# 第 5 章

# 数据安全管理

本章将深入探讨如何在 AWS 环境中保护数据的安全。数据是任何应用的核心，无论是用户的个人信息，还是企业的重要数据都需要采取有效的措施来保护。在 AWS 中，有多种工具和服务可以帮助实现这一目标。

**本章要点：**

(1) Amazon S3 的保护。

(2) 数据库的保护(包括 RDS 和 DynamoDB)。

(3) EBS 卷的保护。

(4) 数据备份和恢复。

(5) Amazon Macie 服务。

## 5.1 Amazon S3 的保护

Amazon S3 是一种对象存储服务，可以随时随地存储和检索任意数量和大小的数据。为了保护 S3 中的数据，Amazon 提供了数据加密、访问控制和备份恢复等安全功能。这些功能确保了数据的保密性、防止未经授权的访问，并提供了数据持久性和可用性的保障。

### 5.1.1 加密过程概述

加密是确保数据隐私的基本要求之一，特别是对于跨网络传输的数据，需要进行端到端的保护。S3 提供了两种加密方法：传输加密和静态加密。通过客户端的 SSL/TLS 及存储桶级别的策略，可以实现 S3 的传输加密。静态加密可以进一步分为客户端加密和服务器端加密。在客户端加密中，数据在发送到 S3 存储桶之前会被加密，而在服务器端加密中，数据在发送到 S3 存储桶之后和存储到 S3 存储桶之前都会被加密。

简单回顾一下数据加密的基本流程。首先，需要一个由软件或硬件生成的对称式数据密钥。在加密大量数据或较长文件流时，对称式密钥的效率要优于非对称式密钥。对称数据密钥应用于要加密的数据，并通过加密算法(如 AES)进行处理。处理后的结果是密文，这与随机数据几乎无异，然后将加密后的数据存储在某个位置，无论是在 AWS 上还是在

本地。

如果想要解密数据，就必须使用数据密钥。那么，如何将此密钥提供给有权访问和解密数据的人呢？需要将这个对称数据密钥存储在某个位置，但是不能直接将其与加密后的数据一起存储，因为那样就无法达到保护数据的目的。最佳实践是使用另一个密钥（例如主密钥）加密对称数据密钥，然后可以将加密后的数据与加密后的数据密钥一起存储。这样，加密后的数据和加密后的数据密钥就可以存储在同一位置，具有相同的持久性特征。

当为 AWS 云中的数据选择适合的加密解决方案时，应考虑以下 3 个问题：

（1）密钥应该存储在哪里？是存储在自己的硬件存储中，还是使用 AWS 提供的硬件存储？

（2）密钥应该在哪里使用？是由客户端软件使用，还是在 AWS 上使用？

（3）谁来管理密钥？是分配用户级或应用程序级权限，还是让 AWS 管理权限？

对于 S3 来讲，它使用对称加密来加密数据。当上传一个对象并请求加密时，S3 会生成一个唯一的对称密钥，用这个密钥对数据进行加密。同时，S3 还会使用 AWS Key Management Service（KMS）中的客户主密钥（CMK）来加密这个对称密钥。这个过程被称为“信封加密”（Envelope Encryption），然后 S3 会将加密后的对称密钥一起存储在 S3 中。这样，即使有人能够访问存储在 S3 中的加密后的数据和加密后的对称密钥，他们也无法解密数据，因为他们没有访问 AWS KMS 中的 CMK 的权限，如图 5-1 所示。

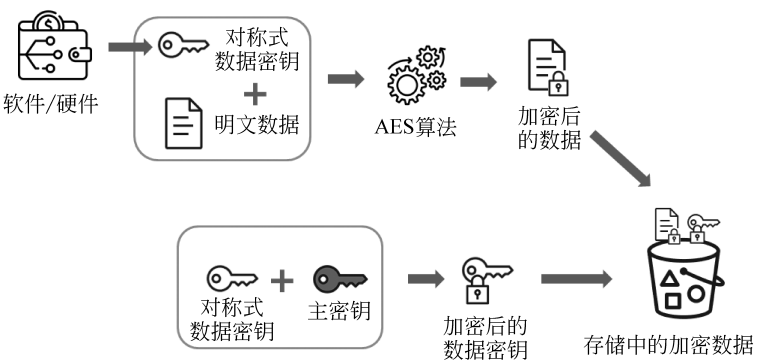


图 5-1　AWS S3 服务器端加密的简化流程

根据安全需求，可以选择使用服务器端加密或客户端加密来加密 AWS 中的数据。每种方法都有其自己的优势。如果需要，则可以选择同时使用这两种方法。

通过客户端加密（Client-Side Encryption，CSE），可以创建并管理自己的加密密钥，这个密钥不会以明文形式被导出到 AWS。应用程序在将数据提交到 AWS 之前会对其进行加密的，并在从 AWS 接收到数据后对其进行解密。数据以加密的形式存储，而所采用的密钥和算法只有自己知道。对 CSE 来讲，数据是在提交到 AWS 之前进行加密的，在从 AWS 中检索后进行解密。指定的加密软件会提供使用的加密密钥。目前，加密客户端可用于 S3、DynamoDB、EMR 文件系统和 AWS 加密软件开发工具包。AWS 加密软件开发工具包是

一个加密库，有助于开发人员更轻松地在其应用程序中实施加密最佳实践。它使开发人员能够专注于开发其应用程序的核心功能，而不是如何最好地加密和解密数据。

加密客户端可以使用本地密钥管理基础设施（Key Management Infrastructure，KMI）提供的密钥，这将提供客户端主密钥（CSE-C）。CSE-C 和未加密的数据绝不会被发送到 AWS。务必安全地管理加密密钥。如果丢失了加密密钥，则将无法解密数据。在具有加密客户端的 EC2 实例上运行的应用程序也可以请求 CSE-C。在 EC2 实例上运行的 KMI 也可以提供密钥。

对于服务器端加密（Server-Side Encryption，SSE），系统会在服务收到 API 调用之后为数据进行加密。SSE 对最终用户是透明的，AWS 会定期轮换主密钥。例如，如果将 SEE 与 S3 一起使用，则可以请求 S3 在将对象保存到数据中心的磁盘之前加密对象，并在下载对象时进行解密。

在服务器端加密中，数据在发送到 S3 存储桶之后和存储到 S3 存储桶之前都会被加密。

**提示**：AWS S3 的服务器端加密与 Windows 操作系统的 NTFS 文件系统的加密原理类似。

## 5.1.2　Amazon S3 服务器端加密


服务器端加密（SSE）是指接收数据的应用程序或服务在目标位置对数据进行加密。SSE 的过程对使用者来讲是“透明”的，S3 会在将数据写入 AWS 数据中心内的磁盘时对这些数据进行对象级别的加密，并在访问这些数据时解密。只要 AWS 验证了请求并且拥有访问权限，访问加密和未加密对象的方式就没有区别。S3 服务器端加密使用 256 位高级加密标准 Galois/Counter 模式（AES-GCM）对所有上传的对象进行加密。

在默认情况下，所有 S3 桶都配置了加密，所有上传到 S3 桶的新对象都会自动静态加密。创建存储桶时，默认加密配置是使用 S3 托管密钥进行的服务器端加密（SSE-S3），如图 5-2 所示。可以修改存储桶的默认加密配置，也可以在 S3 PUT 请求中设置服务器端加密类型。在一个存储桶中，不同的对象可以采用不同的加密类型。


图 5-2　创建新存储桶时的默认加密选项

对于服务器端加密，AWS 提供了 4 个互斥的选项，具体取决于选择如何管理加密密钥

和要应用的加密层数。

**1. SSE-S3（使用 Amazon S3 托管密钥进行的服务器端加密）**

这是服务器端加密的默认选项。存储桶中每个对象都使用唯一的密钥进行加密，这个密钥被称为“数据加密密钥”。作为额外的保护措施，SSE-S3 使用定期轮换的根密钥（Root Key）来加密数据和加密密钥。这就是所谓的“信封加密”，它提供了额外的安全层，因为即使有人获取了加密的数据和加密的数据密钥，他们也无法解密数据，除非他们还有访问根密钥的权限。SSE-S3 是最简单易用的加密模式。

SSE-S3 没有额外费用，但是，配置和使用 SSE-S3 的请求会产生标准的 S3 请求费用。

**2. SSE-C（使用客户提供的密钥进行服务器端加密）**

在使用 SSE-C 时，客户管理加密密钥，而 S3 管理加密和解密。这适用于希望自己管理加密密钥的场景。可以设置自己的加密密钥，并且在操作的时候作为请求的一部分提供给 S3，S3 将数据写入磁盘时进行加密，加密结束后会从内存中删除此加密密钥。当访问此对象时，必须提供相同的加密密钥作为请求的一部分。S3 在将对象数据返回之前会首先验证提供的加密密钥是否匹配，然后解密对象，但是需要注意，如果丢失了加密密钥，则将无法访问使用该密钥加密的对象，因此，在使用 SSE-C 时，密钥管理是非常重要的。

AWS 管理控制台不提供在上传对象和管理对象时使用 SSE-C 选项，只能使用 REST API、SDK 来指定 SSE-C。

SSE-C 没有额外费用，但是，配置和使用 SSE-C 的请求会产生标准的 S3 请求费用。

**3. SSE-KMS（使用 AWS KMS 密钥的服务器端加密）**

SSE-KMS 是通过将 AWS KMS 服务与 S3 集成来提供的。KMS 是一项服务，可将安全、高度可用的硬件和软件结合起来，以提供可扩展到云的密钥管理系统。使用 KMS，可以更好地控制密钥。例如，可以查看单独的密钥、编辑控制策略及遵循 AWS CloudTrail 中的密钥。此外，还可以创建和管理客户自主管理型密钥，或者使用对于服务和区域为唯一的 AWS 托管式密钥。

当 SSE-KMS 加密一个对象时，S3 会生成一个用于验证数据完整性的校验和，也会对其进行加密。这个加密后的校验和会作为对象的元数据（描述对象的数据）的一部分存储起来。这样，即使有人能够访问这个校验和，也无法读取它的内容，除非他们有解密这个校验和的密钥。

使用 AWS KMS，用户可以更好地控制自己的密钥。这需要理解 AWS KMS 的工作原理和如何与 S3 集成，因此相比 SSE-S3、SSE-C 要复杂一些。

当使用 SSE-KMS 时，既可以使用默认的 AWS 托管式密钥，也可以指定已创建的客户托管式密钥。使用客户托管密钥可提供更大的灵活性和控制力。例如，可以创建、轮换和禁用客户托管密钥。还可以定义访问控制和审核，用于保护数据的客户托管密钥。

如果选择使用 SSE-KMS 加密数据，则 KMS 和 S3 将执行以下信封加密操作，如图 5-3 所示。

（1）S3 首先向 KMS 发出请求，请求用于对称式加密的数据密钥及使用客户主密钥

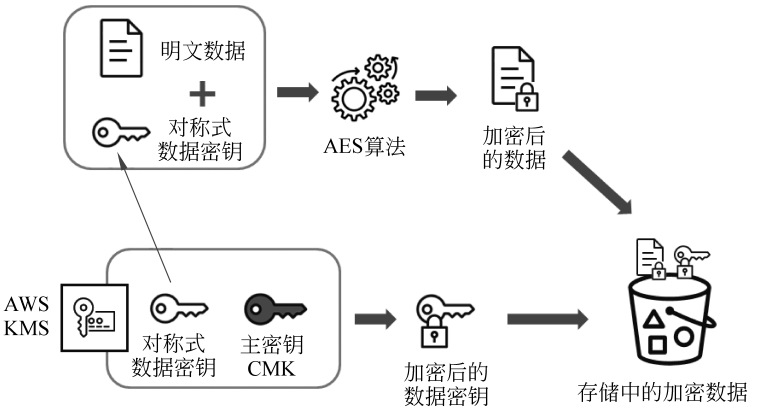


图 5-3　SSE-KMS 的信封加密

(Customer Master Key,CMK)加密后的数据密钥。

(2) KMS 生成明文的数据加密密钥,然后使用 CMK 对其进行加密。KMS 会将明文数据密钥和加密后的数据密钥发送给 S3。

(3) S3 使用数据密钥加密数据,并在使用后尽快从内存中删除该明文密钥。

(4) S3 将加密的数据密钥作为元数据与加密数据一起存储。

**注意**:CMK 永远不会离开 KMS。

当请求解密数据时,S3 和 KMS 将执行以下操作:

(1) S3 从对象的元数据中获取加密后的数据密钥,然后在 Decrypt 请求中将其发送给 AWS KMS。

(2) KMS 使用相同的 CMK 对加密的数据密钥进行解密,然后将得到的明文数据密钥返给 S3。

(3) S3 使用明文数据密钥对已加密的数据进行解密,并在使用后尽快从内存中删除该明文数据密钥。

SSE-KMS 本身没有额外费用,但访问 AWS KMS 是需要付费的。访问使用 SSE-KMS 加密的数百万或数十亿个对象的工作负载可能会产生大量到 KMS 的请求。S3 会为每个对象使用单独的 KMS 数据密钥。在这种情况下,每次对 KMS 加密的对象发出请求时,S3 都会调用 AWS KMS。

AWS 提供了 S3 存储桶密钥功能,可以降低使用 KMS 密钥的 SSE-KMS 的成本。使用 SSE-KMS 的桶级别密钥可以通过减少从 S3 到 AWS KMS 的请求流量,从而可以将 AWS KMS 请求成本最高降低 99%。在创建新存储桶的时候,默认启用存储桶密钥功能,如图 5-4 所示。

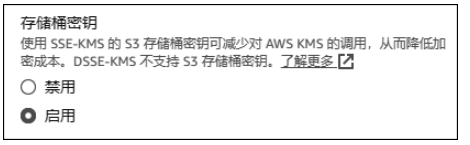


图 5-4　创建存储时存储桶密钥选项

启用 S3 桶密钥功能后,AWS 会从 KMS 生成生存期较短的桶级密钥,然后暂时将其保留在

S3 中。此桶级密钥将在新对象的生命周期中为其创建数据密钥，如图 5-5 所示。S3 桶密钥在 S3 内限时使用，从而减少了 S3 向 KMS 发出请求以完成加密操作的需求。这样可以减少从 S3 到 KMS 的流量，从而可以将 AWS KMS 请求成本最高降低 99%。

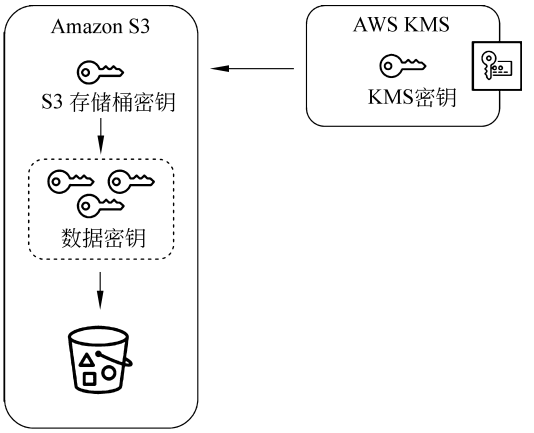


图 5-5　存储桶密钥的工作原理

#### 4. DSSE-KMS（使用 AWS KMS 密钥的双层服务器端加密）

DSSE-KMS 是一种相对较新的选项，它与 SSE-KMS 的原理类似，但 DSSE-KMS 采用的是两层单独的对象级加密，而不是一层。借助 DSSE-KMS，可以更轻松地满足需要，对数据应用多层加密并完全控制加密密钥的合规性标准。使用 DSSE-KMS 无须支付额外费用，但需要注意，DSSE-KMS 不支持 S3 桶密钥，因此无法降低 AWS KMS 的请求成本。


### 5.1.3　Amazon S3 资源保护

Amazon S3 资源保护的重要性不言而喻，AWS 提供了多种技术手段实现这一目标。例如，可以通过 S3 访问控制列表和策略来精细地管理谁可以访问 S3 资源，从而防止未经授权的访问。S3 阻止公有访问功能可以防止数据被公开，避免因误操作而导致的数据泄露。S3 版本控制则可以保存、检索和恢复每个对象的所有版本，这对于防止意外删除或修改数据非常有用。S3 的对象锁定功能可以为存储在 S3 中的对象提供另一层保护，防止对象被意外删除或覆盖。通过跨区域复制，可以在不同的地理区域之间复制对象，以增加数据的耐久性和可用性。

#### 1. Amazon S3 访问控制列表和策略

在默认情况下，所有 S3 资源都是私有的。只有资源拥有者才能访问资源。资源拥有者是指创建资源的 AWS 账户。例如，用于创建存储桶和对象的 AWS 账户拥有这些资源。S3 支持用户身份验证，以控制对数据的访问。可以使用各种访问控制机制，如策略和访问控制列表（ACL），选择性地向用户和用户组授予权限。S3 控制台会突出显示公开可访问的存储桶，注明公开可访问性来源，并且还会在存储桶策略或存储桶 ACL 发生的更改将使存储桶公开可访问时发出警告，如图 5-6 所示。


图 5-6　对可公开访问存储桶的警告

S3 支持的访问控制机制类型可分为两组：基于资源和基于用户。在基于资源的组中，有存储桶策略、存储桶 ACL 和对象 ACL。在基于用户的组中，有 IAM 用户策略。每个存储桶和对象都有与其关联的 ACL。可以使用 ACL 向用户组提供存储桶或者对象的读取或写入访问权限。借助 ACL，可以只向其他 AWS 账户授予访问 S3 资源的权限，而不能针对账户下的特定用户。

S3 中的存储桶策略可用于为一个存储桶内的部分或所有对象添加或拒绝权限。策略可以附加到用户、组或 S3 存储桶上，实现对权限的集中管理。通过存储桶策略，可以向 AWS 账户或其他 AWS 账户内的用户授予 S3 资源的访问权限。

对象的 ACL 是管理对存储桶的拥有者未拥有对象的访问的唯一方式。可以只编写一条策略语句，向一个 AWS 账户授予对数百万具有特定键名称前缀的对象的读取权限。例如，授予对以键名称前缀 logs 开头的对象的读取权限，但是，如果访问权限因对象而异，则使用存储桶策略授予对各个对象的权限可能不太实际。此外，存储桶策略还有 20KB 的大小限制。在这种情况下，使用对象 ACL 可能是比较合适的选择。

存储桶的 ACL 唯一建议的应用场景是授予 S3 日志传输组写入权限，以便将访问日志对象写入存储桶。如果希望 S3 将访问日志传输到存储桶，则需要向日志传输组授予对存储桶的写入权限。如果要管理所有 S3 权限的跨账户权限，则存储桶策略是最佳解决方案。可以使用 ACL 授予对其他账户的跨账户权限，但 ACL 仅支持一组有限的权限，它们并不包括所有 S3 权限。通常，可以使用用户策略或存储桶策略来管理权限。

可以选择通过创建用户并向用户（或用户组）附加策略来分别管理权限，或者也可能认为基于资源的策略（如存储桶策略）更适合场景。

**2. Amazon S3 阻止公有访问**

S3 为存储桶和账户提供了阻止公有访问的设置，有助于管理对 S3 资源的公有访问。S3 阻止公有访问的功能将阻止任何试图允许对 S3 存储桶中的数据进行公有访问的设置。可以为单个 S3 存储桶或账户中的所有存储桶配置阻止公有访问的设置。当将阻止公有访问的设置应用于某一账户时，这些设置将适用于全球所有 AWS 区域。需要注意的是，可能无法在每个对象的基础上应用这些设置。

在默认情况下，新的存储桶和对象不允许公有访问，但用户可以通过修改存储桶策略或对象权限来允许公有访问。S3 阻止公有访问提供的设置可以覆盖这些策略和权限，从而能够限制对这些资源的公有访问。有 4 个独立的设置，可以任意组合使用，每个设置都可以应

用于一个存储桶或整个 AWS 账户，如图 5-7 所示。


图 5-7　存储桶的阻止公有访问

当 S3 收到访问存储桶或对象的请求时，它会检查该存储桶或存储桶拥有者的账户是否启用了阻止公有访问的设置。如果存在阻止公有访问的设置，则 S3 会拒绝该请求。如果存储桶的阻止公有访问设置与其他拥有者的账户设置不同，则 S3 会应用最具限制性的存储桶级别和账户级别设置的组合。

### 3. Amazon S3 版本控制

版本控制是一种在同一存储桶中保留对象的多个版本的方法。通过版本控制，可以保留、检索和恢复存储在 S3 存储桶中的每个对象的所有版本。当启用存储桶的版本控制时，S3 会保存所有更改过的对象的各个版本，以防止用户因意外操作或数据损坏而丢失数据。启用了版本控制的存储桶允许恢复因意外删除或覆盖操作而丢失的对象。

例如，如果删除（而不是永久移除）一个对象，S3 则会插入一个删除标记，该标记将成为当前对象版本。始终可以恢复以前的版本。覆盖对象会在存储桶中生成新的对象版本。启用版本控制后，在默认情况下将检索最新写入的版本。可以通过在请求中指定版本来检索对象的早期版本。还可以通过到期策略来限制 S3 中保留的对象版本的数量。值得注意的是，一旦启用了存储桶的版本控制，就无法再将其恢复到未启用版本控制的状态，但是，可以在该存储桶上暂停版本控制。

### 4. Amazon S3 对象锁定

S3 对象锁定使能够使用"一次写入，多次读取"（Write Once，Read Many，WORM）模式存储对象。使用 S3 对象锁定，可以在固定的时间段内或无限期地阻止删除或覆盖对象。S3 对象锁定功能可以在客户定义的保留期内阻止删除对象版本，因此可以通过实施保留策略来进一步保护数据或满足监管要求。无论对象在哪个存储类中，S3 对象锁定保护都将保留，并且会在存储类之间的整个 S3 生命周期转换期间保留。

S3 对象锁定仅适用于受版本控制的存储桶，而保留期和依法保留则适用于个别对象版本。S3 对象锁定提供了两种方式来管理对象保留：保留期限（Retention Period）和依法保

留(Legal Hold)。保留期限指定对象可以保持锁定状态的固定时间段。在此期间,对象将受 WORM 保护,不能被覆盖或删除。依法保留提供的保护与保留期相同,但没有到期日期,而依法保留将一直有效,直至明确将其删除。

S3 对象锁定可以在两种模式中配置。当部署为监管模式(Compliance Mode)时,具有特定 IAM 权限的 AWS 账户能够从对象上删除对象锁定。如果需要更强的不变性以便遵守法规,则可以使用合规模式(Governance Mode)。在合规模式中,包括根账户在内的任何用户都不能移除保护。

**5. 跨区域复制**

跨区域复制是一种自动、异步地的不同 AWS 区域之间复制数据的方法。在数据保护方面,可能会出于多种原因配置复制,例如合规性要求可能要求在更远的距离存储数据。通过跨区域复制,可以在远距离的区域之间复制数据以满足这些要求。另一个原因是灾难恢复。如果某个区域发生自然灾害或攻击,则复制是一个保护数据的绝佳方法。

S3 可以将存储桶中的所有对象或一部分对象复制到任何 AWS 区域。目标存储桶中的对象副本是源存储桶中对象的精确副本。它们具有相同的键名称和元数据。S3 使用 TLS 加密跨 AWS 区域传输的所有数据。

S3 还可以使用由 AWS KMS 或由 S3 管理的密钥跨 AWS 区域复制加密的对象。用于加密对象的密钥副本被发送到目标存储桶。为了防止数据被意外或恶意删除,可以将 S3 对象跨区域复制到其他区域,并在目的地更改拥有者,使难以或无法在两个区域删除相同数据。当将 S3 数据复制到另一个 AWS 区域时,此所有权覆盖是可用选项。可以为源存储桶和目标存储桶设置不同的拥有者,以进一步增强数据保护策略。

也可以设置 S3 跨区域复制(Cross-Region Replication,CRR)策略,以便将数据直接复制到其他 AWS 区域的 S3 Glacier 存储类中,从而实现备份或其他数据保护的目的。可以建立一项简单的 S3 CRR 规则,将存储在一个 AWS 区域内的源存储桶中的每个对象复制到另一个 AWS 区域内的低成本目标存储桶中。

### 5.1.4　Amazon S3 访问分析器


S3 访问分析器可以监控资源访问策略,从而确保这些策略仅提供对 S3 资源的预期访问。S3 访问分析器可评估存储桶访问策略,并使能够发现并快速修复具有潜在意外访问风险的存储桶。

启用之后,S3 访问分析器将被自动显示在 S3 控制台中,可以查看有关存储桶的相关结果和见解,如图 5-8 所示。

图 5-8 中显示有一个名为 tom 的存储桶是可以公开访问的。如果希望立即关闭对存储桶的公开访问,则可以先选择该存储桶,然后单击“阻止所有公开访问”按钮。AWS 建议阻止对存储桶的所有公开访问。

单击列表中的存储桶名称,可以跳转到存储桶的权限页面。在那里,可以检查授予公开访问权限的 ACL 或策略,并对配置进行必要的更改。

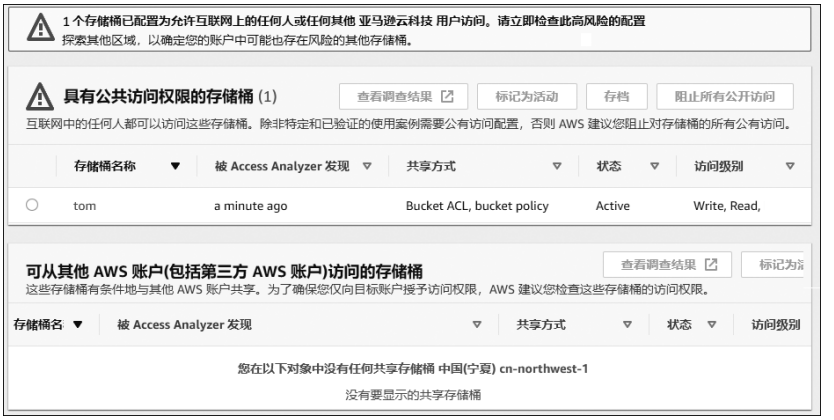


图 5-8　Amazon S3 访问分析器

如果确定需要公开访问，例如，静态网站托管或跨账户共享资源，则可以对查找的存储桶进行“存档”操作，这表明打算保留原来的访问设置。


### 5.1.5　Amazon S3 访问点

随着业务对 S3 存储共享数据集的使用越来越广泛，数据由各种应用程序、团队和个人聚合并访问。管理这种共享存储桶的访问需要复杂的存储桶策略，以控制具有不同权限级别的众多应用程序的访问。随着应用程序数量的增加，存储桶策略变得越来越复杂，牵一发而动全身，管理起来耗时且需要审计以确保更改不会对其他应用程序产生意外影响。

S3 的接入点(Access Point)是一种在 S3 中存取数据的方式，它简化了对共享数据集的大规模数据访问管理。接入点是附加到桶的命名网络端点，可以通过这些接入点执行 S3 对象操作(如 GetObject 和 PutObject)。可以为一个存储桶创建多个接入点，每个接入点都具有不同的权限和网络控制，S3 将这些控制应用于通过该接入点发出的所有请求。S3 接入点和关系型数据库的视图在某些方面的用途相似，它们都提供了一种方式来管理和控制对数据的访问。

S3 访问点使大规模数据访问管理变得简单。以前，需要为不同需求的业务编写、读取、跟踪和审计数百个复杂的权限规则，但现在，不再需要这样做。使用 S3 访问点，可以为每种业务需求创建特定的访问点，并使用针对这些应用程序量身定制的策略访问共享数据集，如图 5-9 所示。这简化了数据访问管理的过程，并使应用程序更容易理解和使用。

以下是关于 S3 访问点的一些常见的使用案例。

(1) 大型共享数据集：使用访问点，可以针对需要访问共享数据集的每个应用程序，将一个大型存储桶策略分解为多个单独的离散的访问点策略。这样可以更轻松地集中精力为应用程序制定正确的访问策略，而不必担心打断共享数据集中任何其他应用程序正在执行的操作。

(2) 将访问限制在 VPC 中：S3 访问点可以将所有 S3 存储访问限制为通过 VPC 执行。