫 [14]	🚫 [15]
GetPublicKey	✓	✓	🚫 [2] 或 [3]	✓	不适用	🚫 [14]	✓
ImportKeyMaterial	❓ [9]	❓ [9]	🚫 [9]	✓	🚫 [9]	🚫 [14]	✓
ListAliases	✓	✓	✓	✓	✓	✓	✓
ListGrants	✓	✓	✓	✓	✓	✓	✓
ListKeyPolicies	✓	✓	✓	✓	✓	✓	✓
ListKeyRotations	❓ [7]	❓ [7]	❓ [7]	🚫 [6]	🚫 [7]	❓ [7]	❓ [7]
ListResourceTags	✓	✓	✓	✓	✓	✓	✓

API	已启用	已禁用	待删除 待删除副本	待导入	不可用	创建	Updating
PutKeyPolicy	✓	✓	✓	✓	✓	✓	✓
ReEncrypt	✓	⊗ [1]	⊗ [2] 或 [3]	⊗ [5]	⊗ [11]	⊗ [14]	✓
ReplicateKey	✓	⊗ [1]	⊗ [2] 或 [3]	⊗ [5]	不适用	⊗ [14]	⊗ [15]
RetireGrant	✓	✓	✓	✓	✓	✓	✓
RevokeGrant	✓	✓	✓	✓	✓	✓	✓
RotateKeyOnDemand	? [7]	⊗ [1] 或 [7]	⊗ [3] 或 [7]	⊗ [5]	⊗ [7]	⊗ [14]	? [7]
ScheduleKeyDeletion	✓	✓	⊗ [3]	✓	✓	✓	⊗ [15]
Sign	✓	⊗ [1]	⊗ [2] 或 [3]	⊗ [5]	不适用	⊗ [14]	✓

API	已启用	已禁用	待删除 待删除副本	待导入	不可用	创建	Updating
TagResource	✓	✓	⊗ [3]	✓	✓	✓	✓
UntagResource	✓	✓	⊗ [3]	✓	✓	✓	✓
UpdateAlias	✓	✓	? [10]	✓	✓	✓	✓
UpdateKeyDescription	✓	✓	⊗ [3]	✓	✓	✓	✓
UpdatePrimaryRegion	✓	⊗ [1]	⊗ [2] 或 [3]	⊗ [5]	不适用	⊗ [14]	✓
验证	✓	⊗ [1]	⊗ [2] 或 [3]	⊗ [5]	不适用	⊗ [14]	✓
VerifyMac	✓	⊗ [1]	⊗ [2] 或 [3]	⊗ [5]	不适用	⊗ [14]	✓

表详细信息

- [1] DisabledException: *<key ARN>* is disabled.
- [2] DisabledException: *<key ARN>* is pending deletion (or pending replica deletion).
- [3] KMSInvalidStateException: *<key ARN>* is pending deletion (or pending replica deletion).
- [4] KMSInvalidStateException: *<key ARN>* is not pending deletion (or pending replica deletion).
- [5] KMSInvalidStateException: *<key ARN>* is pending import because no key material has ever been imported or one of the imported key materials is deleted or expired.
- [6] UnsupportedOperationException: *<key ARN>* origin is EXTERNAL which is not valid for this operation.
- [7] 如果 KMS 密钥位于自定义密钥存储中：UnsupportedOperationException。
- [8] 如果 KMS 密钥已导入密钥材料：KMSInvalidStateException
- [9] 如果 KMS 密钥无法导入密钥材料：UnsupportedOperationException。
- [10] 如果源 KMS 密钥正等待删除，则该命令将成功。如果目标 KMS 密钥正等待删除，则该命令将失败，并显示以下错误消息：KMSInvalidStateException : *<key ARN>* is pending deletion.
- [11] KMSInvalidStateException: *<key ARN>* is unavailable. 您无法在不可用的 KMS 密钥上执行此操作。
- [12] 操作成功，但 KMS 密钥的密钥状态未更改，直到它变得可用。
- [13] 虽然自定义密钥存储中的 KMS 密钥处于等待删除状态，但其密钥状态保持 PendingDeletion 不变，即使 KMS 密钥变得不可用也是如此。这允许您在等待期内随时取消删除 KMS 密钥。
- [14] KMSInvalidStateException: *<key ARN>* is creating. Amazon KMS 在复制多区域密钥 (ReplicateKey) 时引发此异常。
- [15] KMSInvalidStateException: *<key ARN>* is updating. Amazon KMS 在更新多区域密钥的主区域 (UpdatePrimaryRegion) 时引发此异常。

密钥类型引用

Amazon KMS 支持不同类型的 KMS 密钥的不同功能。例如，只能使用对称加密 KMS 密钥来生成对称数据密钥和非对称数据密钥对。此外，仅对称加密 KMS 密钥支持导入密钥材料和自动密钥轮换，并且在自定义密钥存储中只能创建对称加密 KMS 密钥。

此参考包括两个表。

- 密钥类型表列出了对称加密 KMS 密钥、非对称 KMS 密钥和 HMAC KMS 密钥有效的 Amazon KMS 操作。
- 特殊功能表列出了对多区域 KMS 密钥、包含导入的密钥材料的 KMS 密钥，以及自定义密钥存储中的 KMS 密钥有效的 Amazon KMS 操作。

密钥类型表

您可能需要水平或垂直滚动才能查看此表中的所有数据。

Amazon KMS API 操作	对称加密 KMS 密钥	HMAC KMS 密钥	非对称 KMS 密钥 (ENCRYP DECRYP	非对称 KMS 密钥 (SIGN_VERIFY)	非对称 KMS 密钥（KEY_AG REEMENT）
CancelKeyDeletion	支持	是	是	是	是
CreateAlias	是	是	是	是	是
CreateGrant	是	是	是	是	是
CreateKey	是	是	是	是	是
Decrypt	是	否	是	否	否
DeleteAlias	是	是	是	是	是
DeleteImportedKeyMaterial 仅在包含导入的密钥材料的 KMS 密钥上有效（`Origin` 为 `EXTERNAL`）。	支持	是	是	是	是
DeriveSharedSecret	否	否	否	否	是

Amazon KMS API 操作	对称加密 KMS 密钥	HMAC KMS 密钥	非对称 KMS 密钥 (ENCRYP DECRYPT	非对称 KMS 密钥 (SIGN_VERIFY)	非对称 KMS 密钥（KEY_AGREEMENT）
DescribeKey	是	是	是	是	是
DisableKey	是	是	是	是	是
DisableKeyRotation	是 仅对带有密钥材料 (isAWS_KMS Origin 的 KMS 密 Amazon KMS 钥有效。	否	否	否	否
EnableKey	是	是	是	是	是
EnableKeyRotation	是 仅对带有密钥材料 (isAWS_KMS Origin 的 KMS 密 Amazon KMS 钥有效。	否	否	否	否

Amazon KMS API 操作	对称加密 KMS 密钥	HMAC KMS 密钥	非对称 KMS 密钥 (ENCRYPT DECRYPT	非对称 KMS 密钥 (SIGN_VERIFY)	非对称 KMS 密钥（KEY_AGREEMENT）
Encrypt	是	否	是	否	否
GenerateDataKey	是	否	否	否	否
GenerateDataKeyPair 生成受对称加密 KMS 密钥保护的非对称数据密钥对。	是 在自定义密钥存储中的 KMS 密钥上无效。	否	否	否	否
GenerateDataKeyPairWithoutPlaintext 生成受对称加密 KMS 密钥保护的非对称数据密钥对。	是 在自定义密钥存储中的 KMS 密钥上无效。	否	否	否	否
GenerateDataKeyWithoutPlaintext	是	否	否	否	否
GenerateMac	否	是	否	否	否
GetKeyPolicy	是	是	是	是	是

Amazon KMS API 操作	对称加密 KMS 密钥	HMAC KMS 密钥	非对称 KMS 密钥 (ENCRYP DECRYP	非对称 KMS 密钥 (SIGN_VERIFY)	非对称 KMS 密钥（KEY_AGREEMENT）
GetKeyRotationStatus	是	是 （KeyRotonEnabl 将始终为 false。	是 （KeyRotonEnabl 将始终为 false。	是 （KeyRotationEnabled 将始终为 false。）	是 （KeyRotationEnabled 将始终为 false。）
GetParametersForImport 仅在包含导入的密钥材料的 KMS 密钥上有效（Origin 为 EXTERNAL）。	支持	是	是	是	是
GetPublicKey	否	否	是	是	是
ImportKeyMaterial 仅在包含导入的密钥材料的 KMS 密钥上有效（Origin 为 EXTERNAL）。	支持	是	是	是	是
ListAliases	是	是	是	是	是
ListGrants	是	是	是	是	是
ListKeyPolicies	是	是	是	是	是

Amazon KMS API 操作	对称加密 KMS 密钥	HMAC KMS 密钥	非对称 KMS 密钥 (ENCRYP DECRYP	非对称 KMS 密钥 (SIGN_VERIFY)	非对称 KMS 密钥（KEY_AG REEMENT）
ListKeyRotations	是	是 （Rotati 字段将始终为 null 或空。）	是 （Rotati 字段将始终为 null 或空。）	是 （Rotations 字段将始终为 null 或空。）	是 （Rotations 字段将始终为 null 或空。）
ListResourceTags	支持	是	是	是	是
ListRetirableGrants	是	是	是	是	是
PutKeyPolicy	是	是	是	是	是
ReEncrypt	是	否	是	否	否
ReplicateKey - 仅在多区域密钥上有效	支持	是	是	是	是
RetireGrant	是	是	是	是	是
RevokeGrant	是	是	是	是	是

Amazon KMS API 操作	对称加密 KMS 密钥	HMAC KMS 密钥	非对称 KMS 密钥 (ENCRYP DECRYP	非对称 KMS 密钥 (SIGN_VERIFY)	非对称 KMS 密钥（KEY_AG REEMENT）
RotateKey OnDemand	是 仅对来源为 AWS_KMS 或 EXTERNAI 的客户自主管理型对称加密 KMS 密钥有效。	否	否	否	否
ScheduleK eyDeletion	是	是	是	是	是
Sign	否	否	否	是	否
TagResource	是	是	是	是	是
UntagResource	是	是	是	是	是

Amazon KMS API 操作	对称加密 KMS 密钥	HMAC KMS 密钥	非对称 KMS 密钥 (ENCRYP DECRYPT	非对称 KMS 密钥 (SIGN_VERIFY)	非对称 KMS 密钥（KEY_AGREEMENT）
UpdateAlias 当前 KMS 密钥和新的 KMS 密钥必须为相同类型（要么都是对称的，要么都是非对称的，要么都是 HMAC），并且它们必须用于相同的密钥用途。	支持	是	是	是	是
UpdateKeyDescription	是	是	是	是	是
UpdateReplicaRegion - 仅在多区域密钥上有效	支持	是	是	是	是
Verify	否	否	否	是	否
VerifyMac	否	是	否	否	否

特殊功能表

下表显示了每种类型的特殊用途密钥支持的 Amazon KMS API 操作。

在阅读此表时，请注意以下交互：

- 多区域密钥：

  - 多区域密钥可以是对称加密 KMS 密钥、非对称 KMS 密钥、HMAC KMS 密钥，以及包含导入的密钥材料的 KMS 密钥。
  - 您不能在自定义密钥存储中创建多区域密钥。
- 导入的密钥材料
  - 您可以导入对称加密 KMS 密钥、非对称 KMS 密钥和 HMAC KMS 密钥的密钥材料。
  - 您可创建具有导入密钥材料的多区域密钥。
  - 您不能在自定义密钥存储中使用导入的密钥材料创建密钥。
  - 带有导入密钥材料的 KMS 密钥不支持自动密钥轮换 (`EnableKeyRotation`、`DisableKeyRotation`)。
  - 使用导入的密钥材料对称加密 KMS 密钥支持按需密钥轮换 (`RotateKeyOnDemand`)。
- 自定义密钥存储
  - 自定义密钥存储仅支持对称加密 KMS 密钥。
  - 自定义密钥存储中的 KMS 密钥不支持对非对称密钥对（`GenerateDataKeyPair`、`GenerateDataKeyPairWithoutPlaintext`）进行对称操作。
  - 自定义密钥存储中的 KMS 密钥不支持自动密钥转换 (`EnableKeyRotation`、`DisableKeyRotation`)。
  - 您不能在自定义密钥存储中创建多区域密钥。

您可能需要水平或垂直滚动才能查看此表中的所有数据。

Amazon KMS API 操作	多区域密钥	导入的密钥材料	自定义密钥存储中的 KMS 密钥
CancelKeyDeletion	✓	✓	✓
CreateAlias	✓	✓	✓
CreateGrant	✓	✓	✓
CreateKey 您可以使用 `CreateKey` 创建多区域主键、包含导入的密钥材料的 KMS	✓	✓	✓

Amazon KMS API 操作	多区域密钥	导入的密钥材料	自定义密钥存储中的 KMS 密钥
密钥，或自定义密钥存储中的 KMS 密钥。若要创建多区域副本密钥，请使用 ReplicateKey 。			
Decrypt	✓ 仅当 KeyUsage 为 ENCRYPT_DECRYPT 时才有效	✓	✓
DeleteAlias	✓	✓	✓
DeleteImportedKeyMaterial	✓ 仅对包含导入的密钥材料的密钥有效（Origin 为 EXTERNAL）	✓	⊗
DeriveSharedSecret	✓ 仅在 KeyUsage 为 KEY_AGREEMENT 时有效）	✓ 仅在 KeyUsage 为 KEY_AGREEMENT 时有效）	⊗
DescribeKey	✓	✓	✓
DisableKey	✓	✓	✓

Amazon KMS API 操作	多区域密钥	导入的密钥材料	自定义密钥存储中的 KMS 密钥
DisableKeyRotation	✓ 仅对带有密钥材料 (OriginisAWS_KM 的对称加密密 Amazon KMS 钥有效。	⊗	⊗
EnableKey	✓ 仅在对称加密 KMS 密钥上有效	✓	✓
EnableKeyRotation	✓ 仅对带有密钥材料 (OriginisAWS_KM 的对称加密密 Amazon KMS 钥有效。	⊗	⊗
Encrypt	✓ 仅当 KeyUsage 为 ENCRYPT_D ECRYPT 时才有效	✓	✓

Amazon KMS API 操作	多区域密钥	导入的密钥材料	自定义密钥存储中的 KMS 密钥
GenerateDataKey	✓ 仅在对称加密 KMS 密钥上有效	✓	✓
GenerateDataKeyPair	✓ 仅在对称加密 KMS 密钥上有效	✓	⊗
GenerateDataKeyPairWithoutPlaintext	✓ 仅在对称加密 KMS 密钥上有效	✓	⊗
GenerateDataKeyWithoutPlaintext	✓ 仅在对称加密 KMS 密钥上有效	✓	✓
GenerateMac 仅在 HMAC KMS 密钥上有效	✓	✓	⊗
GetKeyPolicy	✓	✓	✓
GetKeyRotationStatus	✓	✓ （KeyRotationEnabled 将始终为 false。）	⊗

Amazon KMS API 操作	多区域密钥	导入的密钥材料	自定义密钥存储中的 KMS 密钥
GetParametersForImport	✓ 仅对包含已导入的密钥材料的密钥有效（Origin 为 EXTERNAL）。	✓	⊗
GetPublicKey 仅对非对称 KMS 密钥有效。	✓	✓	⊗
ImportKeyMaterial	✓ 仅对包含已导入的密钥材料的密钥有效（Origin 为 EXTERNAL）。	✓	⊗
ListAliases	✓	✓	✓
ListGrants	✓	✓	✓
ListKeyPolicies	✓	✓	✓
ListKeyRotations	✓ 仅对EXTERNAL源自AWS_KMS或的对称加密密钥有效。	✓ 仅对对称加密密钥有效。	✓

Amazon KMS API 操作	多区域密钥	导入的密钥材料	自定义密钥存储中的 KMS 密钥
ListResourceTags	✓	✓	✓
ListRetirableGrants	✓	✓	✓
PutKeyPolicy	✓	✓	✓
ReEncrypt	✓ 仅当 KeyUsage 为 ENCRYPT_DECRYPT 时才有效	✓	✓
ReplicateKey	✓ 仅在多区域主键上有效。	✓ 仅在多区域主键上有效。	⊗
RetireGrant	✓	✓	✓
RevokeGrant	✓	✓	✓
RotateKeyOnDemand	✓ 仅对EXTERNAL源自AWS_KMS或的对称加密密钥有效。	✓ 仅对对称加密密钥有效。	⊗
ScheduleKeyDeletion	✓	✓	✓

Amazon KMS API 操作	多区域密钥	导入的密钥材料	自定义密钥存储中的 KMS 密钥
Sign 仅当 KeyUsage 为 SIGN_VERIFY 时才有效。	✓	✓	⊗
TagResource	✓	✓	✓
UntagResource	✓	✓	✓
UpdateAlias – 当前 KMS 密钥和新的 KMS 密钥必须为相同类型（要么都是对称的，要么都是非对称的，要么都是 HMAC），并且它们必须用于相同的密钥用途。	✓	✓	✓
UpdateKeyDescription	✓	✓	✓
UpdateReplicaRegion	✓	✓ 仅在多区域密钥上有效。	⊗
Verify 仅当 KeyUsage 为 SIGN_VERIFY 时才有效。	✓	✓	⊗
VerifyMac 仅在 HMAC KMS 密钥上有效	✓	✓	⊗

密钥规范引用

创建非对称 KMS 密钥或 HMAC KMS 密钥时，可以选择其密钥规范。密钥规范是每个密钥的属性 Amazon KMS key，代表您的 KMS 密钥的加密配置。密钥规范在创建 KMS 密钥时选择，并且无法更改。如果选择了错误的密钥规范，可删除 KMS 密钥，然后创建一个新的密钥规范。

> **Note**
>
> KMS 密钥的密钥规范被称为“客户主密钥规范”。该CreateKey操作的`CustomerMasterKeySpec`参数已被弃用。请改用 `KeySpec` 参数。`CreateKey`和DescribeKey操作的响应包括具有相同值的`KeySpec`和`CustomerMasterKeySpec`成员。

密钥规范确定 KMS 密钥是对称还是非对称、KMS 密钥中密钥材料的类型以及 Amazon KMS 支持 KMS 密钥的加密算法、签名算法或消息身份验证码 (MAC) 算法。选择哪个密钥规范通常取决于使用案例和法规要求。但是，KMS 密钥加密操作的密钥规范不同，其价格和限额也不同。有关定价的详细信息，请参阅 Amazon Key Management Service 定价。有关请求配额的信息，请参阅 请求配额。

要限制委托人在创建 KMS 密钥时可以使用的密钥规范，请使用 k ms: KeySpec 条件密钥。您还可以使用`kms:KeySpec`条件密钥来允许委托人仅对具有特定密钥规范的 KMS 密钥调用 Amazon KMS 操作。例如，您可以拒绝删除具有 `RSA_4096` 密钥规范的 KMS 密钥的计划权限。

Amazon KMS 支持 KMS 密钥的以下关键规格：

对称加密密钥规范（默认值）

- SYMMETRIC_DEFAULT

RSA 密钥规范（加密和解密或签名和验证）

- RSA_2048
- RSA_3072
- RSA_4096

椭圆曲线密钥规范

- 非对称 NIST 标准椭圆曲线密钥对（签名和验证——或者——派生共享密钥）
  - ECC_NIST_P256 (secp256r1)
  - ECC_NIST_P384 (secp384r1)
  - ECC_NIST_P521 (secp521r1)

- ECC_NIST_ EDWARDS25519 (ed25519)-仅限签名和验证
  - 注意：对于 ECC_NIST_ EDWARDS25519 KMS 密钥，_SHA_512 签名算法需要，而 ED25519 _PH_SHA_512 则需要。`MessageType:RAW` `ED25519MessageType:DIGEST`这些消息类型不能互换使用。
- 其他非对称椭圆曲线密钥对（签名和验证）
  - ECC_SECG_P256K1 (secp256k1)，常用于加密货币。

SM2 密钥规范（加密和解密——或者——签名和验证——或者——派生共享机密）

- SM2（仅限中国地区）

HMAC 密钥规范

- HMAC_224
- HMAC_256
- HMAC_384
- HMAC_512

ML-DSA 密钥规范

- ML_DSA_44
- ML_DSA_65
- ML_DSA_87

SYMMETRIC_DEFAULT 密钥规范

默认密钥规范 SYMMETRIC_DEFAULT 是对称加密 KMS 密钥的密钥规范。当您在 Amazon KMS 控制台中选择 Symmetric 密钥类型以及加密和解密密钥用法时，它会选择密钥规范。`SYMMETRIC_DEFAULT`在CreateKey操作中，如果您未指定`KeySpec`值，则会选择 SYMMETRIC_DEFAULT。如果您没有理由使用其他密钥规范，SymMETIC_DEFAULT 是个不错的选择。

SYMMETRIC_DEFAULT 代表 AES-256-GCM，这是一种基于伽罗瓦计数器模式（GCM）中的高级加密标准（AES）的对称算法，具有 256 位密钥，是用于安全加密的行业标准。此算法生成的密文支持附加身份验证数据 (AAD)，如加密上下文，且 GCM 对密文提供额外的完整性检查。

在 AES-256-GCM 下加密的数据现在和将来都受到保护。密码学家认为这种算法具备抗量子性。理论上，在未来针对使用 256 位 AES-GCM 密钥创建的密文的大规模量子计算攻击中，密钥的有效安全性将会降至 128 位。但是，这种安全级别足以使对 Amazon KMS 密文的暴力攻击变得不可行。

唯一的例外是中国区域，其中 SYMMETRIC_DEFAULT 表示使用加密的 128 位对称密钥。 SM4 您只能在中国区域内创建 128 位 SM4 密钥。您无法在中国区域内创建 256 位的 AES-GCM KMS 密钥。

您可以使用中的对称加密 KMS 密钥 Amazon KMS 来加密、解密和重新加密数据，以及保护生成的数据密钥和数据密钥对。 Amazon 与之集成的服务 Amazon KMS 使用对称加密 KMS 密钥对您的静态数据进行加密。您可以将自己的密钥材料导入对称加密 KMS 密钥中，并在自定义密钥存储中创建对称加密 KMS 密钥。有关可以对对称 KMS 密钥和非对称 KMS 密钥执行的操作的比较表格，请参阅比较对称 KMS 密钥与非对称 KMS 密钥。

您可以使用中的对称加密 KMS 密钥 Amazon KMS 来加密、解密和重新加密数据，并生成数据密钥和数据密钥对。您可以创建多区域对称加密 KMS 密钥，将自己的密钥材料导入对称加密 KMS 密钥中，并在自定义密钥存储中创建对称加密 KMS 密钥。有关您可以对不同类型 KMS 密钥执行的操作进行比较的表格，请参阅 密钥类型引用。

RSA 密钥规范

使用 RSA 密钥规范时， Amazon KMS 会创建一个带有 RSA 密钥对的非对称 KMS 密钥。私钥永远不会处于 Amazon KMS 未加密状态。你可以在里面使用公钥 Amazon KMS，也可以下载公钥在外面使用 Amazon KMS。

> **Warning**
>
> 在外部加密数据时 Amazon KMS，请确保可以解密密文。如果您使用已从 Amazon KMS删除的 KMS 密钥中的公有密钥、配置用于签名和验证的 KMS 密钥中的公有密钥或者使用 KMS 密钥不支持的加密算法，则数据将无法恢复。

在中 Amazon KMS，您可以使用带有 RSA 密钥对的非对称 KMS 密钥进行加密和解密，或者签名和验证，但不能两者兼而有之。此属性（称为Key usage）与密钥规范分开确定，但应在选择密钥规范之前做出决定。

Amazon KMS 支持以下用于加密和解密或签名和验证的 RSA 密钥规范：

- RSA_2048
- RSA_3072
- RSA_4096

RSA 密钥规范因 RSA 密钥长度（以位为单位）而异。选择哪个 RSA 密钥规范可能取决于安全标准或任务要求。一般来说，可使用对任务而言实用又实惠的最大密钥。KMS 密钥加密操作的 RSA 密钥规

范不同，其价格也不同。有关 Amazon KMS 定价的信息，请参阅Amazon 密钥管理服务定价。有关请求配额的信息，请参阅 请求配额。

用于加密和解密的 RSA 密钥规范

使用 RSA 非对称 KMS 密钥行加密和解密时，用公有密钥加密，然后用私有密钥解密。当您调用 RSA KMS 密钥的Encrypt操作时，会 Amazon KMS 使用 RSA 密钥对中的公钥和您指定的加密算法来加密您的数据。 Amazon KMS 要解密密文，请调用Decrypt操作并指定相同的 KMS 密钥和加密算法。Amazon KMS 然后使用 RSA key pair 中的私钥来解密您的数据。

您也可以下载公钥并使用它来加密外部的数据 Amazon KMS。请务必使用 Amazon KMS 支持 RSA KMS 密钥的加密算法。要解密密文，请采用相同的 KMS 密钥和加密算法调用 Decrypt 函数。

Amazon KMS 支持两种具有 RSA 密钥规格的 KMS 密钥加密算法。这些算法在 PKCS #1 2.2 版中定义，它们内部使用的哈希函数有所不同。在中 Amazon KMS，RSAES_OAEP 算法始终使用相同的哈希函数进行哈希处理和掩码生成函数 ()。MGF1调用 Encrypt 和 Decrypt 操作时，需要指定加密算法。您可以为每个请求选择不同的算法。

支持 RSA 密钥规范的加密算法

加密算法	算法描述
RSAES_OAEP_SHA_1	PKCS #1 2.2 版，7.1 节。使用 OAEP Padding 进行 RSA 加密，在哈希和 MGF1 掩码生成函数中使用 SHA-1 以及空标签。
RSAES_OAEP_SHA_256	PKCS #1，7.1 节。使用 OAEP Padding 进行 RSA 加密，在哈希和 MGF1 掩码生成函数中使用 SHA-256 以及空标签。

无法将 KMS 密钥配置为使用特定的加密算法。但是，您可以使用 kms: EncryptionAlgorithm 策略条件来指定允许委托人使用 KMS 密钥的加密算法。

要获取 KMS 密钥的加密算法，请在 Amazon KMS 控制台中查看 KMS 密钥的加密配置或使用DescribeKey操作。 Amazon KMS 当您在 Amazon KMS 控制台中或使用GetPublicKey操作下载公钥时，还会提供密钥规范和加密算法。

您可以根据可在每个请求中加密的明文数据的长度，选择 RSA 密钥规范。下表显示了在对 Encrypt 操作的单次调用中，可以加密的明文的最大长度（以字节为单位）。值因密钥规范和加密算法而异。为进行比较，使用对称加密 KMS 密钥一次最多可加密 4096 字节。

要计算这些算法的最大明文长度（以字节为单位），请使用以下公式：(*key_size_in_bits*/8)-(2 * *hash_length_in_bits* /8)-2。例如，对于具有 SHA-256 的 RSA_2048，最大明文长度为 (2048/8) - (2 * 256/8) -2 = 190 字节。

Encrypt 操作中的最大明文长度（以字节为单位）

	加密算法	
密钥规范	RSAES_OAEP_SHA_1	RSAES_OAEP_SHA_256
RSA_2048	214	190
RSA_3072	342	318
RSA_4096	470	446

用于签名和验证的 RSA 密钥规范

使用 RSA 非对称 KMS 密钥进行签名和验证时，用私有密钥为消息生成签名，然后用公有密钥验证签名。

当您调用非对称 KMS 密钥的`Sign`操作时，会 Amazon KMS 使用 RSA 密钥对中的私钥、消息和您指定的签名算法来生成签名。 Amazon KMS 要验证签名，请调用 Verify 操作。指定签名，以及相同的 KMS 密钥、消息和签名算法。 Amazon KMS 然后使用 RSA key pair 中的公钥来验证签名。您也可以下载公钥并使用它来验证外部的签名 Amazon KMS。

Amazon KMS 支持所有具有 RSA 密钥规范的 KMS 密钥的以下签名算法。在您调用 Sign 和 Verify 操作后，需要指定签名算法。您可以为每个请求选择不同的算法。使用 RSA 密钥对进行签名时，首选 RSASSA-PSS 算法。为了与现有应用程序兼容，我们加入了 RSASSA-PKCS1-v1_5 算法。

支持 RSA 密钥规范的签名算法

签名算法	算法描述
RSASSA_PSS_SHA_256	PKCS #1 v2.2，第 8.1 节，带有 PSS 填充的 RSA 签名，消息摘要和 MGF1 掩码生成功能均使用 SHA-256 以及 256 位盐

签名算法	算法描述
RSASSA_PSS_SHA_384	PKCS #1 v2.2，第 8.1 节，带有 PSS 填充的 RSA 签名，消息摘要和 MGF1 掩码生成功能均使用 SHA-384 以及 384 位盐
RSASSA_PSS_SHA_512	PKCS #1 v2.2，第 8.1 节，带有 PSS 填充的 RSA 签名，消息摘要和 MGF1 掩码生成功能均使用 SHA-512 以及 512 位盐
RSSASSA_ _v1_5_SHA_256 PKCS1	PKCS #1 2.2 版，8.2 节，具有 PKCS #1v1.5 填充和 SHA-256 的 RSA 签名
RSSASSA_ _V1_5_SHA_384 PKCS1	PKCS #1 2.2 版，8.2 节，具有 PKCS #1v1.5 填充和 SHA-384 的 RSA 签名
RASSASSA_ _V1_5_5_SHA_512 PKCS1	PKCS #1 2.2 版，8.2 节，具有 PKCS #1v1.5 填充和 SHA-512 的 RSA 签名

无法将 KMS 密钥配置为使用特定的签名算法。但是，您可以使用 kms: SigningAlgorithm 策略条件来指定允许委托人使用 KMS 密钥的签名算法。

要获取 KMS 密钥的签名算法，请在 Amazon KMS 控制台中或使用DescribeKey操作查看 KMS 密钥的加密配置。 Amazon KMS 当您在 Amazon KMS 控制台中或使用GetPublicKey操作下载公钥时，还会提供密钥规范和签名算法。

椭圆曲线密钥规范

使用椭圆曲线 (ECC) 密钥规范时， Amazon KMS 会创建一个带有 ECC 密钥对的非对称 KMS 密钥，用于签名和验证或派生共享密钥（但不能两者兼而有之）。生成签名或派生共享密钥的私有密钥永远不会让 Amazon KMS 处于未加密状态。您可以使用公钥来验证内部的签名 Amazon KMS，也可以下载公钥以在外部使用 Amazon KMS。

使用 Edwards Curve 密钥规范时， Amazon KMS 会创建一个带有 Ed25519 密钥对的非对称 KMS 密钥进行签名和验证。生成签名的私钥永远不会处于 Amazon KMS 未加密状态。您可以使用公钥来验证内部的签名 Amazon KMS，也可以下载公钥以在外部使用 Amazon KMS。

Amazon KMS 支持非对称 KMS 密钥的以下 ECC 密钥规范。

- 非对称 NIST 标准椭圆曲线密钥对（签名和验证——或者——派生共享密钥）
  - ECC_NIST_P256 (secp256r1)
  - ECC_NIST_P384 (secp384r1)
  - ECC_NIST_P521 (secp521r1)
  - ECC_NIST_ EDWARDS25519 (ed25519)-仅限签名和验证
    - 注意：对于 ECC_NIST_ EDWARDS25519 KMS 密钥，_SHA_512 签名算法需要，而 ED25519 _PH_SHA_512 则需要。MessageType:RAW ED25519MessageType:DIGEST这些消息类型不能互换使用。
- 其他非对称椭圆曲线密钥对（签名和验证）
  - ECC_SECG_P256K1 (secp256k1)，常用于加密货币。

选择哪个 ECC 密钥规范可能取决于安全标准或任务要求。一般来说，可使用对任务而言实用又实惠的点最多的曲线。

如果您要创建非对称 KMS 密钥来派生共享密钥，请使用 NIST 标准的椭圆曲线密钥规范之一（ECC_SECG_P256K1 和 ECC_NIST_ 除外）。EDWARDS25519唯一支持的用于派生共享密钥的密钥协议算法是椭圆曲线加密辅助因子 Diffie-Hellman Priman Primitive（ECDH）。有关如何离线派生共享密钥的示例，请参阅the section called “离线派生共享密钥”。

如果您正在创建非对称 KMS 密钥以用于加密货币，请使用 ECC_SEG_P256K1 密钥规范。此密钥规范也可以用于其他目的，但比特币和其他加密货币必须使用此密钥规范。

下表显示了每个 ECC 密钥规范所 Amazon KMS 支持的签名算法。无法将 KMS 密钥配置为使用特定的签名算法。但是，您可以使用 kms: SigningAlgorithm 策略条件来指定允许委托人使用 KMS 密钥的签名算法。

支持 ECC 密钥规范的签名算法

密钥规范	签名算法	算法描述
ECC_NIST_P256	ECDSA_SHA_256	NIST FIPS 186-4, Section 6.4, ECDSA signature using the curve specified by the key and SHA-256 表示消息摘要。
ECC_NIST_P384	ECDSA_SHA_384	NIST FIPS 186-4, Section 6.4, ECDSA signature using the

密钥规范	签名算法	算法描述
		curve specified by the key and SHA-384 表示消息摘要。
ECC_NIST_P521	ECDSA_SHA_512	NIST FIPS 186-4, Section 6.4, ECDSA signature using the curve specified by the key and SHA-512 表示消息摘要。
ECC_SECG_P256K1	ECDSA_SHA_256	NIST FIPS 186-4, Section 6.4, ECDSA signature using the curve specified by the key and SHA-256 表示消息摘要。
ECC_NIST_ EDWARDS25519	ED25519_SHA_512	NIST FIPS 186-5, Section 7, EdDSA signature using the curve specified by the key and SHA-512 表示消息摘要。KMS 需要`MessageType:RAW` 使用此算法。
ECC_NIST_ EDWARDS25519	ED25519_PH_SHA_512	NIST FIPS 186-5, Section 7, EdDSA signature using the curve specified by the key and SHA-512 表示消息摘要。KMS 需要`MessageType:DIGEST` 使用此算法。

HMAC KMS 密钥的密钥规范

Amazon KMS 支持不同长度的对称 HMAC 密钥。您选择的密钥规范可取决于安全、法规或业务要求。密钥的长度决定了GenerateMac和VerifyMac操作中使用的 MAC 算法。一般来说，较长的密钥更安全。使用适用于您的使用场景的最长密钥。

HMAC 密钥规范	MAC 算法
HMAC_224	HMAC_SHA_224
HMAC_256	HMAC_SHA_256
HMAC_384	HMAC_SHA_384
HMAC_512	HMAC_SHA_512

ML-DSA 密钥规范

ML-DSA 密钥是 Module-Lattice-Based数字签名算法 (ML-DSA) 中使用的加密密钥，该算法专为后量子密码学而设计。该算法是美国国家标准与技术研究院（NIST）标准化工作的一部分，详见联邦信息处理标准（FIPS）204 所述。

ML-DSA 密钥用于公私密钥对系统。私有密钥用于对数据进行签名，公有密钥则用于验证签名。即使面对潜在的量子计算机威胁，该系统也可确保数字消息或文档的真实性、完整性和不可否认性。

使用 ML-DSA 密钥规范创建密钥时， Amazon KMS 会创建一个带有 ML-DSA 密钥对的非对称 KMS 密钥用于签名和验证。生成签名的私钥永远不会处于 Amazon KMS 未加密状态。您可以使用公钥来验证内部的签名 Amazon KMS，也可以下载公钥以在外部使用 Amazon KMS。

Amazon KMS 支持以下 ML-DSA 非对称 KMS 密钥规范：

- ML_DSA_44
- ML_DSA_65
- ML_DSA_87

Amazon KMS 支持所有 ML-DSA 密钥规范的 ML_DSA_SHAKE_256 签名算法。

SM2 密钥规范（仅限中国地区）

该 SM2 密钥规范是在中国国家商用密码管理局 (OSCCA) 发布的一 GM/T 系列规范中定义的椭圆曲线密钥规范。密 SM2 钥规范仅在中国地区可用。使用 SM2 密钥规范时，使用密钥对 Amazon KMS 创建非对称 KMS SM2 密钥。你可以在里面使用你的 SM2 密钥 pair Amazon KMS，也可以下载公钥在外面使用 Amazon KMS。有关更多信息，请参阅 the section called “使用 SM2 密钥对进行离线验证（仅限中国区域）”。

每个 KMS 密钥都只能有一个Key usage。您可以使用 SM2 KMS 密钥进行签名和验证、加密和解密或获取共享密钥。在创建 KMS 密钥时，您必须指定密钥用途，并且密钥一经创建便无法更改。

如果您要创建非对称 KMS 密钥来派生共享密钥，请使用密 SM2 钥规范。唯一支持的用于派生共享密钥的密钥协议算法是椭圆曲线加密辅助因子 Diffie-Hellman Priman Primitive（ECDH）。

Amazon KMS 支持以下 SM2 加密和签名算法：

- SM2PKE 加密算法

  SM2PKE 是一种基于椭圆曲线的加密算法，由 OSCCA 在 0003.4-2012 中定义。 GM/T
- SM2DSA 签名算法

  SM2DSA 是一种基于椭圆曲线的签名算法，由 OSCCA 在 0003.2-2012 中定义。 GM/T SM2DSA 需要一个可区分的 ID，该ID使用哈希算法进行 SM3 哈希处理，然后与您传递给的消息或消息摘要组合在一起。 Amazon KMS然后对这个串联的值进行哈希处理并由其签名。 Amazon KMS

Amazon KMS 权限

此表旨在帮助您了解 Amazon KMS 权限，以便您可以控制对 Amazon KMS 资源的访问。表格下方会显示列标题的定义。

您还可以在服务授权参考的 Amazon Key Management Service 的操作、资源和条件键主题中了解 Amazon KMS 权限。但是，该主题并未列出可用于优化每个权限的所有条件键。

要详细了解哪些 Amazon KMS 操作对称加密 KMS 密钥、非对称 KMS 密钥以及 HMAC KMS 密钥有效，请参阅密钥类型引用。

> **Note**
>
> 您可能需要水平或垂直滚动才能查看表中的所有数据。

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
CancelKeyDeletion	密钥策略	否	KMS 密钥	KMS 密钥操作的条件：

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
kms:CancelKeyDeletion				kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService
ConnectCustomKeyStore kms:ConnectCustomKeyStore	IAM 策略	否	*	kms:CallerAccount

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
CreateAlias kms:CreateAlias	IAM policy（适用于别名）	否	别名	无（控制对别名的访问时）
要使用此操作，调用方需要对以下两个资源具有 kms:CreateAlias 权限： • 别名（在 IAM policy 中） • KMS 密钥（在密钥策略中） 有关更多信息，请参阅 控制对别名的访问。	密钥策略（适用于 KMS 密钥）	否	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService
CreateCustomKeyStore kms:CreateCustomKeyStore	IAM 策略	否	*	kms:CallerAccount

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
CreateGrant kms:CreateGrant	密钥策略	是	KMS 密钥	加密上下文条件： kms:EncryptionContext:context-key kms:EncryptionContextKeys 授予条件： kms:GrantConstraintType kms:GranteePrincipal kms:GrantIsForAWSResource kms:GrantOperations kms:RetiringPrincipal KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
				aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService
CreateKey kms:CreateKey	IAM 策略	否	*	kms:BypassPolicyLockoutSafetyCheck kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ViaService aws:RequestTag/tag-key（Amazon 全局条件键） aws:ResourceTag/tag-key（Amazon 全局条件键） aws:TagKeys（Amazon 全局条件键）

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
Decrypt kms:Decrypt	密钥策略	是	KMS 密钥	加密操作的条件 kms:EncryptionAlgorithm kms:RequestAlias 加密上下文条件： kms:EncryptionContext:context-key kms:EncryptionContextKeys KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
DeleteAlias kms:DeleteAlias	IAM policy（适用于别名）	否	别名	无（控制对别名的访问时）
要使用此操作，调用方需要对以下两个资源具有 kms:DeleteAlias 权限： • 别名（在 IAM policy 中） • KMS 密钥（在密钥策略中） 有关更多信息，请参阅 控制对别名的访问。	密钥策略（适用于 KMS 密钥）	否	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService
DeleteCustomKeyStore kms:DeleteCustomKeyStore	IAM 策略	否	*	kms:CallerAccount

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
DeleteImportedKeyMaterial kms:DeleteImportedKeyMaterial	密钥策略	否	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
DeriveSharedSecret kms:DeriveSharedSecret	密钥策略	是	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService 加密操作的条件： kms:KeyAgreementAlgorithm
DescribeCustomKeyStores kms:DescribeCustomKeyStores	IAM 策略	否	*	kms:CallerAccount

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
DescribeKey kms:DescribeKey	密钥策略	是	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService 其他条件： kms:RequestAlias

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
DisableKey kms:DisableKey	密钥策略	否	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
DisableKeyRotation kms:DisableKeyRotation	密钥策略	否	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService
DisconnectCustomKeyStore kms:DisconnectCustomKeyStore	IAM 策略	否	*	kms:CallerAccount

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
EnableKey kms:EnableKey	密钥策略	否	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
EnableKeyRotation kms:EnableKeyRotation	密钥策略	否	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService 自动密钥轮换条件： kms:RotationPeriodInDays

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
Encrypt kms:Encrypt	密钥策略	是	KMS 密钥	加密操作的条件 kms:EncryptionAlgorithm kms:RequestAlias 加密上下文条件： kms:EncryptionContext:context-key kms:EncryptionContextKeys KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
GenerateDataKey kms:GenerateDataKey	密钥策略	是	KMS 密钥	加密操作的条件 kms:EncryptionAlgorithm kms:RequestAlias 加密上下文条件： kms:EncryptionContext:context-key kms:EncryptionContextKeys KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
GenerateDataKeyPair kms:GenerateDataKeyPair	密钥策略	是	KMS 密钥 生成受对称加密 KMS 密钥保护的非对称数据密钥对。	数据密钥对的条件： kms:DataKeyPairSpec 加密操作的条件 kms:EncryptionAlgorithm kms:RequestAlias 加密上下文条件： kms:EncryptionContext:context-key kms:EncryptionContextKeys KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键）

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
				kms:ViaService

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
GenerateDataKeyPairWithoutPlaintext kms:GenerateDataKeyPairWithoutPlaintext	密钥策略	是	KMS 密钥 生成受对称加密 KMS 密钥保护的非对称数据密钥对。	数据密钥对的条件： kms:DataKeyPairSpec 加密操作的条件 kms:EncryptionAlgorithm kms:RequestAlias 加密上下文条件： kms:EncryptionContext:context-key kms:EncryptionContextKeys KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键）

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
				kms:ViaService
GenerateDataKeyWithoutPlaintext kms:GenerateDataKeyWithoutPlaintext	密钥策略	是	KMS 密钥	加密操作的条件 kms:EncryptionAlgorithm kms:RequestAlias 加密上下文条件： kms:EncryptionContext:context-key kms:EncryptionContextKeys KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
GenerateMac kms:GenerateMac	密钥策略	是	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService 加密操作的条件： kms:MacAlgorithm kms:RequestAlias
GenerateRandom kms:GenerateRandom	IAM 策略	不适用	*	无

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
GetKeyPolicy kms:GetKeyPolicy	密钥策略	否	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
GetKeyRotationStatus kms:GetKeyRotationStatus	密钥策略	是	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
GetParametersForImport kms:GetParametersForImport	密钥策略	否	KMS 密钥	kms:WrappingAlgorithm kms:WrappingKeySpec KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
GetPublicKey kms:GetPublicKey	密钥策略	是	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService 其他条件： kms:RequestAlias

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
ImportKeyMaterial kms:ImportKeyMaterial	密钥策略	否	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService 其他条件：kms:ExpirationModel kms:ValidTo
ListAliases kms:ListAliases	IAM 策略	否	*	无

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
ListGrants kms:ListGrants	密钥策略	是	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService 其他条件： kms:GrantIsForAWSResource

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
ListKeyPolicies kms:ListKeyPolicies	密钥策略	否	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
ListKeyRotations kms:ListKeyRotations	密钥策略	否	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService
ListKeys kms:ListKeys	IAM 策略	否	*	无

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
ListResourceTags kms:ListResourceTags	密钥策略	否	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService
ListRetirableGrants kms:ListRetirableGrants	IAM 策略	指定的委托人必须位于本地账户中，但操作将返回所有账户中的授权。	*	无

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
PutKeyPolicy kms:PutKeyPolicy	密钥策略	否	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService 其他条件： kms:BypassPolicyLockoutSafetyCheck

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
ReEncrypt kms:ReEncryptFrom kms:ReEncryptTo 要使用此操作，调用方需要对以下两个 KMS 密钥具有权限： • kms:ReEncryptFrom KMS 密钥上的 ，用于解密 • kms:ReEncryptTo KMS 密钥上的 ，用于加密	密钥策略	是	KMS 密钥	加密操作的条件 kms:EncryptionAlgorithm kms:RequestAlias 加密上下文条件： kms:EncryptionContext:context-key kms:EncryptionContextKeys KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService 其他条件：

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
				kms:ReEncryptOnSameKey
ReplicateKey kms:ReplicateKey 要使用此操作，调用方需要具有以下权限： • 多区域主键上的 kms:ReplicateKey • 副本区域中的 IAM policy 中的 kms:CreateKey	密钥策略	否	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService 其他条件： kms:ReplicaRegion

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
RetireGrant kms:RetireGrant 撤销授予的权限主要由授予决定。单独的策略无法允许访问此操作。有关更多信息，请参阅停用和撤销授权。	IAM 策略 （此权限在密钥策略中无效。）	是	KMS 密钥	加密上下文条件： kms:EncryptionContext:context-key kms:EncryptionContextKeys 授予条件： kms:GrantConstraintType KMS 密钥操作的条件： kms:CallerAccount kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
RevokeGrant kms:RevokeGrant	密钥策略	是	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService 其他条件： kms:GrantIsForAWSResource

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
RotateKeyOnDemand kms:RotateKeyOnDemand	密钥策略	否	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
ScheduleKeyDeletion kms:ScheduleKeyDeletion	密钥策略	否	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
Sign kms:Sign	密钥策略	是	KMS 密钥	签名和验证条件： kms:MessageType kms:RequestAlias kms:SigningAlgorithm KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
TagResource kms:TagResource	密钥策略	否	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService 标记条件： aws:RequestTag/tag-key（Amazon 全局条件键） aws:TagKeys（Amazon 全局条件键）

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
UntagResource kms:UntagResource	密钥策略	否	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService 标记条件： aws:RequestTag/tag-key（Amazon 全局条件键） aws:TagKeys（Amazon 全局条件键）

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
UpdateAlias kms:UpdateAlias	IAM policy（适用于别名）	否	别名	无（控制对别名的访问时）
要使用此操作，调用方需要对以下三个资源具有 kms:UpdateAlias 权限： • 别名 • 当前关联的 KMS 密钥 • 新关联的 KMS 密钥 有关更多信息，请参阅 控制对别名的访问。	密钥策略（适用于 KMS 密钥）	否	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService
UpdateCustomKeyStore kms:UpdateCustomKeyStore	IAM 策略	否	*	kms:CallerAccount

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
UpdateKeyDescription kms:UpdateKeyDescription	密钥策略	否	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
UpdatePrimaryRegion kms:UpdatePrimaryRegion 要使用此操作，调用方需要对将成为副本密钥的多区域主键和将成为主键的多区域副本密钥同时具有 kms:UpdatePrimaryRegion 权限。	密钥策略	否	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService 其他条件 kms:PrimaryRegion

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
Verify kms:Verify	密钥策略	是	KMS 密钥	签名和验证条件： kms:MessageType kms:RequestAlias kms:SigningAlgorithm KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService

操作和权限	策略类型	跨账户使用	资源（适用于 IAM policy）	Amazon KMS 条件键
VerifyMac kms:VerifyMac	密钥策略	是	KMS 密钥	KMS 密钥操作的条件： kms:CallerAccount kms:KeySpec kms:KeyUsage kms:KeyOrigin kms:MultiRegion kms:MultiRegionKeyType kms:ResourceAliases aws:ResourceTag/tag-key（Amazon 全局条件键） kms:ViaService 加密操作的条件： kms:MacAlgorithm kms:RequestAlias

列描述

此表中的各列提供以下信息：

- 操作和权限列出每个 Amazon KMS API 操作及允许执行该操作的权限。您可以在策略语句的 `Action` 元素中指定操作。
- 策略类型指示权限是否可在密钥策略或 IAM policy 中使用。

密钥策略意味着您可以在密钥策略中指定权限。当密钥政策包含启用 IAM policy 的策略语句时，您可以在 IAM policy 中指定权限。

IAM policy 意味着您只能在 IAM policy 中指定权限。

- 跨账户使用显示了授权用户可以对其他 Amazon Web Services 账户 中的资源执行的操作。

  值 Yes（是）表示委托人可以对其他 Amazon Web Services 账户 中的资源执行操作。

  值 No（否）表示委托人只能对其自己的 Amazon Web Services 账户 中的资源执行操作。

  如果您为不同账户中的委托人授予一个不能在跨账户资源上使用的权限，则该权限将无效。例如，如果您为另一个账户中的委托人授予对您账户中的 KMS 密钥的 kms:TagResource 权限，则他们试图在您的账户中标记 KMS 密钥将失败。

- 资源列出了应用权限的 Amazon KMS 资源。Amazon KMS 支持两种资源类型：KMS 密钥和别名。在密钥策略中，`Resource` 元素的值始终为 `*`，这表示密钥策略附加到的 KMS 密钥。

  使用以下值表示 IAM policy 中的 Amazon KMS 资源。

  KMS 密钥

  当资源是 KMS 密钥时，请使用其密钥 ARN。有关帮助信息，请参阅 the section called “查找密钥 ID 和密钥 ARN”。

  `arn:`*`Amazon_partition_name`*`:kms:`*`Amazon_Region`*`:`*`Amazon_account_ID`*`:key/`*`key_ID`*

  例如：

  arn:aws:kms:us-west-2:111122223333:key/1234abcd-12ab-34cd-56ef-1234567890ab

  别名

  当资源是别名时，请使用其别名 ARN。有关帮助信息，请参阅 the section called “查找别名和别名 ARN”。

  `arn:`*`Amazon_partition_name`*`:kms:`*`Amazon_region`*`:`*`Amazon_account_ID`*`:alias/`*`alias_na`

  例如：

  arn:aws:kms:us-west-2:111122223333:alias/ExampleAlias

***（星号）**

当权限不适用于特定资源（KMS 密钥或别名）时，请使用星号 (*)。

在 Amazon KMS 权限的 IAM policy 中，`Resource` 元素中的星号表示所有 Amazon KMS 资源（KMS 密钥和别名）。当 Amazon KMS 权限不适用于任何特定的 KMS 密钥或别名时，您也可以在 `Resource` 元素中使用星号。例如，允许或拒绝 `kms:CreateKey` 或 `kms:ListKeys` 权限时，必须将 `Resource` 元素设置为 `*`。

- Amazon KMS 条件键列出可用于控制对操作的访问的 Amazon KMS 条件键。您可以在策略的 `Condition` 元素中指定条件。有关更多信息，请参阅 Amazon KMS 条件键。此列还包含 Amazon 全局条件上下文键，这些键受 Amazon KMS 支持但并不受所有 Amazon 服务支持。

Amazon KMS 内部操作

Amazon Key Management Service（Amazon KMS）提供的密钥和密码操作由可针对云进行扩展的 FIPS 140-3 安全性 3 级认证硬件安全模块（HSM）保护。Amazon KMS 密钥和功能供多项 Amazon 云服务使用，也可以用于保护您应用程序中的数据。本技术指南提供有关使用 Amazon KMS 时在 Amazon 内运行的加密操作的详细信息。

Amazon KMS扩展和保护 HSM 需要 内部构件以实现全球分布式密钥管理服务。

主题

- 域和域状态
- 内部通信安全
- 多区域密钥的复制过程
- 持久性保护

域和域状态

Amazon Web Services 区域 内受信任的 Amazon KMS 实体的协作集合称为域。域包括一组受信任的实体、一组规则和一组机密密钥（称为域密钥）。域密钥在作为域成员的 HSM 之间共享。域状态包括以下字段。

名称

用于标识此域的域名。

成员

作为域成员的 HSM 列表，包括其公有签名密钥和公有协议密钥。

运算符

实体、公有签名密钥和代表此服务运营商的角色（Amazon KMS 运营商或服务主机）的列表。

规则

在 HSM 上运行的每条命令必须满足的仲裁规则列表。

域密钥

域中当前正在使用的域密钥（对称密钥）列表。

完整域状态仅在 HSM 上可用。域状态作为导出的域令牌在 HSM 域成员之间同步。

域密钥

域中的所有 HSM 均共享一组域密钥 $\{DK_r\}$。这些密钥通过域状态导出例程共享。导出的域状态可以导入作为域成员的任何 HSM 中。

域密钥 $\{DK_r\}$ 组始终包含一个活动域密钥和多个停用的域密钥。域密钥每天轮换，以确保 Amazon 符合 Recommendation for Key Management - Part 1 的要求。域密钥轮换期间，在传出域密钥下加密的所有现有 KMS 密钥将在新的活动域密钥下重新加密。活动域密钥用于加密任何新 EKT。过期的域密钥只能用于解密以前加密的 EKT，其天数相当于最近轮换的域密钥的数量。

导出的域令牌

我们经常需要在域参与者之间同步状态。这可以通过在对域进行更改时导出域状态来实现。域状态将导出为导出的域令牌。

名称

用于标识此域的域名。

成员

作为域成员的 HSM 列表，包括其签名和协议公有密钥。

运算符

实体、公有签名密钥和代表此服务运营商的角色的列表。

规则

在 HSM 域成员上运行的每条命令必须满足的仲裁规则列表。

加密的域密钥

信封加密的域密钥。域密钥通过上面列出的每个成员的签名成员进行加密，然后封装到其公有协议密钥中。

Signature

由 HSM 生成的域状态签名，必须是导出域状态的域成员。

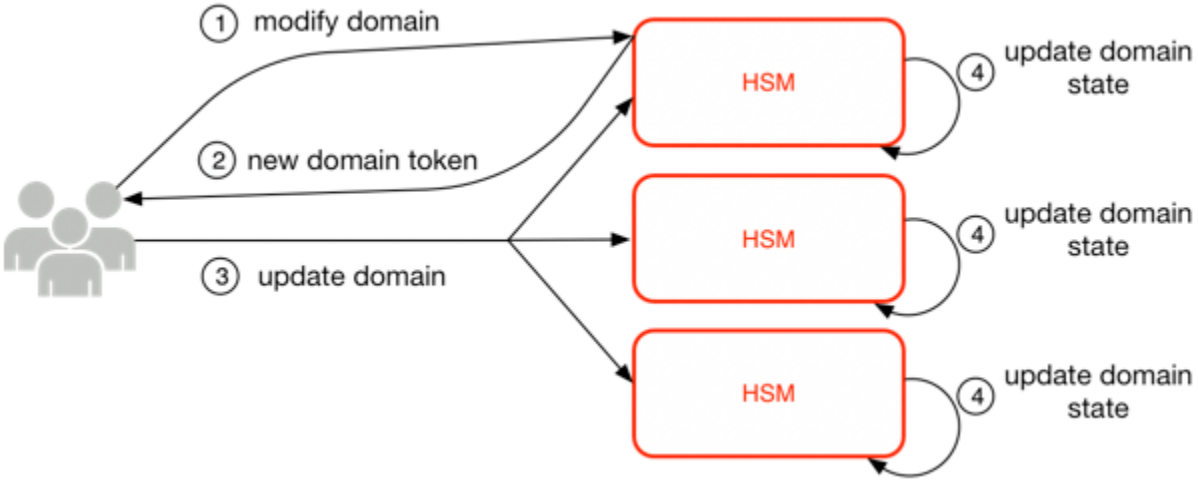
导出的域令牌构成域内运行的实体的基本信任源。

管理域状态

域状态通过经仲裁身份验证的命令进行管理。这些更改包括修改域中受信任参与者的列表、修改用于运行 HSM 命令的仲裁规则以及定期轮换域密钥。这些命令将按每条命令进行身份验证，而不是经过身份验证的会话操作，如下面的图像所示。

处于初始化和运行状态的 HSM 包含一组自行生成的非对称身份密钥、一个签名密钥对和一个密钥建立密钥对。通过手动过程，Amazon KMS 运营商可以建立要在区域中第一个 HSM 上创建的初始域。此初始域包含完整的域状态，如本主题中之前所定义。它通过连接命令安装到域中定义的每个 HSM 成员。

HSM 加入初始域后，它将绑定到该域中定义的规则。这些规则管理使用客户加密密钥或者更改主机或域状态的命令。使用加密密钥的经过身份验证的会话 API 操作在之前已定义。



上图描述了如何修改域状态。该过程包括四个步骤：

1. 向 HSM 发送基于仲裁的命令以修改域。
2. 将生成新的域状态，并将其导出为新的导出域令牌。HSM 上的状态未修改，这意味着更改未在 HSM 上实施。
3. 向新导出的域令牌中的每个 HSM 发送第二条命令，以使用新的域令牌更新其域状态。
4. 新导出的域令牌中列出的 HSM 可以对命令和域令牌进行身份验证。它们还可以解压缩域密钥，以更新域中所有 HSM 上的域状态。

HSM 彼此之间不直接通信。相反，一定数量的运营商会请求更改域状态，从而生成新的导出域令牌。域的服务主机成员用于将新的域状态分发给域中的每个 HSM。

域的离开和加入通过 HSM 管理功能完成。域状态的修改通过域管理功能完成。

离开域

使 HSM 离开域，从内存中删除该域的所有剩余部分和密钥。

加入域

使 HSM 加入新域或将其当前域状态更新为新域状态。现有域用作对此消息进行身份验证的初始规则集的来源。

创建域

导致在 HSM 上创建新域。返回可分发给域的成员 HSM 的第一个域令牌。

修改运营商

从域中授权运营商及其角色的列表中添加或删除运营商。

修改成员

从域中授权 HSM 的列表中添加或删除 HSM。

修改规则

修改在 HSM 上运行命令所需的仲裁规则集。

轮换域密钥

导致创建新的域密钥并将其标记为活动域密钥。这会将现有活动密钥移动到已停用的密钥，并从域状态中删除最旧的已停用密钥。

内部通信安全

服务主机或 Amazon KMS 运营商与 HSM 之间的命令通过 经身份验证的会话 中描述的两种机制进行保护：仲裁签名请求方法和使用 HSM 服务主机协议的经身份验证会话。

仲裁签名命令旨在让任何单个运营商都无法修改 HSM 提供的重要安全保护。在经过身份验证的会话中运行的命令可帮助确保只有授权的服务运营商才能执行涉及 KMS 密钥的操作。所有客户绑定的机密信息均可跨 Amazon 基础设施进行保护。

密钥建立

为了保护内部通信，Amazon KMS 使用两种不同的密钥建立方法。第一种方法定义为使用离散对数密码学的成对密钥建立方案建议中的 C(1, 2, ECC DH)。此方案有一个带静态签名密钥的启动程序。启动程序会生成一个临时椭圆曲线 Diffie-Hellman (ECDH) 密钥并签名，旨在用于具有静态 ECDH 协议密钥的收件人。此方法使用一个临时密钥和两个使用 ECDH 的静态密钥。这是标签 C(1, 2, ECC DH) 的衍生。此方法有时称为一次性 ECDH。

第二种密钥建立方法是 C(2, 2, ECC, DH)。在此方案中，双方都有一个静态签名密钥，然后会生成、签名和交换临时 ECDH 密钥。此方法使用两个静态密钥和两个临时密钥（各自使用 ECDH）。这是标签 C(2, 2, ECC, DH) 的衍生。此方法有时称为临时 ECDH 或 ECDHE。所有 ECDH 密钥均在曲线 secp384r1 (NIST-P384) 上生成。

HSM 安全边界

Amazon KMS 的内部安全边界是 HSM。HSM 具有专有接口，在其处于运行状态时没有其他活动物理接口。运行的 HSM 在初始化期间使用必要的加密密钥进行预置，从而在域中建立其角色。HSM 的敏感加密材料仅存储在易失性存储器中，并在 HSM 退出运行状态（包括预期或非预期关机或重置）时擦除。

HSM API 操作可以通过单个命令进行身份验证，也可以通过服务主机建立的相互认证的机密会话进行身份验证。

仲裁签名命令

仲裁签名命令由运营商发给 HSM。本节介绍如何创建、签名和验证基于仲裁的命令。这些规则相当简单。例如，命令 Foo 需要对角色 Bar 的两名成员进行身份验证。创建和验证基于仲裁的命令有三个步骤。第一步是初始命令创建；第二步是提交给要签名的其他运营商；第三步是验证和执行。

为介绍这些概念，假设有一组真实的运营商公有密钥和角色 $\{QOS_s\}$，以及一组仲裁规则 QR = $\{Command_i, Rule_{\{i,\ t\}}\}$，其中每个 Rule 均为一组角色，且最小数字为 N $\{Role_t, N_t\}$。为使命令满足仲裁

规则，命令数据集必须由 {QOS_s} 中列出的一组运营商进行签名，以使其满足该命令列出的规则之一。如前所述，仲裁规则和运营商组存储在域状态和导出的域令牌中。

实际上，初始签名者会将命令 Sig_1 = Sign(dO_{p1}, Command) 签名。第二个运营商也会将命令 Sig_2 = Sign(dO_{p2}, Command) 签名。双重签名的消息将发送给 HSM 执行。HSM 将执行以下操作：

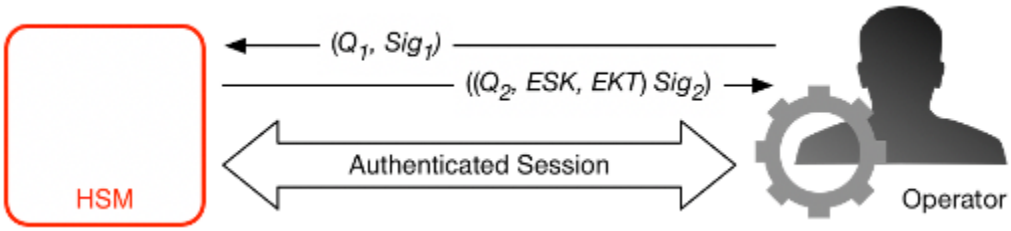
1. 对于每个签名，它会从域状态中提取签名者的公有密钥，并验证命令中的签名。
2. 它会验证该组签名者是否满足命令规则。

经身份验证的会话

您的密钥操作在面向外部的 Amazon KMS 主机和 HSM 之间运行。这些命令与加密密钥的创建和使用以及安全随机数生成有关。这些命令在服务主机与 HSM 之间的会话验证的通道上运行。除了需要真实性以外，这些会话还需要机密性。这些会话上运行的命令包括返回明文数据密钥和为您解密的消息。要确保这些会话不会因中间人攻击而被破坏，请将会话进行身份验证。

此协议在 HSM 与服务主机之间执行经过相互身份验证的 ECDHE 密钥协议。交换由服务主机发起，并由 HSM 完成。HSM 还返回通过协商密钥加密的会话密钥 (SK) 和包含会话密钥的导出密钥令牌。导出的密钥令牌包含一个有效期，经过该期限后服务主机必须重新协商会话密钥。

服务主机是域成员，具有身份签名密钥对（$dHOS_i$，$QHOS_i$）和 HSM 身份公有密钥的真实副本。它使用其身份签名密钥组来安全地协商会话密钥，该密钥可在服务主机与域中的任何 HSM 之间使用。导出的密钥令牌有一个与其关联的有效期，经过该期限后必须协商一个新密钥。



该过程从服务主机识别开始，它需要会话密钥才能在自身和域 HSM 成员之间发送和接收敏感通信流。

1. 服务主机会生成 ECDH 临时密钥对 (d_1, Q_1)，并使用其身份密钥 Sig1 = Sign(dOS,Q_1) 进行签名。
2. HSM 使用其当前域令牌验证收到的公有密钥的签名，然后创建 ECDH 临时密钥对 (d_2, Q_2)。然后根据使用离散对数密码学的成对密钥建立方案建议完成 ECDH 密钥交换，以形成协商的 256 位 AES-GCM 密钥。HSM 会生成新的 256 位 AES-GCM 会话密钥。它使用协商密钥来加密会话密钥，从而形成加密的会话密钥 (ESK)。它还加密域密钥下的会话密钥作为导出的密钥令牌 EKT。最后，它使用其身份密钥对 Sig_2 = Sign(dHSK, (Q_2, ESK, EKT)) 将返回值签名。

3. 服务主机使用其当前域令牌验证收到的密钥的签名。然后，服务主机会根据使用离散对数密码学的成对密钥建立方案建议完成 ECDH 密钥交换。接下来，它会解密 ESK 以获取会话密钥 SK。

在 EKT 的有效期内，服务主机可使用协商的会话密钥 SK 向 HSM 发送信封加密的命令。通过此身份验证会话的每个服务主机发起的命令都包含 EKT。HSM 使用相同的协商会话密钥 SK 进行响应。

多区域密钥的复制过程

Amazon KMS 使用跨区域复制机制将 KMS 密钥中的密钥材料从位于某个 Amazon Web Services 区域的 HSM 复制到位于另一个 Amazon Web Services 区域 的 HSM。要使此机制工作，要复制的 KMS 密钥必须是多区域密钥。将 KMS 密钥从某个区域复制到另一个区域时，这些区域中的 HSM 无法直接通信，因为它们位于隔离的网络中。相反，在跨区域复制期间交换的消息将由代理服务传送。

在跨区域复制期间，由 Amazon KMS HSM 生成的每条消息都会使用复制签名密钥进行加密签名。复制签名密钥 (RSK) 是 NIST P-384 曲线上的 ECDSA 密钥。每个区域至少具有一个 RSK，并且每个 RSK 的公有组件都与同一 Amazon 分区中的每个其他区域共享。

将密钥材料从区域 A 复制到区域 B 的跨区域复制过程如下：

1. 区域 B 中的 HSM 在 NIST P-384 曲线上生成一个临时的 ECDH 密钥，即复制协议密钥 B (RAKB)。RAKB 的公有组件由代理服务发送到区域 A 中的 HSM。
2. 区域 A 中的 HSM 将接收 RAKB 的公有组件，然后在 NIST P-384 曲线上生成另一个临时的 ECDH 密钥，即复制协议密钥 A (RAKA)。HSM 将在 RAKA 和 RAKB 的公有组件上运行 ECDH 密钥建立方案，并从输出中派生一个对称密钥，即复制包装密钥 (RWK))。RWK 用于对即将复制的多区域 KMS 密钥的密钥材料进行加密。
3. RAKA 的公有组件和使用 RWK 加密的密钥材料将通过代理服务发送到区域 B 中的 HSM。
4. 区域 B 中的 HSM 将接收 RAKA 的公有组件和使用 RWK 加密的密钥材料。通过在 RAKB 和 RAKA 的公有组件上运行 ECDH 密钥建立方案，RWK 将派生 HSM。
5. 区域 B 中的 HSM 将使用 RWK 解密来自区域 A 的密钥材料。

持久性保护

通过使用离线 HSM、导出域令牌的多个非易失性存储以及加密 KMS 密钥的冗余存储，为服务生成的密钥提供额外的服务持久性。离线 HSM 是现有域的成员。除了不在线和不参与常规域操作之外，离线 HSM 显示的域状态与现有 HSM 成员完全相同。

持久性设计旨在保护区域中的所有 KMS 密钥，以防 Amazon 遇到在线 HSM 或我们的主存储系统内存储的成组 KMS 密钥大规模丢失的情况。采用导入密钥材料的 Amazon KMS keys 不包含在其他 KMS 密钥提供的持久性保护范围内。如果 Amazon KMS 发生区域范围的故障，则可能需要将导入的密钥材料重新导入到 KMS 密钥中。

离线 HSM 及其访问凭证存储在多个独立地理位置受监控安全屋内的保险箱中。每个保险箱至少需要一名 Amazon 安全干事和一名 Amazon KMS 运营商（来自 Amazon 中两个独立的团队）以获得这些材料。这些材料的使用受内部策略（要求存在一定数量 Amazon KMS 运营商）的约束。

文档历史记录

本主题介绍了有关 Amazon Key Management Service 开发人员指南的重要更新。

主题

最近的更新

下表介绍了自 2018 年 1 月起对此文档的一些重要更改。除了此处列出的主要更改以外，我们还会经常更新文档，以改进说明和示例以及处理您发送给我们的反馈意见。要获得有关重要更改的通知，请订阅 RSS 源。

您可能需要水平或垂直滚动才能查看此表中的所有数据。

变更	说明	日期
功能更新	Amazon KMS 支持使用导入的密钥材料按需轮换对称加密、多区域密钥。	2025 年 11 月 26 日
功能更新	Amazon KMS 支持 CloudTrail 事件`tlsDetails` 部分的新`keyExchange` 字段。	2025年11月24日
功能更新	Amazon KMS 支持新的 KeySpec ECC EDWARDS25519 _NIST_。	2025 年 11 月 7 日
双栈支持	Amazon KMS 支持双堆栈。	2025 年 6 月 18 日
功能更新	增加支持用于后量子密码签名的模格数字签名算法（ML-DSA）。	2025 年 6 月 13 日

导入的密钥轮换	您可以对具有导入的密钥材料（来源为 EXTERNAL）的对称加密 KMS 密钥执行按需轮换。	2025 年 6 月 5 日
功能更新	在中国区域增加支持多区域 KMS 密钥。	2024 年 11 月 21 日
Amazon 托管策略更新	通过在策略版本 v2 的托管策略中添加语句 ID (Sid) 来更新AWSKeyManagementServiceMultiRegionKeysServiceRolePolicy服务相关角色。	2024 年 11 月 21 日
配额更改	提高了ImportKeyMaterial和DeleteImportedKeyMaterial请求的默认请求速率。	2024 年 7 月 23 日
配额更改	提高了对称加密 KMS 密钥、RSA KMS 密钥以及 ECC 和 KMS 密钥的默认加密操作请求速率。 SM2	2024 年 7 月 8 日
新特征	KEY_AGREEMENT 为 NIST 推荐的椭圆曲线 (ECC) 和（仅限 SM2 中国区域）KMS 密钥添加了新KeyUsage类型，并增加了对派生共享密钥的支持。	2024 年 6 月 13 日
密钥轮换更新	增加了对自定义轮换周期的支持，包括自动密钥轮换、按需密钥轮换以及对密钥材料轮换的可见性。	2024 年 4 月 12 日

托管式策略的更新	添加了新的权限AWSKeyManagementServiceCustomKeyStoresServiceRolePolicy ，Amazon KMS 允许监控包含您的 Amazon CloudHSM 集群的 VPC 中的更改，以便在出现故障时 Amazon KMS 可以提供清晰的错误消息。	2023 年 11 月 10 日
功能更新	添加了对 DryRun API 参数的支持。	2023 年 7 月 5 日
功能更新	增加了对导入所有类型密钥材料的支持，自定义 Amazon KMS 密钥库除外。	2023 年 6 月 5 日
功能更新	Nitro Amazon KMS APIs Enclaves 的更新	2023 年 3 月 10 日
功能更新	RSAES_PKCS1_V1_5 包装算法已被弃用。 Amazon KMS 根据美国国家标准与技术研究院 (NIST) 的加密密钥管理指南，将在 2023 年 10 月 1 日RSAES_PKCS1_V1_5 之前终止所有支持。我们建议您立即开始使用不同的包装算法。	2023 年 2 月 28 日
功能更新	增加了对外部密钥存储的支持，该功能允许您使用外部的加密密钥来保护您的 Amazon 资源。 Amazon	2022 年 11 月 29 日
配额更改	将每个账户和地区的 Amazon KMS keys 资源配额增加到 100,000 个 KMS 密钥。	2022 年 7 月 8 日

功能更新	在更多内容中增加了对 HMAC KMS 密钥的支持 Amazon Web Services 区域	2022 年 7 月 8 日
新主题	在《 Amazon KMS 开发者指南》的 “安全” 一章中添加了 Amazon Key Management Service 主题中的弹性。	2022 年 6 月 14 日
新特征	增加了对生成和验证 HMAC 代码的 Amazon KMS 密钥和 API 操作的支持。	2022 年 4 月 19 日
文档更改	将术语客户主密钥 (CMK) 替换为 Amazon KMS key 和 KMS 密钥。	2021 年 8 月 30 日
新功能	添加了对多区域密钥的支持，这是一组具有相同密钥 ID 和密钥材料的不同区域中的可互操作 KMS 密钥。您可以使用多区域密钥加密一个区域中的数据，并解密不同区域中的数据。	2021 年 6 月 8 日
新功能	添加了对基于属性的访问控制 (ABAC) 的支持。您可以使用标签和别名来控制对您的 Amazon KMS keys访问权限。	2020 年 12 月 17 日
新功能	增加了对 VPC 终端节点策略的支持。	2020 年 7 月 9 日
新增内容	解释的安全属性 Amazon KMS。	2020 年 6 月 18 日

新功能	增加了对非对称 Amazon KMS keys 和非对称数据密钥的支持。	2019 年 11 月 25 日
更新功能	您可以在 Amazon KMS 控制台 Amazon 托管式密钥 中查看的密钥策略。此功能过去仅限于客户托管密钥。	2019 年 11 月 15 日
新功能	说明如何在 TLS 中使用混合后量子密钥交换算法来调用 Amazon KMS。	2019 年 11 月 4 日
配额更改	增加了某些管理 KMS 密钥 APIs 的公司的资源配额。	2019 年 9 月 18 日
配额更改	更改了 KMS 密钥、别名和每个 KMS 密钥的授权数的资源配额。	2019 年 3 月 27 日
配额更改	更改了使用自定义密钥存储中的 Amazon KMS keys 的加密操作的每秒请求配额。	2019 年 3 月 7 日
新功能	说明如何创建和管理 Amazon KMS 自定义密钥库。每个密钥库都由您拥有和控制的 Amazon CloudHSM 集群提供支持。	2018 年 11 月 26 日
新控制台	说明如何使用独立于 IAM Amazon KMS 控制台的新控制台。原始控制台及其使用说明将在短时间内保持可用状态，以便您有时间熟悉新控制台。	2018 年 11 月 7 日
配额更改	已更改共享请求配额以供使用 Amazon KMS keys。	2018 年 8 月 21 日

新增内容	说明如何 Amazon Secrets Manager 使用 Amazon KMS密钥加密密钥中的密钥值。	2018 年 7 月 13 日
新增内容	说明 DynamoDB 如何 Amazon KMS Amazon KMS keys 使用来支持其服务器端加密选项。	2018 年 5 月 23 日
新功能	说明如何使用您的 VPC 中的私有终端节点直接连接 Amazon KMS，而不是通过互联网进行连接。	2018 年 1 月 22 日

早期更新

下表描述了 2018 年之前对《 Amazon Key Management Service 开发者指南》所做的重要更改。

您可能需要水平或垂直滚动才能查看此表中的所有数据。

更改	描述	日期
新增内容	添加了有关标签在 Amazon KMS的文档。	2017 年 2 月 15 日
新增内容	添加了有关 监控 Amazon KMS keys 和 使用 Amazon CloudWatch 监控 KMS 密钥的文档。	2016 年 8 月 31 日
新增内容	添加了有关导入的密钥材料的文档。	2016 年 8 月 11 日
新增内容	添加了以下文档：IAM 策略、权限参考 和 条件键。	2016 年 7 月 5 日
更新	更新了 KMS 密钥访问权限和权限 一章中的文档部分。	2016 年 7 月 5 日

更改	描述	日期
更新	更新了限额页面，在其中说明新的默认配额。	2016 年 5 月 31 日
更新	更新了 限额 页面，在其中说明新的默认配额，并更新了授权令牌文档以提高清晰性和准确性。	2016 年 4 月 11 日
新增内容	添加了有关 允许多个 IAM 主体访问 KMS 密钥 和 使用 IP 地址条件 的文档。	2016 年 2 月 17 日
更新	更新了 中的关键政策 Amazon KMS 和 更改密钥策略 页面以提高清晰性和准确性。	2016 年 2 月 17 日
更新	更新了管理 KMS 密钥主题页面以提高清晰性。	2016 年 1 月 5 日
新增内容	添加了相关的文档 CloudTrail。	2015 年 11 月 18 日
新增内容	添加了有关 更改密钥策略 的说明。	2015 年 11 月 18 日
更新	更新了有关 Amazon Relational Database 的使用方式的文档 Amazon KMS。	2015 年 11 月 18 日
新增内容	添加了有关 Amazon 的文档 WorkSpaces。	2015 年 11 月 6 日
更新	更新了 中的关键政策 Amazon KMS 页面以提高清晰性。	2015 年 10 月 22 日

更改	描述	日期
新增内容	增加了有关 删除 Amazon KMS key 的文档，包括有关 创建警报 和 确定 KMS 密钥的过去使用情况 的支持文档。	2015 年 10 月 15 日
新增内容	添加了有关确定对 Amazon KMS keys 的访问权限的文档。	2015 年 10 月 15 日
新增内容	添加了有关Amazon KMS 密钥的密钥状态的文档。	2015 年 10 月 15 日
新增内容	添加了有关 Amazon Simple Email Service 的文档。	2015 年 10 月 1 日
更新	更新了 限额 页面，在其中解释新的请求配额。	2015 年 8 月 31 日
新增内容	添加了有关使用费用的信息 Amazon KMS。请参阅 Amazon KMS 定价。	2015 年 8 月 14 日
新增内容	向中添加了请求配额 Amazon KMS 限额。	2015 年 6 月 11 日
新增内容	添加了演示 `UpdateAlias` 操作用法的新 Java 代码示例。	2015 年 6 月 1 日
更新	将 Amazon Key Management Service 区域表移至 Amazon Web Services 一般参考。	2015 年 5 月 29 日
新增内容	添加了有关Amazon EMR 如何使用 Amazon KMS的文档。	2015 年 1 月 28 日
新增内容	添加了有关 Amazon 的文档 WorkMail。	2015 年 1 月 28 日

更改	描述	日期
新增内容	添加了有关 Amazon Relational Database 如何使用的文档 Amazon KMS。	2015 年 1 月 6 日
新增内容	添加了有关 Amazon Elastic Transcoder 的文档。	2014 年 11 月 24 日
新指南	介绍了 Amazon Key Management Service 开发人员指南。	2014 年 11 月 12 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。