aws

用户指南

AWS Transfer Family

AWS Transfer Family: 用户指南

Table of Contents

什么是 AWS Transfer Family？

AWS Transfer Family 是一种安全的传输服务，使您能够将文件传入和传出 AWS 存储服务。Transfer Family 是该 AWS Cloud 平台的一部分。 AWS Transfer Family 为通过 SFTP、AS2、FTPS 和 FTP 将文件直接传入和传出亚马逊 S3 或 Amazon EFS 提供完全托管的支持。通过维护现有的客户端身份验证、访问和防火墙配置，您可以无缝迁移、自动化和监控文件传输工作流程，因此您的客户、合作伙伴和内部团队或其应用程序不会发生任何变化。

AWS要了解更多信息并开始使用 Amazon Web Services 构建云应用程序，请参阅入门。

AWS Transfer Family 支持将数据从以下存储服务传输或传输到以下 AWS 存储服务。

- Amazon Simple Storage Service (Amazon S3) 存储 有关 Amazon S3 的更多信息，请参阅 Amazon Simple Storage Service 入门。
- Amazon Elastic File System (Amazon EFS) Network File System (NFS) 系统。有关 Amazon EFS 的更多信息，请参阅什么是 Amazon Elastic File System？

AWS Transfer Family 支持通过以下协议传输数据：

- Secure Shell (SSH) File Transfer Protocol (SFTP)：第 3 版
- 安全文件传输协议 (FTPS)
- 文件传输协议 (FTP)
- 适用性声明 2 (AS2)

> **Note**
>
> 对于 FTP 和 FTPS 数据连接，Transfer Family 用于建立数据通道的端口范围为 8192—8200。

File transfer 协议用于不同行业的数据交换工作流程，例如金融服务、医疗保健、广告和零售等。Transfer Family 简化了将文件传输工作流程迁移到 AWS。

以下是 Amazon S3 的一些常见 Transfer Family 使用案例：

- 数据湖 AWS 可供第三方上传，例如供应商和合作伙伴。
- 与客户进行基于订阅的数据分发。

- 组织内部传输。

以下是使用 Amazon EFS 的一些常见使用案例：

- 数据分布
- 供应链
- 内容管理
- Web 服务应用程序

以下是使用 AS2 的一些常见使用案例：

- 具有合规性要求的工作流程依赖于在协议中内置数据保护和安全功能
- 供应链物流
- 付款工作流程
- B usiness-to-business (B2B) 交易
- 与企业资源规划 (ERP) 和客户关系管理 (CRM) 系统集成

使用 Transfer Family，您 AWS 无需运行任何服务器基础架构即可访问支持文件传输协议的服务器。您可以使用此服务将基于文件传输的工作流程迁移到， AWS 同时保持最终用户的客户端和配置不变。首先将主机名与服务器端点关联，然后添加用户并为其配置适当的访问级别。执行此操作后，用户传输请求将直接从 Transfer Family 服务器终端节点获得服务。

Transfer Family 提供以下优势：

- 一项完全托管的服务，可实时扩展以满足您的需求。
- 您无需修改应用程序或运行任何文件传输协议基础设施。
- 将您的数据存储在持久的 Amazon S3 存储空间中，您可以使用原生存储 AWS 服务 进行处理、分析、报告、审计和存档功能。
- 使用 Amazon EFS 作为数据存储，您将获得一个完全托管的弹性文件系统，用于 AWS Cloud 服务和本地资源。Amazon EFS 可在不中断应用程序的情况下按需扩展到 PB 级，并可在您添加和删除文件时自动增加和缩减。这有助于无需预调配和管理容量来满足容量增长需求。
- 一项完全托管的无服务器文件传输工作流服务，可轻松设置、运行、自动化和监控使用 AWS Transfer Family上传的文件的处理。
- 没有预付费用，您仅需支付服务使用费。

在以下各节中，您可以找到对 Transfer Family 不同功能的描述、入门教程、有关如何设置不同协议服务器的详细说明、如何使用不同类型的身份提供程序以及该服务的 API 参考。

要开始使用 Transfer Family，请参阅：

- 如何 AWS Transfer Family 运作
- 先决条件
- AWS Transfer Family 服务器端点入门

如何 AWS Transfer Family 运作

AWS Transfer Family 是一项完全托管的 AWS 服务，您可以使用它通过以下协议将文件传入和传出亚马逊简单存储服务 (Amazon S3) Simple S3 存储或亚马逊弹性文件系统 (Amazon EFS) 文件系统：

- Secure Shell (SSH) File Transfer Protocol (SFTP)：第 3 版
- 安全文件传输协议 (FTPS)
- 文件传输协议 (FTP)
- 适用性声明 2 (AS2)

AWS Transfer Family 最多支持 3 个可用区，并由 auto Scaling 的冗余队列提供支持，用于处理您的连接和传输请求。有关如何使用基于延迟的路由来构建更高的冗余并最大限度地减少网络延迟的示例，请参阅博客文章 SFTP 服务器AWS 传输时最大限度地减少网络延迟。

Transfer Family 托管文件传输工作流程 (MFTW) 是一项完全托管的无服务器文件传输工作流程服务，可轻松设置、运行、自动化和监控使用 AWS Transfer Family上传的文件的处理。客户可以使用 MFTW 自动执行各种处理步骤，例如复制、标记、扫描、筛选、压缩/解压以及加密/解密使用 Transfer Family 传输的数据。这为跟踪和可审核性提供了端到端的可见性。有关更多详细信息，请参阅AWS Transfer Family 托管工作流程。

AWS Transfer Family 支持任何标准文件传输协议客户端。一些常用的客户端如下：

- OpenSSH — Macintosh 和 Linux 命令行实用工具。
- WinSCP — 仅 Windows 图形客户端。
- Cyberduck — Linux、Macintosh 和 Microsoft Windows 图形客户端。
- FileZilla— Linux、Macintosh 和 Windows 图形客户端。

AWS 提供以下 Transfer Family 研讨会。

- 构建一个文件传输解决方案，利用托管 SFTP/FTPS 终 AWS Transfer Family 端节点，利用 Amazon Cognito 和 DynamoDB 进行用户管理。您可以在此处查看本次研讨会的详细信息。
- 构建启用 AS2 的 Transfer Family 端点和 Transfer Family AS2 连接器您可以在此处查看本次研讨会的详细信息。
- 构建一个解决方案，提供规范性指导和动手实验，说明如何在无需修改现有应用程序或管理服务器基础架构 AWS 的情况下构建可扩展且安全的文件传输架构。您可以在此处查看本次研讨会的详细信息。

与 Transfer Family 相关的博客文章

下表列出了包含对Transfer Family客户有用信息的博客文章。该表按时间倒序排序，因此最新的帖子位于表的开头。

博客文章标题和链接	Date
Transfer Family 如何帮助您构建安全、合规的托管文件传输解决方案	2024 年 1 月 3 日
使用检测恶意软件威胁 AWS Transfer Family	2023 年 7 月 20 日
使用扩展 SAP 工作负载 AWS Transfer Family	2023 年 7 月 13 日
使用 PGP 加密和解密文件 AWS Transfer Family	2023 年 6 月 21 日
使用 Azure 活动目录 AWS Transfer Family 进行身份验证和 AWS Lambda	2022 年 12 月 15 日
使用 AWS Transfer Family 托管工作流程自定义文件传送通知	2022 年 10 月 14 日
使用 AWS Transfer Family 工作流程构建云原生文件传输平台	2022 年 1 月 5 日
使用 A AWS Transfer Family 和，启用用户自助密钥管理 AWS Lambda。	2021 年 12 月 17 日

博客文章标题和链接	Date
使用 AWS Transfer Family 和 Amazon S3 增强数据访问控制	2021 年 10 月 5 日
使用 AWS Global Accelerator 和 AWS Transfer Family 服务提高面向互联网的文件传输的吞吐量	2021 年 6 月 7 日
AWS Transfer Family 使用 AWS Web 应用程序防火墙和 Amazon API Gateway 确保安全	2021 年 5 月 5 日
AWS Transfer Family 使用 AWS Web 应用程序防火墙和 Amazon API Gateway 确保安全	2021 年 1 月 15 日
AWS Transfer Family 支持 Amazon Elastic File System	2021 年 1 月 7 日
启用密码身份验证以供 AWS Transfer Family 使用 AWS Secrets Manager	2020 年 11 月 5 日
使用 AWS Transfer Family 和集中数据访问 AWS Storage Gateway	2020 年 6 月 22 日
AWS Lambda 在您的无服务器应用程序中使用 Amazon EFS	2020 年 6 月 18 日
使用 IP 允许列表来保护您的 AWS Transfer Family 服务器	2020 年 4 月 8 日
通过 SFTP 服务器 AWS 传输最大限度地减少网络延迟	2020 年 2 月 19 日
将 SFTP 服务器迁移到 AWS	2020 年 2 月 12 日
使用 chroot 和逻辑目录简化你的 AWS SFTP 结构	2019 年 9 月 26 日
使用 Okta 作为身份提供商 AWS Transfer Family	2019 年 5 月 30 日

先决条件

以下各节描述了使用该 AWS Transfer Family 服务所需的先决条件。您至少需要创建一个亚马逊简单存储服务 (Amazon S3) 存储桶，并通过 (IAM) 角色提供对该存储桶 AWS Identity and Access Management 的访问权限。您的角色还需要建立信任关系。此信任关系允许 Transfer Family 代入 IAM 角色来访问存储桶，以便能为用户的文件传输请求提供服务。

主题

支持的 AWS 区域、终端节点和配额

要以编程方式连接到 AWS 服务，请使用终端节点。例如，美国东部（俄亥俄州）地区客户的终端节点 (us-east-2) 是`transfer.us-east-2.amazonaws.com`。服务限额（也称为限制）是 AWS 账户使用的服务资源或操作的最大数量。在本指南中，您可以在AS2 限额和中找到配额SFTP 连接器配额。

有关支持的 AWS 区域、终端节点和服务配额的更多信息，请参阅中的AWS Transfer Family 终端节点和配额Amazon Web Services 一般参考。

报名参加 AWS

当您注册 Amazon Web Services (AWS) 时，您的 AWS 账户会自动注册使用中的所有服务 AWS，包括 AWS Transfer Family。您只需为使用的服务付费。

如果您已经有一个 AWS 帐户，请跳到下一个任务。如果您还没有 AWS 账户，请使用以下步骤创建。

如果您没有 AWS 账户，请完成以下步骤来创建一个。

报名参加 AWS 账户

1. 打开 https://portal.aws.amazon.com/billing/signup。
2. 按照屏幕上的说明进行操作。

在注册时，将接到一通电话，要求使用电话键盘输入一个验证码。

当您注册时 AWS 账户，就会创建AWS 账户根用户一个。根用户有权访问该账户中的所有 AWS 服务 和资源。作为安全最佳实践，请 为管理用户分配管理访问权限，并且只使用根用户执行需要根用户访问权限的任务。

有关定价以及用于估算使用 AWS Pricing Calculator Transfer Family 的成本的信息，请参阅AWS Transfer Family 定价。

有关 AWS 区域可用性的信息，请参阅中的AWS Transfer Family 终端节点和配额AWS 一般参考。

配置要与使用的存储 AWS Transfer Family

本主题介绍可以与配合使用的存储选项 AWS Transfer Family。你可以使用 Amazon S3 或 Amazon EFS 作为 Transfer Family 服务器的存储。

目录

配置 Amazon S3 存储桶

AWS Transfer Family 访问您的 Amazon S3 存储桶以处理用户的传输请求，因此在设置支持文件传输协议的服务器时，您需要提供 Amazon S3 存储桶。您可以使用现有存储桶或新建一个存储桶。

> **Note**
>
> 您不必使用位于相同 AWS 区域中的服务器和 Amazon S3 存储桶，但是我们建议将此作为最佳实践。

在设置用户时，您可以为每个用户分配一个 IAM 角色。此角色决定了用户对 Amazon S3 存储桶的访问级别。

有关创建新存储桶的更多信息，请参阅 Amazon Simple Storage Service 用户指南中的如何创建 S3 存储桶？。

> **Note**
>
> 您可以使用 Amazon S3 对象锁定在固定的时间段内或无限期内阻止对象被覆盖。Transfer Family 的工作方式与其他服务相同。如果对象存在且受保护，则不允许写入或删除该文件。有关 Amazon S3 对象锁定的更多详细信息，请参阅《Amazon 简单存储服务用户指南》中的使用 Amazon S3 对象锁定。

Amazon S3 接入点

AWS Transfer Family 支持 Amazon S3 接入点，这是 Amazon S3 的一项功能，可让您轻松管理对共享数据集的精细访问。您可以在任何使用 S3 存储桶名称的地方使用 S3 接入点别名。您可以在 Amazon S3 中为拥有不同权限的用户创建数百个访问点，以访问 Amazon S3 存储桶中的共享数据。

例如，您可以使用接入点允许三个不同的团队访问同一个共享数据集，其中一个团队可以从 S3 读取数据，第二个团队可以将数据写入 S3，第三个团队可以从 S3 读取、写入和删除数据。要实现如上所述的精细访问控制，您可以创建一个 S3 接入点，该接入点包含向不同团队提供非对称访问权限的策略。您可以将 S3 接入点与 Transfer Family 服务器配合使用来实现精细的访问控制，而无需创建涵盖数百个用例的复杂 S3 存储桶策略。要详细了解如何在 Transfer Family 服务器上使用 S3 接入点，请参阅使用AWS Transfer Family 和 Amazon S3 增强数据访问控制博客文章。

> **Note**
>
> AWS Transfer Family 目前不支持 Amazon S3 多区域接入点。

亚马逊 S3 的 HeadObject 行为

> **Note**
>
> 在创建或更新 Transfer Family 服务器时，您可以优化 Amazon S3 目录的性能，从而消除`HeadObject`调用。

在 Amazon S3 中，存储桶和对象是主要资源，并且对象存储在存储桶中。Amazon S3 可以模仿分层文件系统，但有时行为可能与典型文件系统不同。例如，在 Amazon S3 中，目录不是头等概念，而是基于对象密钥。 AWS Transfer Family 推断出目录路径的方法是：用正斜杠字符 (/) 分割对象的密钥，将最后一个元素视为文件名，然后将具有相同前缀的文件名分组到同一路径下。当您使用`mkdir`或使用 Amazon S3 控制台创建空目录时，创建零字节对象是为了表示文件夹的路径。这些对象的密钥以尾随的正斜杠结尾。Amazon S3 用户指南中的使用文件夹在 Amazon S3 控制台中组织对象中描述了这些零字节对象。

当您运行`ls`命令时，如果某些结果是 Amazon S3 零字节对象（这些对象的密钥以正斜杠字符结尾），Transfer Family 会针对每个对象`HeadObject`发出请求（详情请参阅《亚马逊简单存储服务 API 参考》HeadObject中）。使用 Amazon S3 作为 Transfer Family 的存储空间时，这可能会导致以下问题。

授予仅写入和列出文件的权限

在某些情况下，您可能只想提供对 Amazon S3 对象的写入权限。例如，您可能希望提供写入（或上传）和列出存储桶中对象的权限，但不提供读取（下载）对象的权限。要使用文件传输客户端执行`ls`和`mkdir`命令，您必须拥有 Amazon S3 `ListObjects` 和`PutObject`权限。但是，当 Transfer Family 需要`HeadObject`调用写入文件或列出文件时，呼叫失败并显示拒绝访问的错误，因为此调用需要`GetObject`权限。

> **Note**
>
> 在创建或更新 Transfer Family 服务器时，您可以优化 Amazon S3 目录的性能，从而消除`HeadObject`调用。

在这种情况下，您可以通过添加 AWS Identity and Access Management (IAM) 策略条件来授予访问`GetObject`权限，该条件仅为以斜杠 (/) 结尾的对象添加权限。此条件可防止`GetObject`调用文件（因此无法读取文件），但允许用户列出和遍历文件夹。以下示例策略仅提供对您的 Amazon S3 存储桶的写入和列出权限。要使用此策略，请*`DOC-EXAMPLE-BUCKET`*替换为存储桶的名称。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowListing",
            "Effect": "Allow",
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
        },
        {
            "Sid": "AllowReadWrite",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ]
         },
      {
            "Sid": "DenyIfNotFolder",
            "Effect": "Deny",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectVersion"
            ],
            "NotResource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*/"
            ]
         }
      ]
}
```

> **Note**
>
> 此策略不允许用户追加文件。换句话说，分配了此策略的用户无法打开文件来向其添加内容或修改文件。此外，如果您的用例要求在上传文件之前进行HeadObject调用，则此策略对您不起作用。

大量零字节对象导致延迟问题

如果您的 Amazon S3 存储桶包含大量这样的零字节对象，Transfer Family 会发出大量`HeadObject`调用，这可能会导致处理延迟。

解决此问题的一种可能方法是删除所有零字节对象。请注意以下几点：

- 空目录将不再存在。只有当目录的名称位于对象的密钥中时，才会存在目录。
- 不会阻止某人再次调用`mkdir`并破坏事务处理。您可以通过制定阻止目录创建的策略来缓解这种情况。
- 有些场景会使用这些 0 字节的对象。例如，您有一个像 /inboxes/customer1000 这样的结构，每天都会清理收件箱目录。

另一种可能的解决方案是通过策略条件限制可见的对象数量，以减少`HeadObject`调用次数。要使之成为可行的解决方案，您需要接受这样一个事实，即您可能只能查看有限的一组子目录。

配置 Amazon EFS 文件系统

AWS Transfer Family 访问亚马逊 Elastic File System (Amazon EFS)，为用户的传输请求提供服务。因此，在设置支持文件传输协议的服务器时，您必须提供 Amazon EFS 文件系统。您可以使用现有文件系统或新建一个文件系统。

请注意以下几点：

- 当您使用 Transfer Family 服务器和 Amazon EFS 文件系统时，服务器和文件系统必须处于同一位置 AWS 区域。
- 服务器和文件系统不必位于同一个账户中。如果服务器和文件系统不在同一个账户中，则文件系统策略必须为用户角色提供明确的权限。

  有关如何设置多个账户的信息，请参阅AWS Organizations 用户指南中的管理组织中的 AWS 账户。
- 在设置用户时，您可以为每个用户分配一个 IAM 角色。此角色决定了用户对 Amazon EFS 文件系统的访问级别。
- 有关挂载 Amazon EFS 文件系统的详细信息，请参阅挂载 Amazon EFS 文件系统。

有关如何 AWS Transfer Family 与 Amazon EFS 协同工作的更多详细信息，请参阅《亚马逊弹性文件系统用户指南》中的 “使用 AWS Transfer Family 访问您的 Amazon E FS 文件系统中的文件”。

Amazon EFS 文件所有权

Amazon EFS 使用便携式操作系统接口 (POSIX) 文件权限模型来表示文件所有权。

在 POSIX 中，系统中的用户分为三个不同的权限类别：当您允许用户使用访问存储在 Amazon EFS 文件系统中的文件时 AWS Transfer Family，必须为他们分配一个 “POSIX 配置文件”。此配置文件用于确定他们对 Amazon EFS 文件系统中文件和目录的访问权限。

- 用户 (u)：文件或目录的所有者。通常，文件或目录的创建者也是所有者。
- 组 (g)：一组需要对他们共享的文件和目录具有相同访问权限的用户。
- 其他 (o)：除所有者和群组成员外，有权访问系统的所有其他用户。此权限类别也称为“公有”类别。

在 POSIX 权限模型中，每个文件系统对象（文件、目录、符号链接、命名管道和套接字）都与前面提到的三组权限相关联。Amazon EFS 对象具有关联的 Unix 风格模式。此模式值定义了对该对象执行操作的权限。

此外，在 Unix 风格的系统上，用户和组被映射到数字标识符，Amazon EFS 使用这些标识符来表示文件所有权。对于 AmazonEFS，对象由单个所有者和单个组所有。当用户尝试访问文件系统对象时，Amazon EFS 使用映射的数字 ID 来检查权限。

为 Transfer Family 设置 Amazon EFS 用户

设置 Amazon EFS 用户之前，您可以执行以下操作之一：

- 您可以在 Amazon EFS 中创建用户并设置他们的主文件夹。有关详细信息，请参阅 在 Amazon EFS 上配置 Transfer Family 用户。
- 如果您愿意添加根用户，则可以创建 Amazon EFS 根用户。

> **Note**
>
> Transfer Family 服务器不支持 Amazon EFS 接入点来设置 POSIX 权限。Transfer Family 用户的 POSIX 配置文件（如上一节所述）提供了设置 POSIX 权限的功能。这些权限是在用户级别设置的，用于基于 UID、GID 和辅助 GID 的精细访问。

在 Amazon EFS 上配置 Transfer Family 用户

Transfer Family 会将用户映射到您指定的 UID/GID 和目录。如果 UID/GID/目录在 EFS 中尚不存在，则应先创建它们，然后再在 Transfer 中将其分配给用户。有关创建 Amazon EFS 用户的详细信息，请参阅 Amazon Elastic File System 用户指南中的在网络文件系统 (NFS) 级别处理用户、组和权限。

在 Transfer Family 中设置亚 Amazon EFS 用户的步骤

1. 使用 `PosixProfile` 字段在 Transfer Family 中为用户映射 EFS UID 和 GID。
2. 如果您希望用户在登录时在特定文件夹中启动，则可以在`HomeDirectory`字段下指定 EFS 目录。

您可以使用 CloudWatch 规则和 Lambda 函数自动执行该过程。有关与 EFS 交互的 Lambda 函数示例，请参阅AWS Lambda 在您的无服务器应用程序中使用 Amazon EFS。

此外，您还可以为 Transfer Family 用户配置逻辑目录。有关详细信息，请参阅使用逻辑目录简化您的 Transfer Family 目录结构主题中的为 Amazon EFS 配置逻辑目录章节。

创建 Amazon EFS 根用户

如果您的组织愿意通过 SFTP/FTPS 为用户配置启用根用户访问权限，则可以创建一个 UID 和 GID 为 0 的用户（根用户），然后使用该根用户创建文件夹，为其余用户分配 POSIX ID 所有者。此选项的优点是，无需挂载 Amazon EFS 文件系统。

执行添加 Amazon EFS 服务托管用户中描述的步骤，为用户 ID 和组 ID 输入 0（零）。

受支持的 Amazon EFS 命令

对于 AWS Transfer Family，Amazon EFS 支持以下命令。

- `cd`
- `ls/dir`
- `pwd`
- `put`
- `get`
- `rename`
- `chown`: 只有根用户（即 uid 为 0 的用户）可以更改文件和目录的所有权和权限。

- chmod：只有根用户可以更改文件和目录的所有权和权限。
- chgrp：根用户或只能将文件组更改为次要组之一的文件所有者支持。
- ln -s/symlink
- mkdir
- rm/delete
- rmdir
- chmtime

创建 IAM 角色和策略

本主题介绍可与之配合使用的策略和角色类型 AWS Transfer Family，并介绍创建用户角色的过程。它还描述了会话策略的工作原理，并提供了一个用户角色示例。

AWS Transfer Family 使用以下类型的角色：

- 用户角色-允许服务管理的用户访问必要的 Transfer Family 资源。 AWS Transfer Family 在 Transfer Family 用户 ARN 的背景下担任此角色。
- 访问角色 - 仅提供对正在传输的 Amazon S3 文件的访问权限。对于入站 AS2 传输，访问角色使用协议的 Amazon 资源名称（ARN）。对于出站 AS2 传输，访问角色使用连接器的 ARN。
- 调用角色 – 用于作为服务器自定义身份提供程序的 Amazon API Gateway。Transfer Family 在 Transfer Family 服务器 ARN 的背景下扮演这个角色。
- 日志角色-用于将条目登录到 Amazon CloudWatch。Transfer Family 使用此角色记录成功和失败的详细信息以及有关文件传输的信息。Transfer Family 在 Transfer Family 服务器 ARN 的背景下扮演这个角色。对于出站 AS2 传输，日志角色使用连接器 ARN。
- 执行角色 - 允许 Transfer Family 用户调用和启动工作流程。Transfer Family 在 Transfer Family 工作流程 ARN 的背景下担任此角色。

除了这些角色之外，您还可以使用会话策略。会话策略用于在必要时限制访问权限。请注意，这些策略是独立的：也就是说，您不会将这些策略添加到角色中。相反，您可以直接向 Transfer Family 用户添加会话策略。

> **Note**
>
> 创建服务管理的 Transfer Family 用户时，可以选择基于主文件夹自动生成策略。如果您想限制用户访问自己的文件夹，这是一个有用的快捷方式。此外，您还可以在会话策略工作原理中查

看有关会话策略的详细信息以及示例。您还可以在《IAM 用户指南》的会话策略中找到有关会话策略的更多信息。

主题

创建用户角色

在创建用户时，您会做出大量有关用户访问权限的决定。这些决定包括用户可以访问哪些 Amazon S3 存储桶或 Amazon EFS 文件系统、每个 Amazon S3 存储桶的哪些部分和文件系统中的哪些文件可以访问，以及用户拥有哪些权限（例如PUT或GET）。

要设置访问权限，您需要创建基于身份 AWS Identity and Access Management (IAM) 的策略和角色来提供该访问信息。作为此过程的一部分，您为用户提供对 AmazonS3 存储桶或 Amazon EFS 文件系统的访问权限，这些文件系统是文件操作的目标或源。为此，请执行以下简要步骤（稍后将详细介绍）：

创建用户角色

1. 为创建 IAM 策略 AWS Transfer Family。为创建 IAM 策略 AWS Transfer Family中对此进行了描述。
2. 创建 IAM 角色并附加新的 IAM policy。有关示例，请参阅读/写访问策略示例。
3. 在和 IAM 角色 AWS Transfer Family 之间建立信任关系。建立信任关系中对此进行了描述。

以下过程介绍如何创建 IAM policy 和角色。

为创建 IAM 策略 AWS Transfer Family

1. 访问：https://console.aws.amazon.com/iam/，打开 IAM 控制台。
2. 在导航窗格中，选择 Policies (策略)，然后选择 Create policy (创建策略)。
3. 在创建策略页面上，选择 JSON 选项卡。
4. 在显示的编辑器中，将编辑器的内容替换为要附加到 IAM 角色的 IAM policy。

   您可以授予读/写访问权限或限制用户访问其主目录。有关更多信息，请参阅 读/写访问策略示例。

5. 选择查看策略，并提供策略的名称和描述，然后选择创建策略。

接下来，您将创建一个 IAM 角色并向该角色附加新的 IAM 策略。

为创建 IAM 角色 AWS Transfer Family

1. 在导航窗格中，选择 Roles（角色），然后选择 Create role（创建角色）。

   在创建角色页面上，确保已选择AWS 服务。
2. 从服务列表中选择转移，然后选择下一步：权限。这在 AWS Transfer Family 和之间建立了信任关系 AWS。
3. 在附加权限策略部分中，找到并选择刚刚创建的策略，然后选择下一步：标签。
4. （可选）输入标签的键和值，然后选择下一步：审核。
5. 在审核页面上，输入新角色的名称和描述，然后选择创建角色。

接下来，在 AWS Transfer Family 和之间建立信任关系 AWS。

建立信任关系

> **Note**
>
> 在我们的示例中，我们同时使用 `ArnLike` 和 `ArnEquals`。它们在功能上是相同的，因此您可以在制定策略时使用其中任何一个。Transfer Family 文档在条件包含通配符时使用`ArnLike`，`ArnEquals`用于表示完全匹配的条件。

1. 在 IAM 控制台中，选择您刚创建的角色。
2. 在 Summary (摘要) 页面上，选择 Trust relationships (信任关系)，然后选择 Edit trust relationship (编辑信任关系)。
3. 在编辑信任关系编辑器中，确保服务是"`transfer.amazonaws.com`"。编辑后的访问策略如下所示。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
```

```
      "Principal": {
        "Service": "transfer.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

建议您使用 aws:SourceAccount 和 aws:SourceArn 条件键来防止出现混淆代理人问题。源账户是域的所有者，并且源 ARN 是域的 ARN。例如：

```
"Condition": {
    "StringEquals": {
        "aws:SourceAccount": "account_id"
    },
    "ArnLike": {
        "aws:SourceArn": "arn:aws:transfer:region:account_id:user/*"
    }
}
```

如果您希望限制到特定的服务器而不是用户账户中的任何服务器，也可以使用ArnLike条件。例如：

```
"Condition": {
    "ArnLike": {
        "aws:SourceArn": "arn:aws:transfer:region:account-id:user/server-id/*"
    }
}
```

> **Note**
>
> 在上述示例中，将每个*******替换为您自己的信息。

有关混淆代理人问题的详细信息以及更多示例，请参阅防止跨服务混淆代理。

4. 选择更新信任策略以更新访问策略。

现在，您已经创建了一个 IAM 角色， AWS Transfer Family 允许您代表您调用 AWS 服务。您已将创建的 IAM policy 附加到该角色，以授予对用户的访问权限。在AWS Transfer Family 服务器端点入门部分中，此角色和策略将分配给您的用户或用户。

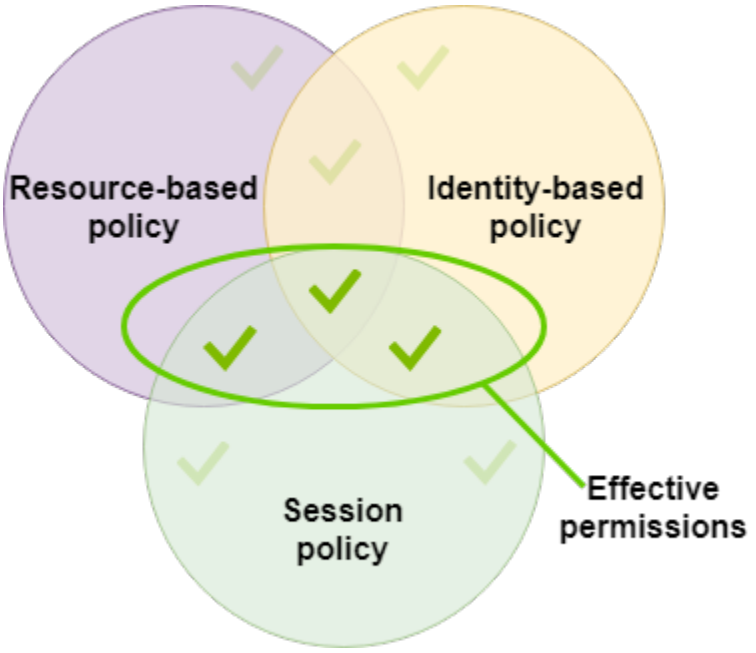
另请参阅

- 有关 IAM 角色的更多一般信息，请参阅 I A M 用户指南中的创建角色以向 AWS 服务委派权限。
- 要详细了解 Amazon S3 资源基于身份的策略，请参阅《Amazon 简单存储服务用户指南》中的 Amazon S3 中的身份和访问管理。
- 要了解有关 Amazon EFS 资源基于身份的策略的更多信息，请参阅《Amazon 弹性文件系统用户指南》中的使用 IAM 控制文件系统数据访问。

会话策略工作原理

管理员创建角色时，该角色通常包含涵盖多个用例或团队成员的广泛权限。如果管理员配置了控制台 URL，则他们可以使用会话策略来减少对生成的会话的权限。例如，如果您创建具有读/写访问权限的角色，则可以设置一个 URL，限制用户只能访问其主目录。

会话策略是当您以编程方式为角色或用户创建临时会话时作为参数传递的高级策略。会话策略对于锁定用户非常有用，这样他们就只能访问存储桶中包含其用户名的对象前缀的部分。下图显示了会话策略的权限是会话策略和基于资源的策略的交集，以及会话策略和基于身份策略的交集。



有关更多详细信息，请参阅 IAM 用户指南中的会话策略。

在中 AWS Transfer Family，只有当您向 Amazon S3 传输或从 Amazon S3 传输数据时，才支持会话策略。以下示例策略是一个会话策略，它仅限制用户访问其 home 目录。请注意以下几点：

- 只有当您需要启用跨账户存取时，才需要`GetObjectACL`和`PutObjectACL`语句。也就是说，您的 Transfer Family 服务器需要访问其他账户中的存储桶。
- 会话策略的最大长度为 2048 个字符。有关更多详细信息，请参阅 API 参考中`CreateUser`操作的策略请求参数。
- 如果您的 Amazon S3 存储桶使用 AWS Key Management Service (AWS KMS) 进行加密，则必须在策略中指定其他权限。有关更多信息，请参阅 Amazon S3 中的数据加密。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowListingOfUserFolder",
            "Action": [
                "s3:ListBucket"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::${transfer:HomeBucket}"
            ],
            "Condition": {
                "StringLike": {
                    "s3:prefix": [
                        "${transfer:HomeFolder}/*",
                        "${transfer:HomeFolder}"
                    ]
                }
            }
        },
        {
            "Sid": "HomeDirObjectAccess",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:DeleteObject",
                "s3:DeleteObjectVersion",
                "s3:GetObjectVersion",
                "s3:GetObjectACL",
```

```
                "s3:PutObjectACL"
            ],
            "Resource": "arn:aws:s3:::${transfer:HomeDirectory}/*"
        }
    ]
}
```

> **Note**
>
> 前面的策略示例假设用户的主目录设置为包含尾部斜杠，以表示它是一个目录。另一方面，如果您设置的用户HomeDirectory不带尾部的斜杠，则应将其作为策略的一部分。

在前面的示例策略中，请注意使用transfer:HomeFolder、transfer:HomeBucket和transfer:HomeDirectory策略参数。这些参数是为用户配置的设置的，如HomeDirectory和中所述实施您的 API Gateway 方法。HomeDirectory这些参数具有以下定义：

- transfer:HomeBucket参数将替换为的HomeDirectory第一个组件。
- transfer:HomeFolder参数将替换为HomeDirectory参数的其余部分。
- transfer:HomeDirectory参数删除了前导正斜杠 (/)，因此可以在Resource语句中将其用作 S3 Amazon 资源名称（ARN）的一部分。

> **Note**
>
> 如果您使用的是逻辑目录（即用户的homeDirectoryType是LOGICAL），则不支持这些策略参数（HomeBucket、HomeDirectory和HomeFolder）。

例如，假设为 Transfer Family 用户配置的HomeDirectory参数是/home/bob/amazon/stuff/。

- transfer:HomeBucket 设置为 /home。
- transfer:HomeFolder 设置为 /bob/amazon/stuff/。
- transfer:HomeDirectory 变为 home/bob/amazon/stuff/。

第一个"Sid"允许用户列出从/home/bob/amazon/stuff/开始的所有目录。

第二个"Sid"限制用户对同一路径/home/bob/amazon/stuff/的put和get访问权限。

读/写访问策略示例

授予对 Amazon S3 存储桶的读取/写入访问权限

以下示例策略 AWS Transfer Family 授予对您的 Amazon S3 存储桶中对象的读/写权限。

请注意以下几点：

- 请将 *DOC-EXAMPLE-BUCKET* 替换为您的 Amazon S3 Bucket 名称。
- 只有当您需要启用跨账户存取时，才需要GetObjectACL和PutObjectACL语句。也就是说，您的 Transfer Family 服务器需要访问其他账户中的存储桶。
- 只有在正在访问的 Amazon S3 存储桶上启用版本控制时，才需要使用GetObjectVersion和DeleteObjectVersion语句。

> **Note**
>
> 如果您曾经为存储桶启用过版本控制，则需要这些权限，因为您只能在 Amazon S3 中暂停版本控制，而不能完全将其关闭。有关详细信息，请参阅未版本化、启用版本控制和已暂停版本控制的存储桶。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowListingOfUserFolder",
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
            ]
        },
        {
            "Sid": "HomeDirObjectAccess",
            "Effect": "Allow",
            "Action": [
```

```
                "s3:PutObject",
                "s3:GetObject",
                "s3:DeleteObject",
                "s3:DeleteObjectVersion",
                "s3:GetObjectVersion",
                "s3:GetObjectACL",
                "s3:PutObjectACL"
            ],
            "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
        }
    ]
}
```

授予文件系统访问 Amazon EFS 文件系统中文件的权限

> **Note**
>
> 除了策略外，您还必须确保您的 POSIX 文件权限授予了相应的访问权限。有关更多信息，请参阅《Amazon Elastic File System 用户指南》中的在网络文件系统（NFS）级别处理用户、组和权限。

以下示例策略允许根文件系统访问您的 Amazon EFS 文件系统中的文件。

> **Note**
>
> 在以下示例中，将*##*替换为您的区域，将*## ID* 替换为文件所在的账户，*file-system-id*使用您的亚马逊弹性文件系统 (Amazon EFS) 的 ID。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "RootFileSystemAccess",
            "Effect": "Allow",
            "Action": [
                "elasticfilesystem:ClientRootAccess",
                "elasticfilesystem:ClientMount",
                "elasticfilesystem:ClientWrite"
```

```
            ],
            "Resource": "arn:aws:elasticfilesystem:region:account-id:file-system/file-
system-id"
        }
    ]
}
```

以下示例策略授予用户文件系统访问您的 Amazon EFS 文件系统中文件的权限。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "UserFileSystemAccess",
            "Effect": "Allow",
            "Action": [
                "elasticfilesystem:ClientMount",
                "elasticfilesystem:ClientWrite"
            ],
            "Resource": "arn:aws:elasticfilesystem:region:account-id:file-system/file-
system-id"
        }
    ]
}
```

Transfer Family 教程

该 AWS Transfer Family 用户指南提供了多个用例的详细演练。

- AWS Transfer Family 服务器端点入门：本教程将引导你创建 SFTP Transfer Family 服务器和服务管理用户，然后演示如何使用客户端传输文件。
- 设置和使用 SFTP 连接器：本教程说明如何设置 SFTP 连接器，然后在 Amazon S3 存储和 SFTP 服务器之间传输文件。
- 将 Amazon API Gateway 方法设置为自定义身份提供商：本教程说明了如何设置 Amazon API Gateway 方法并将其用作自定义身份提供者将文件上传到 AWS Transfer Family 服务器。
- 设置用于解密文件的托管工作流程：本教程说明如何设置包含解密步骤的托管工作流程，以及如何将加密文件上传到 Amazon S3 存储桶，然后查看解密后的文件。
- 设置 AS2 配置：本教程介绍了配置 AS2 Transfer Family 服务器所需的步骤。这里有关于导入证书、创建配置文件和协议、（可选）创建 AS2 连接器以及测试配置的说明。

主题

AWS Transfer Family 服务器端点入门

使用本教程开始使用 AWS Transfer Family（Transfer Family）。您将学习如何使用 Amazon S3 存储创建具有可公开访问端点的启用 SFTP 的服务器，如何添加具有服务托管身份验证的用户，以及如何使用 Cyberduck 传输文件。

主题

- 步骤 3：添加服务托管用户
- 步骤 4：使用客户端传输文件

先决条件

开始之前，请确保完成先决条件中的要求。在此设置中，您将创建一个亚马逊简单存储服务 (Amazon S3) 存储桶和 AWS Identity and Access Management 一个 (IAM) 用户角色。

使用 AWS Transfer Family 控制台需要权限，也需要权限才能配置 Transfer Family 使用的其他 AWS 服务，例如亚马逊简单存储服务 AWS Certificate Manager、亚马逊弹性文件系统和亚马逊 Route 53。例如，对于 AWS 使用 Transfer Family 传入和传出文件的用户，AmazonS3 会FullAccess授予设置和使用 Amazon S3 存储桶的权限。创建 Amazon S3 存储桶需要此策略中的一些权限。

要使用 Transfer Family 控制台，您需要满足以下条件：

- AWSTransferConsoleFullAccess向您的 SFTP 用户授予创建 Transfer Family 资源的权限。
- 只有当您希望 Transfer Family 在 Amazon CloudWatch Logs 中自动为您的服务器创建日志角色或为登录服务器的用户创建用户角色时，才需要 IA@@ M FullAccess（或者具体来说是允许创建 IAM 角色的策略）。
- 要创建和删除 VPC 服务器类型，您需要在策略中添加操作 ec2: CreateVpcEndpoint 和 ec2: DeleteVpcEndpoints。

> **Note**
>
> 一般使用并不需要 A mazonS3 FullAccess 和 IAM FullAccess 政策。 AWS Transfer Family此处将它们作为一种简单的方法来确保您需要的所有权限都得到满足。此外，这些是 AWS 托管策略，它们是可供所有 AWS 客户使用的标准策略。您可以查看这些策略中的个人权限，并确定实现您的目的所需的最低权限集。

步骤 1：登录到 AWS Transfer Family 控制台

要登录 Transfer Family

1. 登录 AWS Management Console 并打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 对于账户 ID 或别名，请输入您的 AWS 账户 ID。

3. 对于IAM 用户名，输入您为 Transfer Family 创建的用户角色名称。
4. 在 “密码” 中，输入您的 AWS 帐户密码。
5. 选择 Sign in（登录）。

步骤 2：创建启用 SFTP 的服务器

Secure Shell (SSH) 文件传输协议 (SFTP) 是一种用于通过互联网安全传输数据的网络协议。该协议支持 SSH 的完整安全和身份验证功能。它被广泛应用于金融服务、医疗保健、零售和广告等各行各业的业务合作伙伴之间交换数据，包括敏感信息。

要创建启用 SFTP 的服务器

1. 从导航窗格中选择服务器，然后选择创建服务器。
2. 在选择协议中，选择 SFTP，然后选择下一步。
3. 在选择身份提供商中，选择服务托管，在 Transfer Family 中存储用户身份和密钥，然后选择下一步。
4. 在选择端点中，执行以下操作：
   a. 对于端点类型，选择可公开访问的端点类型。
   b. 对于自定义主机名，选择无。
   c. 选择下一步。
5. 在选择域名中，选择 Amazon S3。
6. 在配置其他详细信息中，执行以下操作：
   a. 要进行CloudWatch 登录，请选择创建新角色以允许 Transfer Family 自动创建 IAM 角色，前提是您拥有创建新角色的相应权限。创建的 IAM 角色被称为`AWSTransferLoggingAccess`。
   b. 对于加密算法选项，请选择包含允许服务器使用的加密算法的安全策略。默认的安全策略为`TransferSecurityPolicy-2020-06`。
   c. 选择下一步。
7. 在审核并创建中，选择创建服务器。您将进入服务器页面。

您的新服务器状态更改为在线可能需要几分钟时间。此时，您的服务器可以执行文件操作，但您需要先创建一个用户。

步骤 3：添加服务托管用户

要向启用 SFTP 的服务器添加用户

1. 在服务器页面上，选中您要将用户添加到的服务器复选框。
2. 选择添加用户。
3. 在用户配置部分的用户名中，输入用户名。此用户名长度最少为 3 个字符，最多为 100 个字符 您可以在用户名中使用以下字符：a–z、A-Z、0–9、下划线“_”、连字符“-”、句点“.”和“@”符号。用户名不能以连字符、句点或 @ 符号开头。
4. 对于访问权限，选择您之前创建的提供对 Amazon S3 存储桶访问权限的 IAM 角色。

   您可使用创建 IAM 角色和策略中的过程创建此 IAM 角色。该 IAM 角色包括一个提供对您 Amazon S3 存储桶访问权限的 IAM policy。它还包括与 AWS Transfer Family 服务的信任关系，该关系在另一个 IAM 策略中定义。

   > **Note**
   >
   > 服务托管用户的 IAM 角色必须包含访问所需存储桶的权限。访问所需存储桶的权限包含在 S3 中FullAccess，它授予对 S3 资源的管理员级别权限。

5. 对于策略，选择无。
6. 对于主目录，选择用于存储要传输的数据的 Amazon S3 存储桶 AWS Transfer Family。输入用户在使用其客户端登录时转到的 home 目录的路径。

   如果您将此参数留空，则使用 Amazon S3 存储桶的 `root` 目录。在这种情况下，请确保您的 IAM 角色提供对此 `root` 目录的访问权限。

   > **Note**
   >
   > 我们建议您选择包含用户的用户名的目录路径，这使得您可以更高效地使用会话策略。会话策略将用户在 Amazon S3 存储桶中的访问权限限制为该用户的 home 目录。

7. 对于受限，选中该复选框，这样您的用户就无法访问该文件夹之外的任何内容，也看不到 Amazon S3 存储桶或文件夹名称。

> **Note**
>
> 当为用户分配主目录并限制用户访问该主目录时，这应该足以锁定用户对指定文件夹的访问权限。当您需要应用进一步的控制措施，请使用会话策略。

8. 对于SSH 公有密钥，输入 SSH 密钥对的 SSH 公有密钥部分。

   您的密钥先由服务进行验证，然后才能添加新用户。

> **Important**
>
> SSH 公有密钥的格式为 `ssh-rsa <string>`。有关如何生成 SSH 密钥对的说明，请参阅为服务托管用户生成 SSH 密钥。

9. （可选）对于键和值，输入一个或多个标记作为键-值对，然后选择添加标记。
10. 选择 Add (添加) 可将您的新用户添加到所选服务器。

    新用户将出现在服务器详细信息页面的用户部分。

步骤 4：使用客户端传输文件

通过在客户端中指定传输操作，您可以通过 AWS Transfer Family 服务传输文件。 AWS Transfer Family 支持多个客户端。有关详细信息，请参阅 使用客户端通过服务器端点传输文件

本部分包含使用 Cyberduck 和 OpenSSH 的过程。

主题

- 使用 Cyberduck
- 使用 OpenSSH

使用 Cyberduck

AWS Transfer Family 使用 Cyberduck 传输文件

1. 打开 Cyberduck 客户端。
2. 选择打开连接。
3. 在打开连接对话框中，选择 SFTP（SSH 文件传输协议）。

4. 对于服务器，输入您的服务器端点。服务器端点位于服务器详细信息页面上，请参阅查看 SFTP、FTPS 和 FTP 服务器的详细信息。
5. 在端口号中，输入**22** SFTP。
6. 对于 Username (用户名)，输入您在管理服务器端点的用户中创建的用户的名称。
7. 对于SSH 私有密钥，请选择或输入 SSH 私有密钥。
8. 选择连接。
9. 执行文件传输。

   根据您的文件所在的位置，执行以下操作之一：

   - 在您的本地目录（源）中，选择您要传输的文件，然后将这些文件拖放到 Amazon S3 目录（目标）中。
   - 在 Amazon S3 目录（源）中，选择您要传输的文件，然后将这些文件拖放到您的本地目录（目标）中。

使用 OpenSSH

按照下文中的说明，使用 OpenSSH 从命令行传输文件。

> **Note**
>
> 此客户端仅适用于启用 SFTP 的服务器。

AWS Transfer Family 使用 OpenSSH 命令行实用程序传输文件

1. 在 Linux 或 Macintosh 上，打开命令终端。
2. 在提示符中，输入以下命令：`% sftp -i transfer-key sftp_user@service_endpoint`

   在前面的命令中，`sftp_user` 是用户名，`transfer-key` 是 SSH 私有密钥。此处`service_endpoint`是服务器的终端节点，如所选服务器的 AWS Transfer Family 控制台中所示。

   此时应显示 `sftp` 提示符。
3. （可选）要查看用户的主目录，请在`sftp`提示符下输入以下命令：`sftp> pwd`
4. 在下一行上，输入以下文本：`sftp> cd /mybucket/home/sftp_user`

在本入门练习中，将此 Amazon S3 存储桶作为文件传输的目标。

5. 在下一行上，输入以下命令：`sftp> put filename.txt`

   put命令将文件传输到 Amazon S3 存储桶中。

   此时将显示类似于下文的消息，指示文件传输正在进行或者已完成。

   ```
   Uploading filename.txt to /my-bucket/home/sftp_user/filename.txt

   some-file.txt 100% 127 0.1KB/s 00:00
   ```

设置用于解密文件的托管工作流程

本教程说明如何设置包含解密步骤的托管工作流程。本教程还展示了如何将加密文件上传到 Amazon S3 存储桶，然后在同一存储桶中查看解密后的文件。

> **Note**
>
> AWS 存储博客上有一篇文章描述了如何加密和解密文件，使用 PGP 加密和解密文件以及。AWS Transfer Family

主题

步骤 2：配置执行角色

创建一个 AWS Identity and Access Management (IAM) 执行角色，Transfer Family 可以使用该角色启动工作流程。适用于工作流程的 IAM 策略中描述了创建执行角色的过程。

> **Note**
>
> 在创建执行角色时，请务必在执行角色和 Transfer Family 之间建立信任关系，如建立信任关系中所述。

以下执行角色策略包含成功执行您将在本教程中创建的工作流程所需的所有权限。要使用此示例策略，请将 *user input placeholders* 替换为您自己的信息。将*DOC-EXAMPLE-BUCKET*替换为您要上传加密文件的 Amazon S3 存储桶的名称。

> **Note**
>
> 并非每个工作流程都需要此示例中列出的每个权限。您可以根据特定工作流程中的步骤类型来限制权限。使用预定义的步骤中描述了每种预定义步骤类型所需的权限。自定义步骤的 IAM 权限中描述了每种自定义步骤所需的权限。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "WorkflowsS3Permissions",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectTagging",
                "s3:GetObjectVersion",
                "s3:PutObject",
                "s3:PutObjectTagging",
                "s3:ListBucket",
                "s3:PutObjectTagging",
                "s3:PutObjectVersionTagging",
                "s3:DeleteObjectVersion",
                "s3:DeleteObject"
            ],
            "Resource": ["arn:aws:s3:::DOC-EXAMPLE-BUCKET/*",
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET"]
            "Condition": {
                "StringEquals": {
                    "s3:RequestObjectTag/Archive": "yes"
                }
```

```
            }
        },
        {
            "Sid": "DecryptSecret",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": "arn:aws:secretsmanager:region:account-id:secret:aws/transfer/
*"
        }
    ]
}
```

步骤 2：创建托管工作流程

现在，您需要创建一个包含解密步骤的工作流程。

创建一个包含解密步骤的工作流程。

1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在左侧的导航窗格中，选择工作流程，然后选择创建工作流程。
3. 输入以下详细信息：
   - 例如，输入描述**Decrypt workflow example**。
   - 在标称步骤部分中，选择添加步骤。
4. 对于选择步骤类型，选择解密文件，然后选择下一步。
5. 在配置参数对话框中，指定以下内容：
   - 例如，输入描述性步骤名称**decrypt-step**。步骤名称中不允许使用空格。
   - 对于解密文件的目标，请选择 Amazon S3。
   - 对于目标存储桶名称，请选择您在步骤 1 中创建的 IAM 策略*DOC-EXAMPLE-BUCKET*中指定的相同的 Amazon S3 存储桶。
   - 在目标密钥前缀中，输入要在目标存储桶中存储解密文件的前缀（文件夹）的名称，例如，**decrypted-files/**。

> **Note**
>
> 请务必在前缀中添加一个尾部斜杠 (**/**)。

- 在本教程中，请清除覆盖现有文件。清除此设置后，如果您尝试解密与现有文件同名的文件，则工作流处理将停止，并且不会处理新文件。

选择下一步，进入下一个审核屏幕。

6. 审核该步骤的详细信息。如果一切正确，请选择创建步骤。
7. 您的工作流程只需要单个解密步骤，因此无需配置其他步骤。选择创建工作流程以创建新工作流程。

记下新工作流程的工作流程 ID。下一步骤中，您需要用到此 ID。本教程使用 *w-1234abcd5678efghi* 作为示例工作流程 ID。

步骤 3：将工作流程添加至服务器并创建用户

现在您已经有了带有解密步骤的工作流程，您必须将其与 Transfer Family 服务器相关联。本教程介绍如何将工作流程附加至现有 Transfer Family 服务器。或者，您可以创建新的服务器以用于您的工作流程。

将工作流程附加到服务器后，必须创建一个可以通过 SFTP 连接到服务器并触发工作流程运行的用户。

配置 Transfer Family 服务器以运行工作流程

1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在左侧导航窗格中，选择服务器，然后从列表中选择服务器。确保此服务器支持 SFTP 协议。
3. 在服务器的详细信息页面上，向下滚动到其他详细信息部分，然后选择编辑。
4. 在编辑其他详细信息页面的托管工作流程部分，选择您的工作流程，然后选择相应的执行角色。
   - 对于完成文件上传的工作流程，请选择您在步骤 2：创建托管工作流程中创建的工作流程，例如**w-1234abcd5678efghi**。
   - 对于托管工作流程执行角色，选择您在步骤 2：配置执行角色中创建的 IAM 角色。
5. 滚动到页面底部并选择保存以保存您的更改。

记下您正在使用的服务器的 ID。用于存储 PGP AWS Secrets Manager 密钥的密钥的名称部分基于服务器 ID。

添加可以触发工作流程的用户

1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在左侧导航窗格中，选择服务器，然后选择您要用于解密工作流程的服务器。
3. 在服务器的详细信息页面上，向下滚动到用户部分，然后选择添加用户。
4. 对于您的新用户，请输入以下详细信息：
   - 对于用户名，输入 **`decrypt-user`**。
   - 对于角色，请选择可以访问您的服务器的用户角色。
   - 对于主目录，选择您之前使用的 Amazon S3 存储桶，例如*`DOC-EXAMPLE-BUCKET`*。
   - 对于 SSH 公有密钥，请粘贴与您拥有的私有密钥相对应的公有密钥。有关更多信息，请参阅 为服务托管用户生成 SSH 密钥。
5. 选择添加以保存您的新用户。

记下您在这台服务器上的 Transfer Family 用户的名称。该密钥部分基于用户的名称。为简单起见，本教程使用了服务器的任何用户均可使用的默认密钥。

步骤 2：创建 PGP 密钥对

使用支持的 PGP 客户端之一以生成 PGP 密钥对。有关此过程的详细介绍，请参阅生成 PGP 密钥。

生成 PGP 密钥对

1. 在本教程中，您可以使用 gpg (GnuPG) 版本 2.0.22 客户端生成使用 RSA 作为加密算法的 PGP 密钥对。对于此客户端，运行如下命令，并提供电子邮件地址和密码。您可以使用任何您喜欢的姓名或电子邮件地址。请务必记住所使用的值，因为本教程稍后需要输入这些值。

   ```
   gpg --gen-key
   ```

> **Note**
>
> 如果您使用的版本是GnuPG 2.3.0 或以上，则必须运行 `gpg --full-gen-key`。当提示输入要创建的密钥类型时，请选择 RSA 或 ECC。但是，如果您选择 ECC，请确保为椭圆曲线选择NIST或BrainPool 请勿选择Curve 25519。

2. 通过运行以下命令导出私有密钥。将*user@example.com*替换为生成密钥时使用的电子邮件地址。

```
gpg --output workflow-tutorial-key.pgp --armor --export-secret-key user@example.com
```

此命令将私有密钥导出到**workflow-tutorial-key.pgp**文件中。您可以随意命名输出文件。您也可以在将私有密钥文件添加到 AWS Secrets Manager后删除该文件。

步骤 5：将 PGP 私有密钥存储在 AWS Secrets Manager中

您需要以非常具体的方式将私有密钥存储在 Secrets Manager 中，以便工作流程在对上传的文件运行解密步骤时可以找到私有密钥。

> **Note**
>
> 当你在 Secret AWS 账户 s Manager 中存储密钥时，会产生费用。有关定价的信息，请参阅 AWS Secrets Manager 定价。

在 Secrets Manager 中存储 PGP 私有密钥

1. 登录 AWS Management Console 并打开 AWS Secrets Manager 控制台，网址为 https://console.aws.amazon.com/secretsmanager/。
2. 在左侧导航窗格中，选择密钥。
3. 在密钥页面，选择存储新密钥。
4. 在选择密钥类型页面上，对于密钥类型，选择其他类型密钥。
5. 在键/值对部分，选择键/值选项卡。
   - 键 — 输入**PGPPrivateKey**。
   - 值 — 将您的私有密钥文本粘贴至值字段。

6. 选择添加行，然后在密钥/值对部分选择密钥/值选项卡。
   - 键 — 输入**PGPPassphrase**。
   - 值 — 输入您在步骤 2：创建 PGP 密钥对中生成 PGP 密钥对时使用的密码。
7. 选择下一步。
8. 在配置密钥页面，输入密钥的名称和描述。在本教程中，您可以创建一个所有用户都可以使用的默认密钥。假设服务器 ID 是 *s-11112222333344445*，则命名密钥**aws/transfer/*s-11112222333344445*/@pgp-default**。将*s-11112222333344445*替换为 Transfer Family 服务器的 ID。输入密钥的描述。

   > **Note**
   >
   > 要仅为之前创建的用户创建密钥，请为该密钥命名**aws/transfer/*s-11112222333344445*/decrypt-user**。

9. 选择下一步，接受配置轮换页面的默认设置。然后选择下一步。
10. 在审核页面，选择存储以创建和存储密钥。

有关将 PGP 私钥添加到 Secrets Manager 的更多信息，请参阅用于 AWS Secrets Manager 存储 PGP 密钥。

步骤 6：加密文件

使用该gpg程序对文件进行加密，以便在工作流程中使用。要加密文件，请运行以下命令：

```
gpg -e -r marymajor@example.com --openpgp testfile.txt
```

在运行此命令之前，请注意以下事项：

- 对于-r参数，请*marymajor@example.com*替换为创建 PGP 密钥对时使用的电子邮件地址。
- --openpgp标记是可选的。此标记使加密文件符合 OpenPGP RFC4880 标准。
- 此命令将创建一个名为**testfile.txt.gpg**的文件，其位置与**testfile.txt**相同。

步骤 7：运行工作流程并查看结果

要运行工作流程，您需要使用在步骤 3 中创建的用户连接到 Transfer Family 服务器。然后，您可以查看您在步骤 2.5 中指定的 Amazon S3 存储桶，配置目标参数以查看解密后的文件。

运行解密工作流程

1. 打开命令终端。
2. 运行以下命令，替换*your-endpoint*为实际端点和*transfer-key*为用户的 SSH 私有密钥：

   ```
   sftp -i transfer-key decrypt-user@your-endpoint
   ```

   例如，如果私有密钥存储在~/.ssh/decrypt-user中，而您的端点存储在s-11112222333344445.server.transfer.us-east-2.amazonaws.com中，则命令如下所示：

   ```
   sftp -i  ~/.ssh/decrypt-user decrypt-user@s-11112222333344445.server.transfer.us-east-2.amazonaws.com
   ```

3. 运行 pwd 命令。如果成功，此命令将返回以下内容：

   ```
   Remote working directory: /DOC-EXAMPLE-BUCKET/decrypt-user
   ```

   您的目录反映了 Amazon S3 存储桶的名称。
4. 运行如下命令来上传文件并触发要运行的工作流程：

   ```
   put testfile.txt.gpg
   ```

5. 对于解密文件的目标，您在创建工作流程时指定了decrypted-files/文件夹。现在，您可以导航到该文件夹并列出内容。

   ```
   cd ../decrypted-files/
   ls
   ```

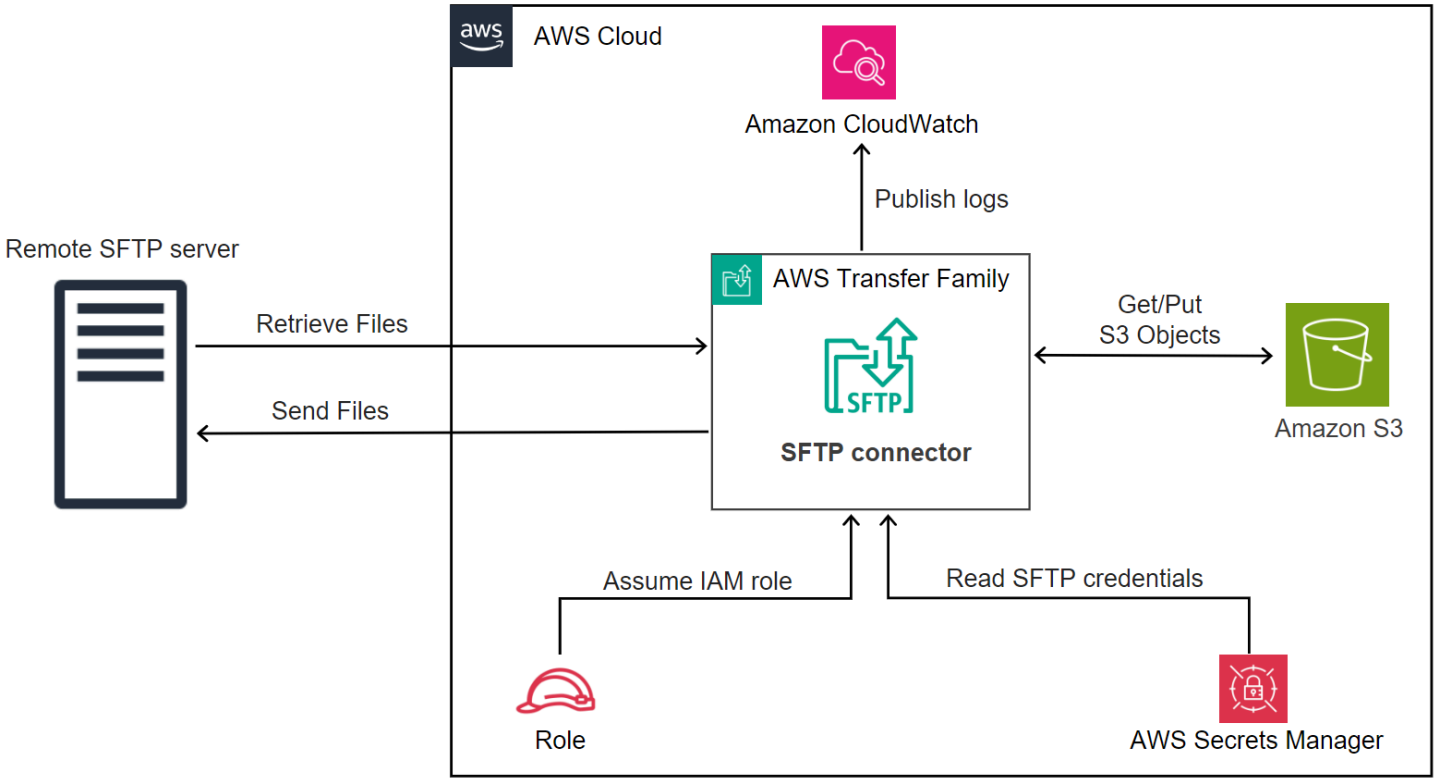
   如果成功，则ls命令将列出testfile.txt文件。您可以下载此文件并验证它是否与之前加密的原始文件相同。

设置和使用 SFTP 连接器

连接器的目的是在您的 AWS 存储设备和合作伙伴的 SFTP 服务器之间建立关系。您可以将文件从 Amazon S3 发送到合作伙伴拥有的外部目的地。您也可以使用 SFTP 连接器从合作伙伴的 SFTP 服务器检索文件。

本教程说明如何设置 SFTP 连接器，然后在 Amazon S3 存储设备和 SFTP 服务器之间传输文件。

SFTP 连接器从中检索 SFTP 凭据 AWS Secrets Manager ，以便对远程 SFTP 服务器进行身份验证并建立连接。连接器向远程服务器发送文件或从远程服务器检索文件，并将文件存储在 Amazon S3 中。IAM 角色用于允许访问 Amazon S3 存储桶和存储在 Secrets Manager 中的证书。而且你可以登录到亚马逊 CloudWatch。



主题

- 步骤 1：创建必要的支持资源
- 步骤 2：创建和测试 SFTP 连接器
- 步骤 3：使用 SFTP 连接器发送和检索文件
- 创建用作远程 SFTP 服务器的 Transfer Family 服务器的步骤

步骤 1：创建必要的支持资源

您可以使用 SFTP 连接器在 Amazon S3 和任何远程 SFTP 服务器之间复制文件。在本教程中，我们使用 AWS Transfer Family 服务器作为远程 SFTP 服务器。我们需要创建和配置以下资源：

- 创建 Amazon S3 存储桶以在您的 AWS 环境中存储文件，以及从远程 SFTP 服务器发送和检索文件：。创建 Amazon S3 存储桶

- 在 Secrets Manager 中创建用于访问 Amazon S3 存储空间和我们的密钥的 AWS Identity and Access Management 角色：创建具有必要权限的 IAM 角色。
- 创建使用 SFTP 协议的 Transfer Family 服务器，以及使用 SFTP 连接器在 SFTP 服务器之间传输文件或从 SFTP 服务器传输文件的服务管理用户：。创建 Transfer Family SFTP 服务器和一个用户
- 创建一个 AWS Secrets Manager 密钥来存储 SFTP 连接器用于登录远程 SFTP 服务器的凭据：。创建密钥并将其存储在 AWS Secrets Manager

创建 Amazon S3 存储桶

创建 Amazon S3 存储桶

1. 登录 AWS Transfer Family 主机，网址为 https://console.aws.amazon.com/s3/。
2. 选择一个地区并输入名称。

   在本教程中，我们的存储桶位于中**US East (N. Virginia) us-east-1**，名称为**sftp-server-storage-east**。
3. 接受默认值并选择创建存储桶。

有关创建 Amazon S3 存储桶的完整详细信息，请参阅如何创建 S3 存储桶？ 在《Amazon 简单存储服务用户指南》中。

创建具有必要权限的 IAM 角色

对于访问角色，创建具有以下权限的策略。

以下示例授予访问*### S3 ## DOC-EXAMPLE-BUCK* ET 以及存储在 Secrets Manager 中的指定密钥所需的权限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
        "Sid": "AllowListingOfUserFolder",
        "Action": [
            "s3:ListBucket",
            "s3:GetBucketLocation"
        ],
        "Effect": "Allow",
```

```
        "Resource": [
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
        ]
    },
    {
        "Sid": "HomeDirObjectAccess",
        "Effect": "Allow",
        "Action": [
            "s3:PutObject",
            "s3:GetObject",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion",
            "s3:GetObjectVersion",
            "s3:GetObjectACL",
            "s3:PutObjectACL"
        ],
        "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
    },
    {
        "Sid": "GetConnectorSecretValue",
        "Effect": "Allow",
        "Action": [
            "secretsmanager:GetSecretValue"
        ],
        "Resource": "arn:aws:secretsmanager:region:account-id:secret:aws/
transfer/SecretName-6RandomCharacters"
    }
  ]
}
```

按如下方式替换项目：

- 对于 *DOC-EXAMPLE-BUCKET###*使用。**s3-storage-east**
- 对于*##*，本教程使用**us-east-1**。
- 要获取*## ID#*请使用您的 AWS 账户 ID。
- 对于 *SecretName-6 RandomCharacters*，我们代表**using sftp-connector1**名字（你将有自己的六个随机字符作为你的秘密）。

您还必须确保此角色包含信任关系，允许连接器在处理用户的转移请求时访问您的资源。有关建立信任关系的详细信息，请参阅 建立信任关系。

> **Note**
>
> 要查看我们在本教程中使用的角色的详细信息，请参阅用户和访问角色的组合。

创建密钥并将其存储在 AWS Secrets Manager

我们需要在 Secrets Manager 中存储一个密钥来存储你的 SFTP 连接器的用户凭证。您可以使用密码、SSH 私钥或两者兼而有之。在本教程中，我们使用的是私钥。

> **Note**
>
> 当你在 Secret AWS 账户 s Manager 中存储密钥时，会产生费用。有关定价的信息，请参阅 AWS Secrets Manager 定价。

在开始存储密钥的过程之前，请检索并格式化您的私钥。私钥必须与在远程 SFTP 服务器上为用户配置的公钥相对应。在本教程中，私钥必须对应于我们用作远程服务器的 Transfer Family SFTP 服务器上为测试用户存储的公钥。

为此，请运行以下命令：

```
jq -sR . path-to-private-key-file
```

例如，如果您的私钥文件位于中`~/.ssh/sftp-testuser-privatekey`，则命令如下所示。

```
jq -sR . ~/.ssh/sftp-testuser-privatekey
```

这会将正确格式的密钥（带有嵌入的换行符）输出到标准输出。将此文本复制到某个地方，因为您需要将其粘贴到以下步骤中（在步骤 6 中）。

若要在 Secrets Manager 中存储 SFTP 连接器的用户凭证

1. 登录 AWS Management Console 并打开 AWS Secrets Manager 控制台，网址为 https://console.aws.amazon.com/secretsmanager/。
2. 在左侧导航窗格中，选择密钥。
3. 在密钥页面，选择存储新密钥。

4. 在选择密钥类型页面上，对于密钥类型，选择其他类型密钥。
5. 在键/值对部分，选择键/值选项卡。
   - 键 — 输入**Username**。
   - valu e — 输入我们的用户名**sftp-testuser**。
6. 要输入密钥，我们建议您使用纯文本选项卡。
   a. 选择添加行，然后输入**PrivateKey**。
   b. 选择纯文本选项卡。该字段现在包含以下文本：

```
{"Username":"sftp-testuser","PrivateKey":""}
```

   c. 在空双引号 (“”) 之间粘贴私钥文本（之前保存）。

   屏幕应如下所示（关键数据显示为灰色）。

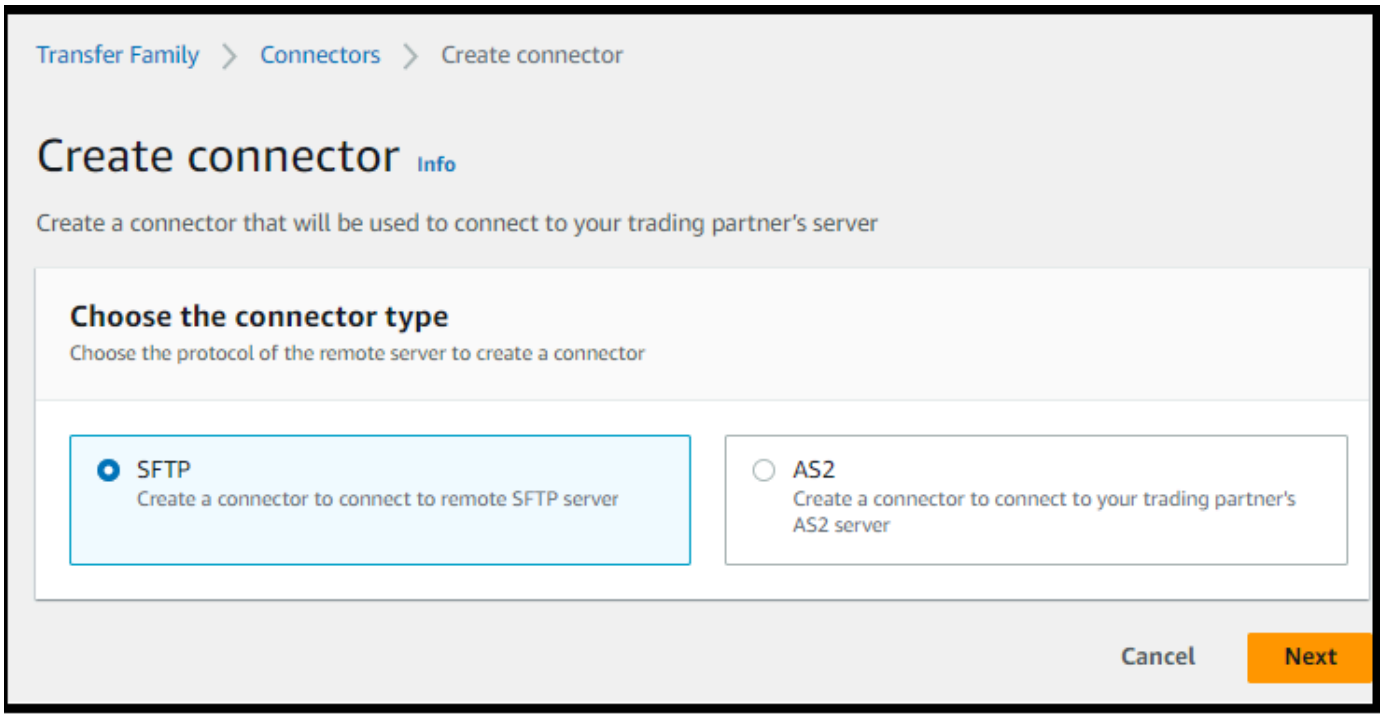
7. 选择下一步。
8. 在配置密钥页面上，输入您的密钥的名称。在本教程中，我们命名了秘密**aws/transfer/sftp-connector1**。
9. 选择下一步，接受配置轮换页面的默认设置。然后选择下一步。
10. 在审核页面，选择存储以创建和存储密钥。

步骤 2：创建和测试 SFTP 连接器

在本节中，我们将创建一个使用我们之前创建的所有资源的 SFTP 连接器。有关更多详细信息，请参阅配置 SFTP 连接器。

若要创建 SFTP 连接器

1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在导航窗格中，选择连接，然后选择创建连接器。
3. 选择 SFTP 作为连接器类型，以创建 SFTP 连接器，然后选择“下一步”。



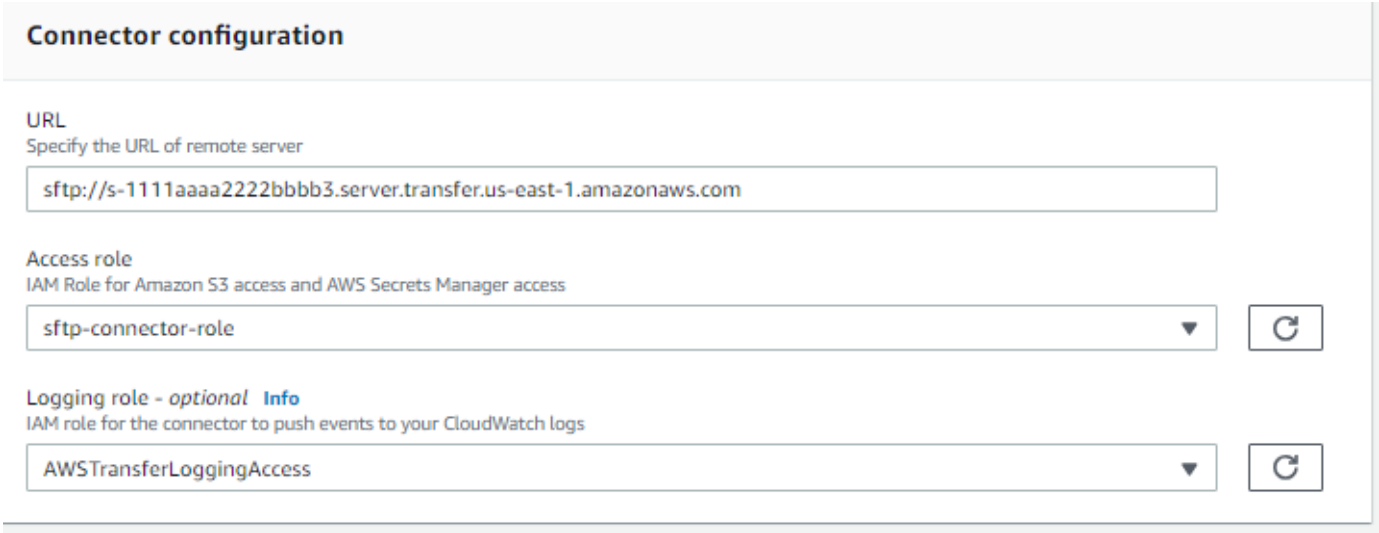
4. 在连接器配置部分中，提供以下信息：

   - 在 URL 中，输入远程 SFTP 服务器的 URL。在本教程中，我们输入用作远程 SFTP 服务器的 Transfer Family 服务器的 URL。

     ```
     sftp://s-1111aaaa2222bbbb3.server.transfer.us-east-1.amazonaws.com
     ```

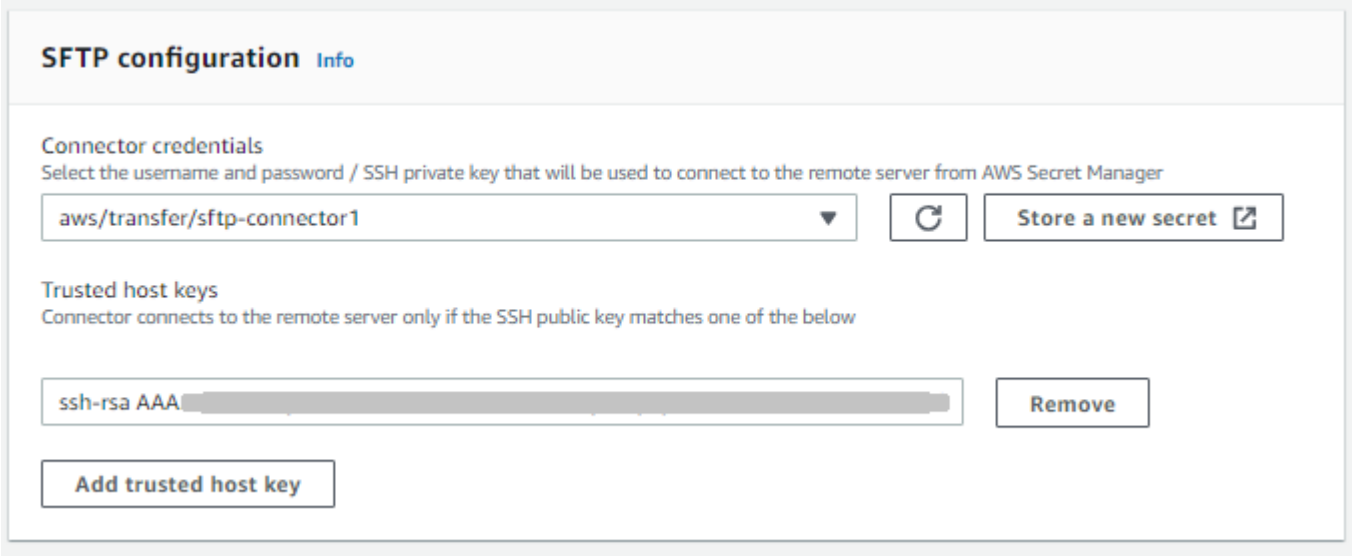
     将 *1111aaaa2222bbb3* 替换成你的 Transfer Family 服务器 ID。
   - 对于访问角色，请输入我们之前创建的角色**sftp-connector-role**。
   - 对于 “记录” 角色，选择**AWSTransferLoggingAccess**。

> **Note**
>
> AWSTransferLoggingAccess 是一个 AWS 托管策略。中详细介绍了该政策AWS 托管策略： AWSTransferLoggingAccess。



5. 在 SFTP Configuration 面板中提供以下信息：

   - 对于连接器凭据，请选择包含 SFTP 凭据的 Secrets Manager 资源的名称。在本教程中，选择**`aws/transfer/sftp-connector1`**。
   - 对于受信任的主机密钥，请粘贴主机密钥的公共部分。您可以通过`ssh-keyscan`为 SFTP 服务器运行来检索此密钥。有关如何格式化和存储可信主机密钥的详细信息，请参阅SftpConnectorConfig数据类型文档。



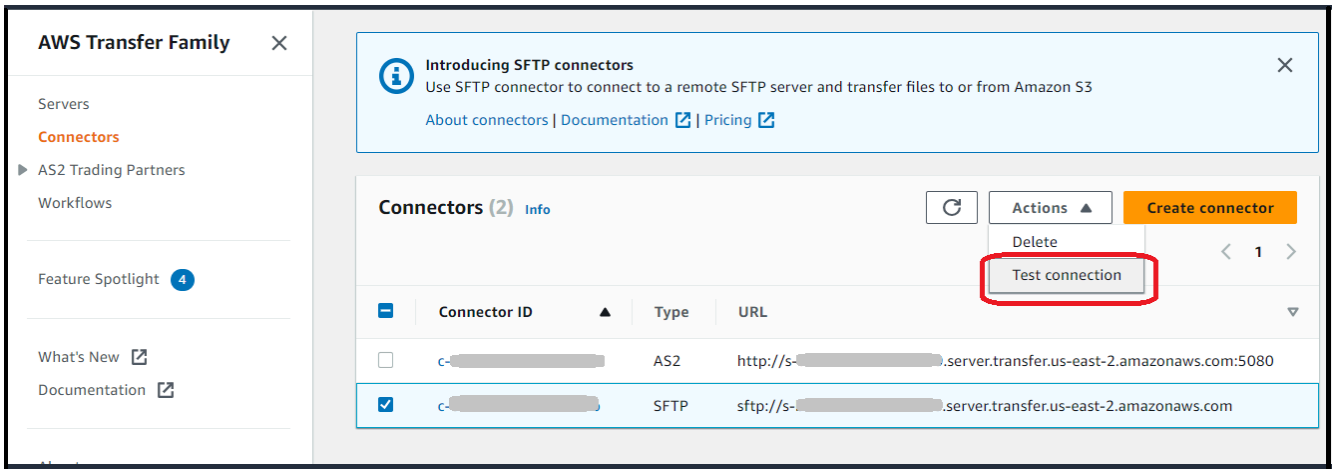
6. 确认所有设置后，选择创建连接器以创建 SFTP 连接器。

创建 SFTP 连接器后，我们建议您在尝试使用新连接器传输任何文件之前对其进行测试。

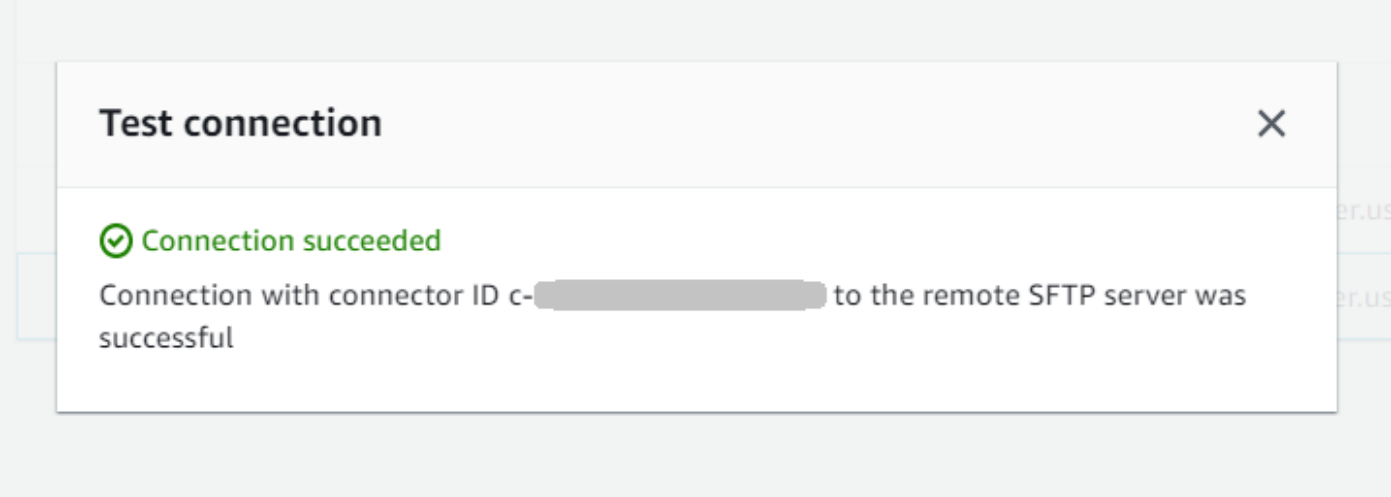
Test a connector using the console

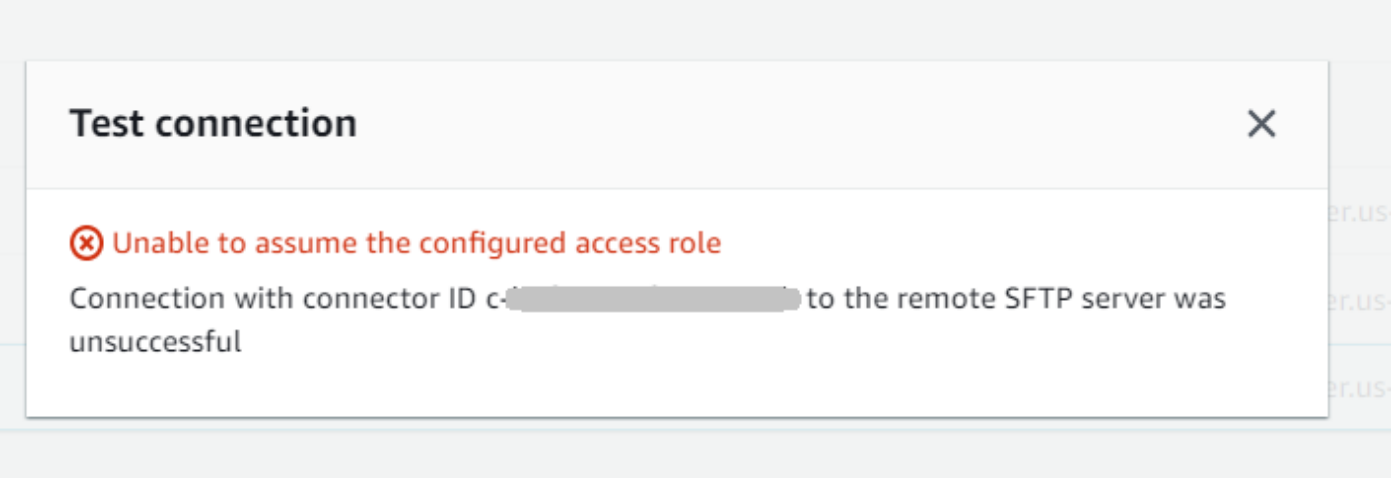
若要测试 SFTP 连接器

1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在左侧导航窗格中，选择连接器，然后选择一个连接器。
3. 从 操作菜单中选择 测试连接。



系统会返回一条消息，指示测试是通过还是失败。如果测试失败，系统会根据测试失败的原因提供错误消息。





Test a connector using the CLI

要使用测试连接器 AWS Command Line Interface，请在命令提示符下运行以下命令（将 c*onnector-i* d 替换为实际的连接器 ID）：

```
aws transfer test-connection --connector-id c-connector-id
```

如果测试成功，则返回以下几行：

```
{
   "Status": "OK",
   "StatusMessage": "Connection succeeded"
}
```

如果测试失败，您会收到一条描述性错误消息，例如：

```
{
   "Status": "ERROR",
   "StatusMessage": "Unable to assume the configured access role"
}
```

步骤 3：使用 SFTP 连接器发送和检索文件

为简单起见，我们假设您的 Amazon S3 存储桶中已经有文件。

> **Note**
>
> 本教程使用了 Amazon S3 存储桶作为源存储位置和目标存储位置。如果您的 SFTP 服务器不使用 Amazon S3 存储，那么无论您在以下命令`sftp-server-storage-east`中看到的任何地方，都可以将路径替换为可从 SFTP 服务器访问的文件位置的路径。

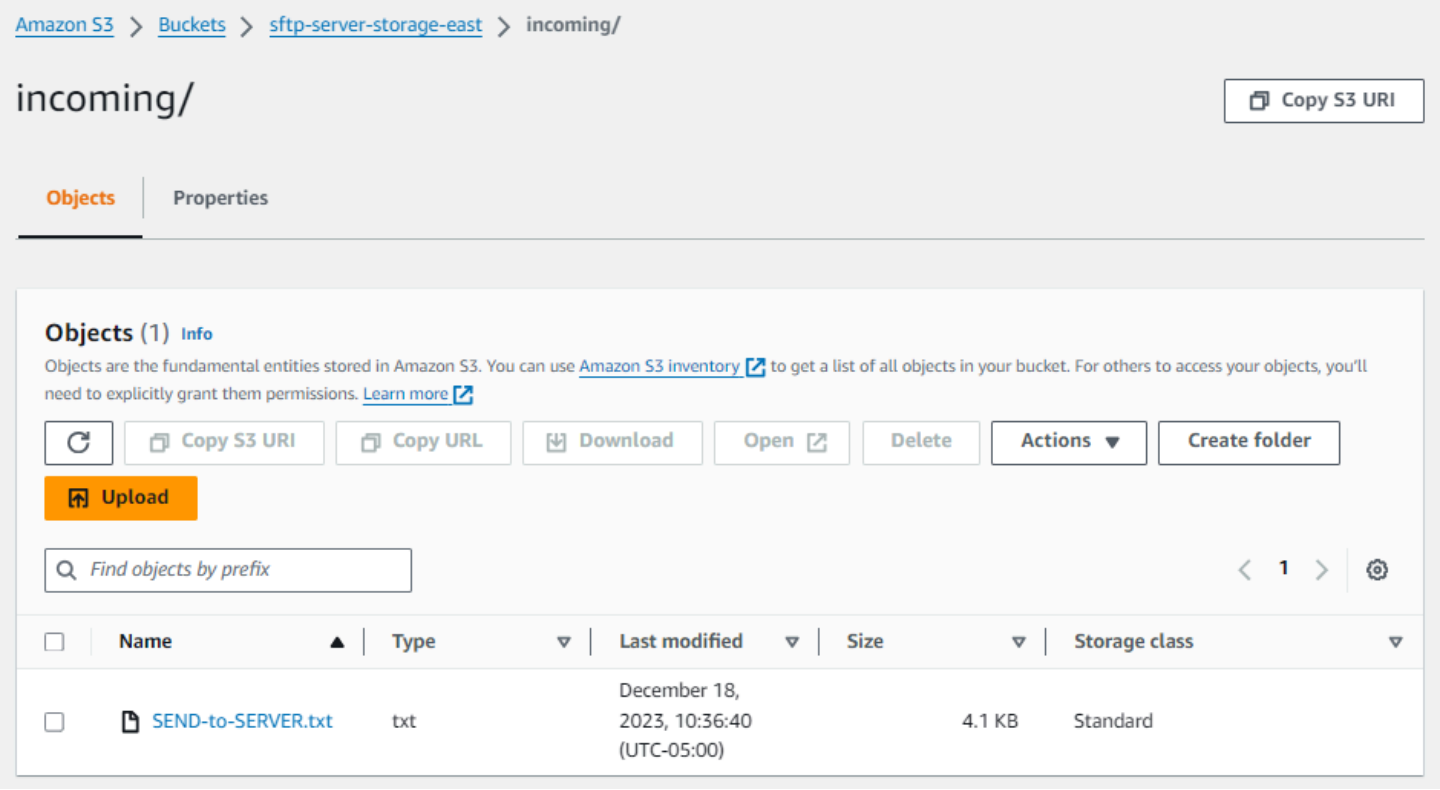
- 我们将名为 Amazon S3 存储的文件发送`SEND-to-SERVER.txt`到 SFTP 服务器。
- 我们将名为的文件`RETRIEVE-to-S3.txt`从 SFTP 服务器检索到 Amazon S3 存储空间。

> **Note**
>
> 在以下命令中，将*`### ID`* 替换为您的连接器 ID。

首先，我们将文件从 Amazon S3 存储桶发送到远程 SFTP 服务器。在命令提示符下，运行以下命令：

```
aws transfer start-file-transfer --connector-id c-connector-id --send-file-paths "/s3-
storage-east/SEND-to-SERVER.txt" /
   --remote-directory-path "/sftp-server-storage-east/incoming"
```

你的`sftp-server-storage-east`存储桶现在应该是这样的。



如果您没有按预期看到该文件，请检查您的 CloudWatch 日志。

查看您的 CloudWatch 日志

1. 打开亚马逊 CloudWatch 控制台，网址为 https://console.aws.amazon.com/cloudwatch/
2. 从左侧导航菜单中选择 “日志组”。
3. 在搜索栏中输入您的连接器 ID 以查找您的日志。
4. 选择从搜索中返回的日志流。
5. 展开最新的日志条目。

如果成功，则日志条目如下所示：

```
{
    "operation": "SEND",
    "timestamp": "2023-12-18T15:26:57.346283Z",
    "connector-id": "connector-id",
    "transfer-id": "transfer-id",
    "file-transfer-id": "transfer-id/file-transfer-id",
    "url": "sftp://server-id.server.transfer.us-east-1.amazonaws.com",
    "file-path": "/s3-storage-east/SEND-to-SERVER.txt",
```

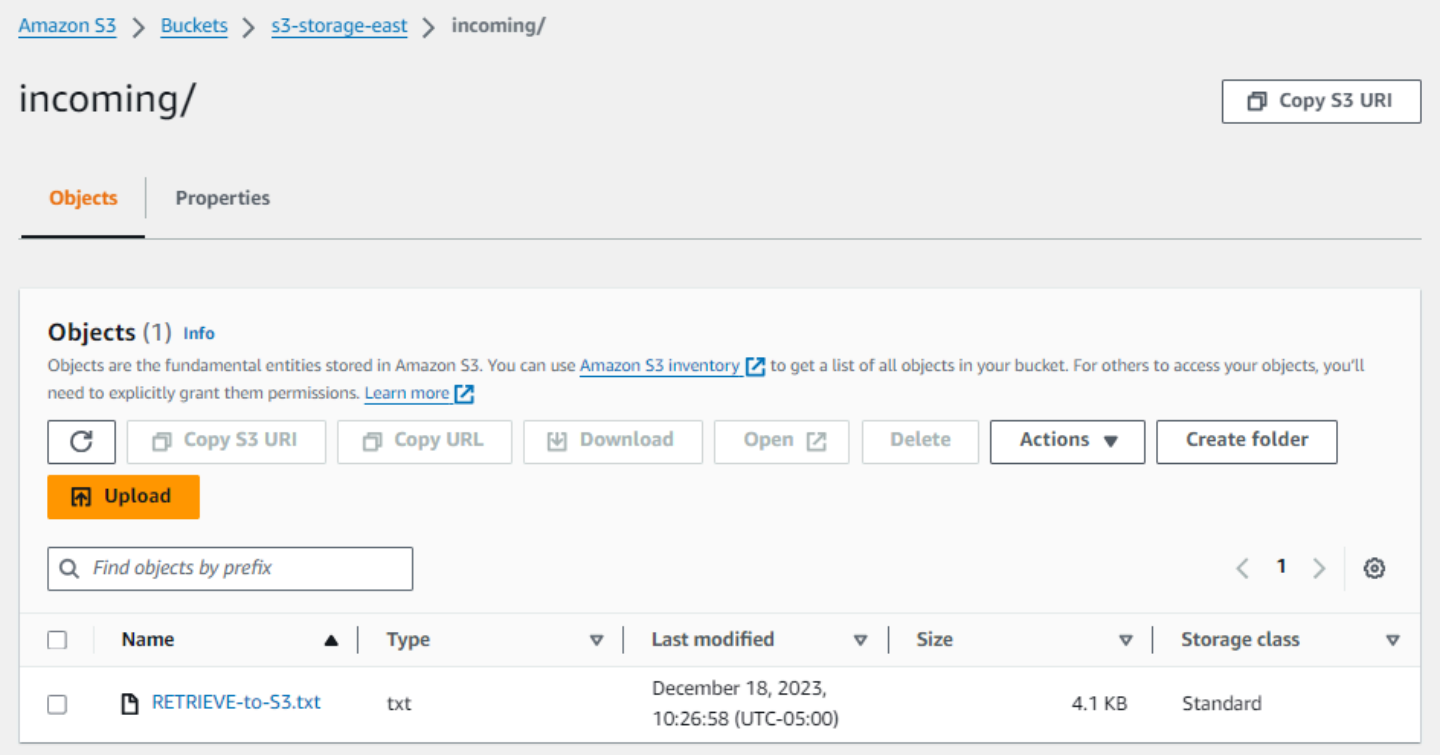
```
    "status-code": "COMPLETED",
    "start-time": "2023-12-18T15:26:56.915864Z",
    "end-time": "2023-12-18T15:26:57.298122Z",
    "account-id": "500655546075",
    "connector-arn": "arn:aws:transfer:us-east-1:500655546075:connector/connector-id",
    "remote-directory-path": "/sftp-server-storage-east/incoming"
}
```

如果文件传输失败，则日志条目将包含一条指明问题的错误消息。常见的错误原因是 IAM 权限问题和文件路径不正确。

接下来，我们将文件从 SFTP 服务器检索到 Amazon S3 存储桶中。在命令提示符下，运行以下命令：

```
aws transfer start-file-transfer --connector-id c-connector-id --retrieve-file-paths "/
sftp-server-storage-east/RETRIEVE-to-S3.txt" --local-directory-path "/s3-storage-east/
incoming"
```

如果传输成功，则您的 Amazon S3 存储桶将包含传输的文件，如下所示。



如果成功，则日志条目如下所示：

```
{
    "operation": "RETRIEVE",
```

```
    "timestamp": "2023-12-18T15:36:40.017800Z",
    "connector-id": "c-connector-id",
    "transfer-id": "transfer-id",
    "file-transfer-id": "transfer-id/file-transfer-id",
    "url": "sftp://s-server-id.server.transfer.us-east-1.amazonaws.com",
    "file-path": "/sftp-server-storage-east/RETRIEVE-to-S3.txt",
    "status-code": "COMPLETED",
    "start-time": "2023-12-18T15:36:39.727626Z",
    "end-time": "2023-12-18T15:36:39.895726Z",
    "account-id": "500655546075",
    "connector-arn": "arn:aws:transfer:us-east-1:500655546075:connector/c-connector-
id",
    "local-directory-path": "/s3-storage-east/incoming"
}
```

创建用作远程 SFTP 服务器的 Transfer Family 服务器的步骤

接下来，我们将概述创建用作本教程远程 SFTP 服务器的 Transfer Family 服务器的步骤。请注意以下几点：

- 我们使用 Transfer Family 服务器来表示远程 SFTP 服务器。典型的 SFTP 连接器用户拥有自己的远程 SFTP 服务器。请参阅 创建 Transfer Family SFTP 服务器和一个用户。
- 因为我们使用的是 Transfer Family 服务器，所以我们也使用的是服务管理的 SFTP 用户。而且，为简单起见，我们将该用户访问 Transfer Family 服务器所需的权限与他们使用连接器所需的权限相结合。同样，大多数 SFTP 连接器用例都有单独的 SFTP 用户，该用户与 Transfer Family 服务器无关。请参阅 创建 Transfer Family SFTP 服务器和一个用户。
- 在本教程中，由于我们在远程 SFTP 服务器上使用 Amazon S3 存储，因此我们需要创建第二个存储桶**s3-storage-east**，以便我们可以将文件从一个存储桶传输到另一个存储桶。

创建 Transfer Family SFTP 服务器和一个用户

大多数用户不需要创建 Transfer Family SFTP 服务器和用户，因为您已经有一台包含用户的 SFTP 服务器，并且您可以使用此服务器来往传输文件。但是，在本教程中，为了简单起见，我们使用了 Transfer Family 服务器来充当远程 SFTP 服务器。

按照中所述创建启用 SFTP 的服务器的步骤创建服务器和步骤 3：添加服务托管用户添加用户。以下是我们在本教程中使用的用户详细信息：

- 创建您的服务管理用户，`sftp-testuser`。

  - 将主目录设置为 `/sftp-server-storage-east/sftp-testuser`
  - 创建用户时，即存储公钥。稍后，当你在 Secrets Manager 中创建密钥时，你需要提供相应的私钥。
- 角色：`sftp-connector-role`。在本教程中，我们对 SFTP 用户和访问 SFTP 连接器使用相同的 IAM 角色。在为组织创建连接器时，您可能有不同的用户和访问角色。
- 服务器主机密钥：创建连接器时需要使用服务器主机密钥。您可以通过`ssh-keyscan`为服务器运行来检索此密钥。例如，如果您的服务器 ID 为`s-1111aaaa2222bbbb3`，且其终端节点位于中`us-east-1`，则以下命令将检索服务器主机密钥：

  ```
  ssh-keyscan s-1111aaaa2222bbbb3.server.transfer.us-east-1.amazonaws.com
  ```

  将此文本复制到某个地方，因为您需要将其粘贴到步骤 2：创建和测试 SFTP 连接器程序中。

用户和访问角色的组合

在本教程中，我们使用的是单一的组合角色。我们既对 SFTP 用户使用此角色，也用于访问连接器。以下示例包含此角色的详细信息，以备您要执行本教程中的任务时使用。

以下示例授予访问我们在 Amazon S3 中的两个存储桶以及存储在 Secrets Manager 中的名为`aws/transfer/sftp-connector1`的密钥所需的权限。在本教程中，这个角色被命名为`sftp-connector-role`。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowListingOfUserFolder",
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::sftp-server-storage-east",
                "arn:aws:s3:::s3-storage-east"
            ]
        },
        {
            "Sid": "HomeDirObjectAccess",
```

```
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:DeleteObject",
                "s3:DeleteObjectVersion",
                "s3:GetObjectVersion",
                "s3:GetObjectACL",
                "s3:PutObjectACL"
            ],
            "Resource": [
                "arn:aws:s3:::sftp-server-storage-east/*",
                "arn:aws:s3:::s3-storage-east/*"
            ]
        },
        {
            "Sid": "GetConnectorSecretValue",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": "arn:aws:secretsmanager:us-east-1:500655546075:secret:aws/
transfer/sftp-connector1-6RandomCharacters"
        }
    ]
}
```

有关为 Transfer Family 创建角色的完整详细信息，请按照中的创建用户角色步骤创建角色。

将 Amazon API Gateway 方法设置为自定义身份提供商

本教程演示如何设置 Amazon API Gateway 方法并将其用作自定义身份提供商将文件上传到AWS Transfer Family服务器。本教程仅使用基本堆栈模板和其他基本功能作为示例。

主题

- 步骤 5：测试 SFTP 连接和文件传输
- 步骤 6：限制对存储桶的访问权限
- 如果使用 Amazon EFS，请更新 Lambda

先决条件

在AWS CloudFormation中创建 Transfer Family 资源之前，请创建您的存储和用户角色。

要指定存储并创建用户角色

1. 根据您使用的存储，请参阅以下文档：
   - 要创建 Amazon S3 存储桶，请参阅Amazon Simple Storage Service 用户指南中的如何创建 S3 存储桶？
   - 要创建 Amazon EFS 文件系统，请参阅配置 Amazon EFS 文件系统。
2. 要创建用户角色，请参阅创建 IAM 角色和策略

在下一部分中创建AWS CloudFormation堆栈时，您将输入存储和用户角色的详细信息。

步骤 1：创建 CloudFormation 堆栈

通过提供的模板创建AWS CloudFormation堆栈

1. 打开 AWS CloudFormation 控制台，地址：https://console.aws.amazon.com/cloudformation。
2. 选择创建堆栈，然后选择使用新资源（标准）。
3. 在先决条件 — 准备模板窗格，请选择模板已就绪。
4. 复制此链接，即基本堆栈模板，然后将其粘贴到 Amazon S3 URL 字段中。
5. 单击下一步。
6. 指定参数，包括堆栈的名称。务必执行以下操作：
   - 替换UserName和的默认值UserPassword。
   - 对于 UserHomeDirectory，请输入您之前创建的存储（Amazon S3 存储桶或 Amazon EFS 文件系统）的详细信息。
   - 将默认UserRoleArn角色替换为您之前创建的用户角色。AWS Identity and Access Management (IAM) 角色必须具有相应的权限。有关 IAM 角色和存储桶策略示例，请参阅 步骤 6：限制对存储桶的访问权限。

- 如果要使用公钥而不是密码进行身份验证，请在 UserPublicKey1 字段中输入您的公钥。首次使用 SFTP 连接到服务器时，将提供私有密钥而不是密码。

7. 选择下一步，然后在配置堆栈选项页面上再次选择下一步。
8. 查看您正在创建的堆栈的详细信息，然后选择创建堆栈。

> **Note**
>
> 在页面底部的功能下，您必须确认AWS CloudFormation可能会创建 IAM 资源。

步骤 2：检查服务器的 API Gateway 方法配置。

> **Note**
>
> 为了提高安全性，可以配置 Web 应用程序防火墙。AWS WAF 是一种 Web 应用程序防火墙，可让您监视转发到 Amazon API Gateway 的 HTTP 和 HTTPS 请求。有关更多信息，请参阅是一个 Web 应用程序防火墙。。

检查服务器的 API Gateway 方法配置并部署它

1. 打开 API Gateway 控制台，网址为：https://console.aws.amazon.com/apigateway/。
2. 选择 AWS CloudFormation 模板生成的传输自定义身份提供商基本模板 API。
3. 在资源窗格中，选择获取，然后选择 方法请求。
4. 在 操作，选择部署 API。对于部署阶段，选择 prod，然后选择部署。

   成功部署 API Gateway 方法后，在阶段编辑器部分查看其性能。

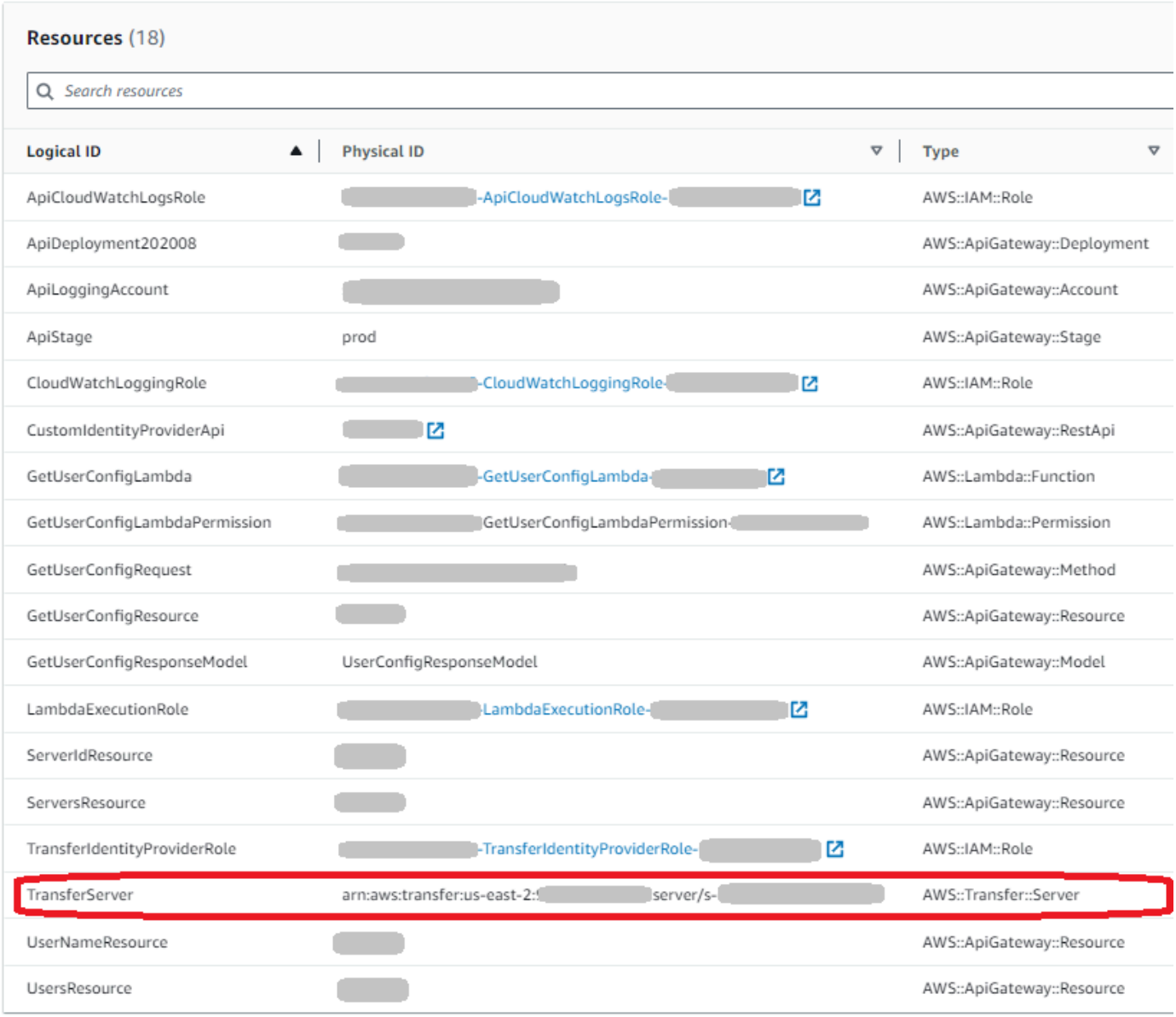
> **Note**
>
> 复制显示在页面顶部的调用 URL 地址。在下一步骤中，您将需要该值。

步骤 3：查看 Transfer Family 服务器详细信息

当您使用模板创建AWS CloudFormation堆栈时，会自动创建一个 Transfer Family 服务器。

要查看您的 Transfer Family 服务器详细信息

1. 打开 AWS CloudFormation 控制台，地址：https://console.aws.amazon.com/cloudformation。
2. 选择您创建的堆栈。
3. 选择资源选项卡。



服务器 ARN 显示在该行的 “物理 ID” 列中。TransferServer服务器 ID 包含在 ARN 中，例如 s-11112222333344445。

4. 通过 https://console.aws.amazon.com/transfer/ 打开AWS Transfer Family控制台，然后在服务器页面上，选择新服务器。

服务器 ID 与中为TransferServer资源显示的 ID 相匹配AWS CloudFormation。

步骤 4：测试您的用户是否可以连接到服务器

要测试您的用户是否可以连接到服务器，请使用 Transfer Family 控制台

1. 打开AWS Transfer Family控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在服务器页面上，选择您的新服务器，选择操作，然后选择测试。
3. 在用户名字段和密码字段中输入登录凭证的文本。这些是您在部署AWS CloudFormation堆栈时设置的值。
4. 对于服务器协议，请选择 SFTP，对于源 IP，请输入**127.0.0.1**。
5. 选择测试。

   如果用户身份验证成功，则测试将返回一个 `StatusCode: 200` HTML 响应和一个包含用户角色和权限详细信息的 JSON 对象。例如：

   ```
   {
       "Response": "{\"Role\": \"arn:aws:iam::123456789012:role/my-user-role\",
   \"HomeDirectory\": \"/${transfer:HomeBucket}/\"}",
       "StatusCode": 200,
       "Message": "",
       "Url": "https://1a2b3c4d5e.execute-api.us-east-2.amazonaws.com/prod/servers/
   s-1234abcd5678efgh0/users/myuser/config"
   }
   ```

   如果测试失败，请将其中一个 API Gateway AWS托管策略添加至您用于 API 的角色中。

步骤 5：测试 SFTP 连接和文件传输

测试 SFTP 连接

1. 在 Linux 或 macOS 设备上，打开命令终端。
2. 根据您是使用密码还是密钥对进行身份验证，输入以下命令之一。

   - 如果您使用的是密码，请输入如下命令：

```
sftp -o PubkeyAuthentication=no myuser@server-
ID.server.transfer.region-code.amazonaws.com
```

出现提示时请输入密码。

- 如果您使用的是密钥对，请输入如下命令：

```
sftp -i private-key-file myuser@server-ID.server.transfer.region-
code.amazonaws.com
```

> **Note**
>
> 对于这些sftp命令，请插入 Transfer Family 服务器所在位置AWS 区域的代码。例如，如果您的服务器位于美国东部（俄亥俄州），请输入**us-east-2**。

3. sftp>出现提示时，请确保您可以上传 (put)、下载 (get) 以及查看目录和文件（pwd和ls）。

步骤 6：限制对存储桶的访问权限

您可以限制谁能够访问特定 Amazon S3 存储桶。以下示例显示了要在 CloudFormation 堆栈和为用户选择的策略中使用的设置。

在此示例中，我们为AWS CloudFormation堆栈设置了以下参数：

- CreateServer: `true`
- UserHomeDirectory: `/myuser-bucket`
- UserName: `myuser`
- UserPassword: `MySuperSecretPassword`

> **Important**
>
> 这是密码示例。在配置 API Gateway 方法时，请务必输入一个强密码。

- UserPublicKey1: *your-public-key*
- UserRoleArn: `arn:aws:iam::`*role-id*`:role/myuser-api-gateway-role`

UserPublicKey1 是您作为公钥/私钥对的一部分生成的公钥。

*role-id*对于您创建的用户角色而言是唯一的。附加到 myuser-api-gateway-role 的策略如下：

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "s3:ListBucket",
            "Resource": "arn:aws:s3:::myuser-bucket"
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObjectAcl",
                "s3:GetObject",
                "s3:DeleteObjectVersion",
                "s3:DeleteObject",
                "s3:PutObjectAcl",
                "s3:GetObjectVersion"
            ],
            "Resource": "arn:aws:s3:::myuser-bucket/*"
        }
    ]
}
```

要使用 SFTP 连接到服务器，请在提示符下输入以下命令之一。

- 如果您使用密码进行身份验证，请运行如下命令：

  sftp -o PubkeyAuthentication=no *myuser*@*transfer-server-ID*.server.transfer.*region-id*.amazonaws.com

  出现提示时请输入密码。

- 如果您使用密钥对进行身份验证，请运行如下命令：

  sftp -i *private-key-file* myuser@*transfer-server-ID*.server.transfer.*region-id*.amazonaws.com

> **Note**
>
> 对于这些sftp命令，请使用 Transfer Family 服务器所在位置AWS 区域的 ID。例如，如果您的服务器位于美国东部（俄亥俄州），请使用us-east-2。

在sftp提示符下，您将被定向到主目录，您可以通过运行pwd命令来查看该目录。例如：

```
sftp> pwd
Remote working directory: /myuser-bucket
```

用户无法查看主目录之上的任何目录。例如：

```
sftp> pwd
Remote working directory: /myuser-bucket
sftp> cd ..
sftp> ls
Couldn't read directory: Permission denied
```

如果使用 Amazon EFS，请更新 Lambda

如果您选择 Amazon EFS 作为 Transfer Family 服务器的存储选项，则需要编辑堆栈的 lambda 函数。

要向 Lambda 函数添加 posix 配置文件

1. 通过 https://console.aws.amazon.com/lambda/ 打开 Lambda 控制台。
2. 选择先前创建的 Lambda 函数。*Lambda ##########-GetUserConfigLambda-lambda ########## CloudFormation #####lambda ##########*
3. 在代码选项卡中，选择 index.js 以显示该函数的代码。
4. 在response中，在Policy和HomeDirectory之间添加以下行：

   PosixProfile: {"Uid": *uid-value*, "Gid": *gid-value*},

   其中，*uid-value* 和 *gid-value* 是分别表示用户 ID 和组 ID 的整数，等于 0 或大于 0。

   例如，添加 Posix 配置文件后，响应字段可能如下所示：

   ```
   response = {
        Role: 'arn:aws:iam::123456789012:role/api-gateway-transfer-efs-role', // The
   user will be authenticated if and only if the Role field is not blank
   ```

```
    Policy: '', // Optional JSON blob to further restrict this user's permissions
    PosixProfile: {"Gid": 65534, "Uid": 65534},
    HomeDirectory: '/fs-fab2c234' // Not required, defaults to '/'
  };
```

设置 AS2 配置

本教程介绍如何使用设置适用性声明 2 (AS2) 配置。 AWS Transfer Family完成此处描述的步骤后，您将拥有一台启用 AS2 的服务器，可以接受来自示例交易伙伴的 AS2 消息。您还将有一个连接器，可用于向示例交易伙伴发送 AS2 消息。

> **Note**
>
> 示例设置的某些部分使用 AWS Command Line Interface (AWS CLI)。如果您尚未安装 AWS CLI，请参阅AWS Command Line Interface 用户指南 AWS CLI中的安装或更新最新版本的。

1. 为自己和您的交易伙伴创建证书。如果您拥有可以使用的现有证书，则可跳过此部分。

   步骤 1：创建 AS2 证书 中介绍了此过程。

2. 创建使用 AS2 协议的 AWS Transfer Family 服务器。或者，您可以向服务器添加弹性 IP 地址，使其面向互联网。

   第 2 步：创建使用 AS2 协议的 Transfer Family 服务器 中介绍了此过程。

   > **Note**
   >
   > 您必须仅为入站传输创建 Transfer Family 服务器。如果您只执行出站传输，则不需要 Transfer Family 服务器。

3. 导入已在第 1 步中创建的证书。

   第 3 步：将证书作为 Transfer Family 证书资源导入 中介绍了此过程。

4. 要设置您的交易伙伴，请创建本地配置文件和合作伙伴配置文件。

   第 4 步：为您和您的交易伙伴创建配置文件 中介绍了此过程。

5. 在您和您的交易伙伴之间创建协议。

第 5 步：创建您与合作伙伴之间的协议 中介绍了此过程。

> **Note**
>
> 您必须仅为入站传输创建协议。如果您只执行出站传输，则无需协议。

6. 在您和您的交易伙伴之间创建连接器。

   第 6 步：创建您与合作伙伴之间的连接器 中介绍了此过程。

> **Note**
>
> 您必须仅为出站传输创建连接器。如果您仅执行入站传输，则不需要连接器。

7. 测试 AS2 文件交换。

   第 7 步：使用 Transfer Family 测试通过 AS2 交换文件 中介绍了此过程。

完成这些步骤后，您可以执行以下操作：

- 使用 Transfer Family `start-file-transfer` AWS Command Line Interface (AWS CLI) 命令将文件发送到支持 AS2 的远程伙伴服务器。
- 通过您的虚拟私有云（VPC）端点在端口 5080 上从启用 AS2 的远程合作伙伴服务器接收文件。

步骤 1：创建 AS2 证书

双方的 AS2 交换都需要 X.509 证书。您可以按喜欢的任何方式创建这些证书。本主题介绍如何通过命令行使用 OpenSSL 创建根证书，然后对从属证书进行签名。双方都必须生成自己的证书。

> **Note**
>
> AS2 证书的密钥长度必须至少为 2048 位，最多为 4096 位。

要与合作伙伴传输文件，请注意以下事项：

- 您可以将证书附加到配置文件。证书包含公钥或私钥。

- 您的交易伙伴将他们的公钥发送给您，而您则将您的公钥发送给他们。
- 您的交易伙伴使用您的公钥对消息进行加密，并使用其私钥对消息进行签名。相反，您可以使用合作伙伴的公钥对消息进行加密，然后使用您的私钥对消息进行签名。

> **Note**
>
> 如果您更喜欢使用 GUI 管理密钥，Portecle则是您可以使用的一个选项。

生成示例证书

> **Important**
>
> 不要将您的私钥发送给您的合作伙伴。在此示例中，您为一方生成一组自签名的公钥和私钥。如果您打算同时充当两个交易伙伴进行测试，则可以重复这些说明以生成两组密钥：每个交易伙伴一组。在这种情况下，您无需生成两个根证书颁发机构 (CA)。

1. 运行以下命令以生成带有 2048 位长度模数的 RSA 私有密钥。

```
/usr/bin/openssl genrsa -out root-ca-key.pem 2048
```

2. 运行以下命令以使用您的`root-ca-key.pem`文件创建自签名证书。

```
/usr/bin/openssl req \
-x509 -new -nodes -sha256 \
-days 1825 \
-subj "/C=US/ST=MA/L=Boston/O=TransferFamilyCustomer/OU=IT-dept/CN=ROOTCA" \
-key root-ca-key.pem \
-out root-ca.pem
```

`-subj` 参数由以下值组成。

	名称	描述
`C`	国家/地区代码	由两个字母组成的代码，代表您的组织所在的国家/地区。

	名称	描述
ST	州、地区或省	组织所在的州、地区或省。（在本例中，区域不是指您的 AWS 区域。）
L	所在地名称	组织所在的城市。
O	组织名称	您组织的法定全名，包括后缀，例如 LLC、Corp 等。
OU	组织部门名称	您组织中负责处理此证书的部门。
CN	公用名或完全限定域名 (FQDN)	在这种情况下，我们将创建一个根证书，因此值为ROOTCA。在这些示例中，我们使用CN来描述证书的用途。

3. 为您的本地配置文件创建签名密钥和加密密钥。

```
/usr/bin/openssl genrsa -out signing-key.pem 2048
/usr/bin/openssl genrsa -out encryption-key.pem 2048
```

> **Note**
>
> 某些启用 AS2 的服务器（例如 OpenAS2）要求您使用相同的证书进行签名和加密。在这种情况下，您可以为这两个目的导入相同的私钥和证书。为此，请运行以下命令而不是之前的两个命令：
>
> ```
> /usr/bin/openssl genrsa -out signing-and-encryption-key.pem 2048
> ```

4. 运行以下命令创建证书签名请求 (CSR)，供根密钥签名。

```
/usr/bin/openssl req -new -key signing-key.pem -subj \
"/C=US/ST=MA/L=Boston/O=TransferFamilyCustomer/OU=IT-dept/CN=Signer" -out signing-
key-csr.pem
```

```
/usr/bin/openssl req -new -key encryption-key.pem -subj \
"/C=US/ST=MA/L=Boston/O=TransferFamilyCustomer/OU=IT-dept/CN=Encrypter" -out
 encryption-key-csr.pem
```

5. 接下来，必须创建一个signing-cert.conf文件和一个encryption-cert.conf文件。

   - 使用文本编辑器创建包含以下内容的signing-cert.conf文件：

     ```
     authorityKeyIdentifier=keyid,issuer
     keyUsage = digitalSignature, nonRepudiation
     ```

   - 使用文本编辑器创建包含以下内容的encryption-cert.conf文件：

     ```
     authorityKeyIdentifier=keyid,issuer
     keyUsage = dataEncipherment
     ```

6. 最后，您可以通过运行以下命令来创建签名证书。

   ```
   /usr/bin/openssl x509 -req -sha256 -CAcreateserial -days 1825 -in signing-key-
   csr.pem -out signing-cert.pem -CA \
   root-ca.pem -CAkey root-ca-key.pem -extfile signing-cert.conf
   ```

   ```
   /usr/bin/openssl x509 -req -sha256 -CAcreateserial -days 1825 -in encryption-key-
   csr.pem -out encryption-cert.pem \
   -CA root-ca.pem -CAkey root-ca-key.pem -extfile encryption-cert.conf
   ```

第 2 步：创建使用 AS2 协议的 Transfer Family 服务器

此过程说明了如何使用 Transfer Family AWS CLI创建启用 AS2 的服务器。

> **Note**
>
> 许多示例步骤都使用从文件加载参数的命令。有关使用文件加载参数的更多详细信息，请参阅如何从文件加载参数。

如果要改用控制台，请参阅使用 Transfer Family 控制台创建 AS2 服务器。

与创建 SFTP 或 FTPS AWS Transfer Family 服务器的方式类似，您可以使用命令的`--protocols` AS2参数创建支持 AS2 的服务器。`create-server` AWS CLI 目前，Transfer Family 仅支持 VPC 端点类型和采用 AS2 协议的 Amazon S3 存储。

当您使用`create-server`命令为 Transfer Family 创建启用 AS2 的服务器时，系统会自动为您创建一个 VPC 端点。此端点公开 TCP 端口 5080，以便它可以接受 AS2 消息。

如果您想向互联网公开您的 VPC 端点，可以将弹性 IP 地址与您的 VPC 端点关联起来。

要使用这些说明，您需要以下内容：

- 您的 VPC 的 ID（例如 vpc-abcdef01）。
- 您的 VPC 子网的 ID（例如 sub net-abcdef01、01、subnet-021345ab）。subnet-subnet-abcdef
- 允许交易伙伴通过 TCP 端口 5080 传入流量的一个或多个安全组 ID（例如 sg-1234567890abcdef0 和 sg-abcdef01234567890）。
- （可选）您要与 VPC 端点关联的弹性 IP 地址。
- 如果您的交易伙伴未通过 VPN 连接到您的 VPC，则需要互联网网关。有关更多信息，请参阅《Amazon VPC 用户指南》中的使用互联网网关连接到互联网。

要创建启用 AS2 的服务器

1. 运行以下命令。将每个 *user input placeholder* 替换为您自己的信息。

```
aws transfer create-server --endpoint-type VPC \
--endpoint-details VpcId=vpc-abcdef01,SubnetIds=subnet-abcdef01,subnet-
abcdef01,subnet-
021345ab,SecurityGroupIds=sg-abcdef01234567890,sg-1234567890abcdef0 --protocols AS2
 \
--protocol-details As2Transports=HTTP
```

2. （可选）您可以将 VPC 端点设为公有。您只能通过`update-server`操作将弹性 IP 地址附加到 Transfer Family 服务器。以下命令停止服务器，使用弹性 IP 地址对其进行更新，然后重新启动服务器。

```
aws transfer stop-server --server-id your-server-id
```

```
aws transfer update-server --server-id your-server-id --endpoint-details \
```

```
AddressAllocationIds=eipalloc-abcdef01234567890,eipalloc-
1234567890abcdef0,eipalloc-abcd012345ccccccc
```

```
aws transfer start-server --server-id your-server-id
```

此start-server命令会自动为您创建 DNS 记录，其中包含您的服务器的公有 IP 地址。要让您的交易伙伴访问服务器，您需要向他们提供以下信息。在这种情况下，*your-region*指的是您的 AWS 区域。

s-*your-server-id*.server.transfer.*your-region*.amazonaws.com

您提供给交易伙伴的完整 URL 如下：

http://s-*your-server-id*.server.transfer.*your-region*.amazonaws.com:5080

3. 要测试您启用 AS2 的服务器是否可以访问，请使用以下命令。确保可以通过您的 VPC 端点的私有 DNS 地址或公有端点（如果您将弹性 IP 地址与端点相关联）访问您的服务器。

   如果您的服务器配置正确，则连接将成功。但是，您将收到 HTTP 状态码 400（错误请求）响应，因为您没有发送有效的 AS2 消息。

   - 对于公共端点（如果您在上一步中关联了弹性 IP 地址），请运行以下命令，替换您的服务器 ID 和区域。

     ```
     curl -vv -X POST http://s-your-server-id.transfer.your-region.amazonaws.com:5080
     ```

   - 如果您在 VPC 内进行连接，请运行以下命令查找 VPC 端点的私有 DNS 名称。

     ```
     aws transfer describe-server --server-id s-your-server-id
     ```

     此describe-server命令在VpcEndpointId参数中返回您的 VPC 端点 ID。使用此值运行以下命令。

     ```
     aws ec2 describe-vpc-endpoints --vpc-endpoint-ids vpce-your-vpc-endpoint-id
     ```

     此describe-vpc-endpoints命令返回一个包含多个DnsName参数的DNSEntries数组。在以下命令中使用区域 DNS 名称（不包括可用区的名称）。

```
curl -vv -X POST http://vpce-your-vpce.vpce-svc-your-vpce-svc.your-
region.vpce.amazonaws.com:5080
```

例如，以下命令显示了上一个命令中占位符的示例值。

```
curl -vv -X POST http://vpce-0123456789abcdefg-fghij123.vpce-
svc-11111aaaa2222bbbb.us-east-1.vpce.amazonaws.com:5080
```

4. （可选）配置日志记录角色。Transfer Family 以结构化 JSON 格式将发送和接收的消息的状态记录到亚马逊 CloudWatch 日志中。要让 Transfer Family 能够访问您账户中的 CloudWatch 日志，您必须在服务器上配置日志角色。

   创建信任`transfer.amazonaws.com`的 AWS Identity and Access Management (IAM) 角色并附加`AWSTransferLoggingAccess`托管策略。有关更多信息，请参阅 创建 IAM 角色和策略。请注意您刚创建的 IAM 角色的 Amazon 资源名称（ARN），然后通过运行以下`update-server`命令将其与服务器关联：

   ```
   aws transfer update-server --server-id your-server-id --logging-role
    arn:aws:iam::your-account-id:role/logging-role-name
   ```

   > **Note**
   >
   > 尽管日志记录角色是可选的，但我们强烈建议您对其进行设置，以便您可以查看消息的状态并对配置问题进行故障排除。

第 3 步：将证书作为 Transfer Family 证书资源导入

此过程介绍如何使用 AWS CLI导入证书。如果您想改用 Transfer Family 控制台，请参阅the section called “导入 AS2 证书”。

要导入您在第 1 步中创建的签名和加密证书，请运行以下`import-certificate`命令。如果您使用相同的证书进行加密和签名，请两次导入相同的证书（一次是`SIGNING`用法，另一次是`ENCRYPTION`用法）。

```
aws transfer import-certificate --usage SIGNING --certificate file://signing-cert.pem \
            --private-key file://signing-key.pem --certificate-chain file://root-ca.pem
```

此命令返回您的签名CertificateId。在下一节中，此证书 ID 被称为*my-signing-cert-id*。

```
aws transfer import-certificate --usage ENCRYPTION --certificate file://encryption-
cert.pem \
            --private-key file://encryption-key.pem --certificate-chain file://root-
ca.pem
```

此命令返回您的加密信息CertificateId。在下一节中，此证书 ID 被称为*my-encrypt-cert-id*。

接下来，通过运行以下命令导入合作伙伴的加密和签名证书。

```
aws transfer import-certificate --usage ENCRYPTION --certificate file://partner-
encryption-cert.pem \
--certificate-chain file://partner-root-ca.pem
```

此命令返回您的合作伙伴的加密信息CertificateId。在下一节中，此证书 ID 被称为*partner-encrypt-cert-id*。

```
aws transfer import-certificate --usage SIGNING --certificate file://partner-signing-
cert.pem \
--certificate-chain file://partner-root-ca.pem
```

此命令返回您的合作伙伴的签名CertificateId。在下一节中，此证书 ID 被称为*partner-signing-cert-id*。

第 4 步：为您和您的交易伙伴创建配置文件

此过程说明如何使用 AWS CLI创建 AS2 配置文件。如果您想改用 Transfer Family 控制台，请参阅the section called “创建 AS2 配置文件”。

通过运行以下命令来创建您的本地 AS2 配置文件。此命令引用包含您的公钥和私钥的证书。

```
aws transfer create-profile --as2-id MYCORP --profile-type LOCAL --certificate-ids \
my-signing-cert-id my-encrypt-cert-id
```

此命令会返回您的配置文件 ID。在下一节中，此 ID 被称为*my-profile-id*。

现在，通过运行以下命令来创建合作伙伴配置文件。此命令仅使用合作伙伴的公钥证书。要使用此命令，请将*user input placeholders*替换为您自己的信息；例如，您的合作伙伴的 AS2 名称和证书 ID。

```
aws transfer create-profile --as2-id PARTNER-COMPANY --profile-type PARTNER --
certificate-ids \
partner-signing-cert-id partner-encrypt-cert-id
```

此命令会返回您的合作伙伴的配置文件 ID。在下一节中，此 ID 被称为*partner-profile-id*。

> **Note**
>
> 在前面的命令中，将 *MYCORP* 替换为您的组织名称，将 *PARTNER-COMPANY* 替换为交易伙伴的组织名称。

第 5 步：创建您与合作伙伴之间的协议

此过程介绍如何使用 AWS CLI创建 AS2 协议。如果您想改用 Transfer Family 控制台，请参阅the section called “创建 AS2 协议”。

协议汇集了两个配置文件（本地和合作伙伴）、它们的证书以及允许双方之间入站 AS2 传输的服务器配置。您可以通过运行以下命令来列出您的项目。

```
aws transfer list-profiles --profile-type LOCAL
aws transfer list-profiles --profile-type PARTNER
aws transfer list-servers
```

此步骤需要一个 Amazon S3 存储桶和 IAM 角色，该角色具有该存储桶的读/写权限。创建此角色的说明与 Transfer Family SFTP、FTP 和 FTPS 协议的说明相同，可在创建 IAM 角色和策略中找到。

要创建协议，您需要以下项目：

- Amazon S3 存储桶名称（以及对象前缀，如果已指定）
- 具有存储桶访问权限的 IAM 角色的 ARN
- 您的 Transfer Family 服务器 ID
- 您的配置文件 ID 和合作伙伴的配置文件 ID

通过运行以下命令创建协议。

```
aws transfer create-agreement --description "ExampleAgreementName" --server-id your-
server-id \
```

```
--local-profile-id your-profile-id --partner-profile-id your-partner-profile-id --base-
directory /DOC-EXAMPLE-DESTINATION-BUCKET/AS2-inbox \
--access-role arn:aws:iam::111111111111:role/TransferAS2AccessRole
```

如果成功，此命令将返回协议的 ID。然后，您可以使用以下命令查看协议的详细信息。

```
aws transfer describe-agreement --agreement-id agreement-id --server-id your-server-id
```

第 6 步：创建您与合作伙伴之间的连接器

此过程介绍如何使用 AWS CLI创建 AS2 连接器。如果您想改用 Transfer Family 控制台，请参阅the section called “配置 AS2 连接器”。

您可以使用 `StartFileTransfer` API 操作通过连接器将存储在 Amazon S3 中的文件发送到交易伙伴的 AS2 端点。您可以通过运行以下命令找到之前创建的配置文件。

```
aws transfer list-profiles
```

创建连接器时，必须提供合作伙伴的 AS2 服务器 URL。将以下文本复制到名为 `testAS2Config.json` 的文件中。

```
{
"Compression": "ZLIB",
"EncryptionAlgorithm": "AES256_CBC",
"LocalProfileId": "your-profile-id",
"MdnResponse": "SYNC",
"MdnSigningAlgorithm": "DEFAULT",
"MessageSubject": "Your Message Subject",
"PartnerProfileId": "partner-profile-id",
"SigningAlgorithm": "SHA256"
}
```

> **Note**
>
> 对于`EncryptionAlgorithm`，除非必须支持需要该`DES_EDE3_CBC`算法的旧版客户端，否则不要指定算法，因为该算法是一种弱加密算法。

然后运行以下命令以创建连接器。

```
aws transfer create-connector --url "http://partner-as2-server-url" \
--access-role your-IAM-role-for-bucket-access \
--logging-role arn:aws:iam::your-account-id:role/service-role/AWSTransferLoggingAccess
 \
--as2-config file:///path/to/testAS2Config.json
```

第 7 步：使用 Transfer Family 测试通过 AS2 交换文件

从您的交易伙伴那里接收文件

如果您将公有弹性 IP 地址与 VPC 端点相关联，Transfer Family 会自动创建包含您的公有 IP 地址的 DNS 名称。子域名是您的 AWS Transfer Family 服务器 ID（格式为`s-1234567890abcdef0`）。采用以下格式向交易伙伴提供您的服务器 URL。

```
http://s-1234567890abcdef0.server.transfer.us-east-1.amazonaws.com:5080
```

如果您没有将公有弹性 IP 地址与 VPC 端点相关联，请查找 VPC 端点的主机名，该端点可以在端口 5080 上通过 HTTP POST 接受交易伙伴发来的 AS2 消息。要检索 VPC 端点详细信息，请使用以下命令。

```
aws transfer describe-server --server-id s-1234567890abcdef0
```

例如，假设前面的命令返回 VPC 端点 ID `vpce-1234abcd5678efghi`。然后，您可以使用以下命令检索 DNS 名称。

```
aws ec2 describe-vpc-endpoints --vpc-endpoint-ids vpce-1234abcd5678efghi
```

此命令返回运行以下命令所需的 VPC 端点的所有详细信息。

DNS 名称列在`DnsEntries`数组中。您的交易伙伴必须在您的 VPC 内才能访问您的 VPC 端点（例如通过 AWS PrivateLink 或 VPN）。采用以下格式向您的合作伙伴提供您的 VPC 端点 URL。

```
http://vpce-your-vpce-id.vpce-svc-your-vpce-svc-id.your-region.vpce.amazonaws.com:5080
```

例如，以下 URL 显示了前面命令中占位符的示例值。

```
http://vpce-0123456789abcdefg-fghij123.vpce-svc-11111aaaa2222bbbb.us-
east-1.vpce.amazonaws.com:5080
```

在此示例中，成功的传输存储在您在第 5 步：创建您与合作伙伴之间的协议中指定的base-directory参数中指定的位置。如果我们成功接收名为myfile1.txt和myfile2.txt的文件，则这些文件将存储为/*path-defined-in-the-agreement*/processed/*original_filename.messageId.original_extension*。在这里，文件存储为/*DOC-EXAMPLE-DESTINATION-BUCKET*/AS2-inbox/processed/myfile1.*messageId*.txt和/*DOC-EXAMPLE-DESTINATION-BUCKET*/AS2-inbox/processed/myfile2.*messageId*.txt。

如果您在创建 Transfer Family 服务器时配置了日志角色，则还可以查看 CloudWatch 日志以了解 AS2 消息的状态。

向您的交易伙伴发送文件

您可以使用 Transfer Family 通过引用连接器 ID 和文件路径发送 AS2 消息，如以下 `start-file-transfer` AWS Command Line Interface (AWS CLI) 命令所示：

```
aws transfer start-file-transfer --connector-id c-1234567890abcdef0 \
--send-file-paths "/DOC-EXAMPLE-SOURCE-BUCKET/myfile1.txt" "/DOC-EXAMPLE-SOURCE-BUCKET/myfile2.txt"
```

要获取连接器详细信息，请运行以下命令：

```
aws transfer list-connectors
```

该 `list-connectors` 命令会返回连接器的连接器 ID、URL 和 Amazon 资源名称（ARN）。

要返回特定连接器的属性，请使用要使用的 ID 运行以下命令：

```
aws transfer describe-connector --connector-id your-connector-id
```

`describe-connector`命令返回连接器的所有属性，包括其 URL、角色、配置文件、消息处置通知 (MDN)、标签和监控指标。

您可以通过查看 JSON 和 MDN 文件来确认合作伙伴已成功接收文件。这些文件是根据文件名和位置中描述的约定命名的。如果您在创建连接器时配置了日志记录角色，则还可以检查 CloudWatch 日志中是否有 AS2 消息的状态。

配置 SFTP、FTPS 或 FTP 服务器端点

本主题提供有关创建和使用一个或多个 SFTP、FTPS 和 FTP 协议的 AWS Transfer Family 服务器端点的详细信息。

主题

身份提供商选项

AWS Transfer Family 提供了几种对用户进行身份验证和管理的方法。下表比较了您可以与 Transfer Family 一起使用的可用身份提供商。

操作	AWS Transfer Family 服务托管	AWS Managed Microsoft AD	Amazon API Gateway	AWS Lambda
受支持的协议	SFTP	SFTP、FTPS、FTP	SFTP、FTPS、FTP	SFTP、FTPS、FTP
基于密钥的身份验证	支持	否	是	支持
密码验证	不支持	是	是	支持
AWS Identity and Access Management (IAM) 和 POSIX	支持	是	是	支持
逻辑主目录	支持	是	是	支持

操作	AWS Transfer Family 服务托管	AWS Managed Microsoft AD	Amazon API Gateway	AWS Lambda
参数化访问权限（基于用户名）	支持	是	是	支持
临时访问结构	支持	否	是	是
AWS WAF	否	否	是	不支持

注意:

- IAM 用于控制 Amazon S3 后备存储的访问权限，Amazon EFS 使用 POSIX。
- Ad hoc 是指在运行时发送用户配置文件的能力。例如，您可以通过将用户名作为变量传递来将用户置于他们的主目录中。
- 有关的详细信息 AWS WAF，请参阅 是一个 Web 应用程序防火墙。。
- 有一篇博客文章描述了使用与微软 Azure AD 集成的 Lambda 函数作为您的 Transfer Family 身份提供商。有关详细信息，请参阅使用 Azure 活动目录 AWS Transfer Family 进行身份验证和。 AWS Lambda
- 我们提供了多个 AWS CloudFormation 模板来帮助您快速部署使用自定义身份提供程序的 Transfer Family 服务器。有关更多信息，请参阅 Lambda 函数模板。

在以下步骤中，您可以创建启用 SFTP 的服务器、启用 FTPS 的服务器、启用 FTP 的服务器或启用 AS2 的服务器。

下一步

- 创建启用 SFTP 的服务器
- 创建启用 FTPS 的服务器
- 创建启用 FTP 的服务器
- 配置 AS2

AWS Transfer Family 端点类型矩阵

创建 Transfer Family 服务器时，您需要选择要使用的端点类型。下表介绍了每种端点类型的特性。

端点类型矩阵

特征	公开	VPC - 互联网	VPC - 内部	VPC_Endpoint（已弃用）
受支持的协议	SFTP	SFTP, FTPS, AS2	SFTP, FTP, FTPS, AS2	SFTP
访问	来自互联网。此端点类型不需要在您的 VPC 中进行任何特殊配置。	通过互联网以及在 VPC 和 VPC 连接的环境中，例如本地数据中心或 VPN。 AWS Direct Connect	在 VPC 和与 VPC 连接的环境中，例如本地数据中心或 VPN。 AWS Direct Connect	在 VPC 和与 VPC 连接的环境中，例如本地数据中心或 VPN。 AWS Direct Connect
静态 IP 地址	您无法附加静态 IP 地址。 AWS 提供随时可能更改的 IP 地址。	您可以将弹性 IP 地址附加到端点。这些地址可以是 AWS自有的 IP 地址或您自己的 IP 地址（自带 IP 地址）。附加到端点的弹性 IP 地址不会更改。 附加到服务器的私有 IP 地址也不会更改。	附加到端点的私有 IP 地址不会更改。	附加到端点的私有 IP 地址不会更改。
源 IP 允许列表	此端点类型不支持按源 IP 地址列出的允许列表。 端点可公开访问并侦听端口 22 上的流量。	要允许通过源 IP 地址进行访问，您可以使用附加到服务器端点的安全组以及附加到端点所在子网的网络 ACL。	要允许通过源 IP 地址进行访问，您可以使用附加到服务器端点的安全组以及附加到端点所在子网的网络访问控制列表。	要允许通过源 IP 地址进行访问，您可以使用附加到服务器端点的安全组以及附加到端点所在子网的网络 ACL。

特征	公开	VPC - 互联网	VPC - 内部	VPC_Endpoint（已弃用）
	Note 对于 VPC 托管的端点，SFTP Transfer Family 服务器可以通过端口 22（默认）或端口 2222 运行。			
客户端防火墙允许列表	您必须允许服务器的 DNS 名称。 由于 IP 地址可能会发生更改，因此请避免将 IP 地址用于您的客户端防火墙允许列表。	您可以允许服务器的 DNS 名称或附加到服务器的弹性 IP 地址。	您可以允许端点的私有 IP 地址或 DNS 名称。	您可以允许端点的私有 IP 地址或 DNS 名称。

Note

`VPC_ENDPOINT`端点类型现已弃用，无法用于创建新的服务器。不要使用`EndpointType=VPC_ENDPOINT`，而是使用新的 VPC 端点类型 (`EndpointType=VPC`)，

您可以将其用作内部端点或面向互联网，如上表所述。有关更多信息，请参阅 停止使用 VPC_ENDPOINT。

考虑以下选项来提高 AWS Transfer Family 服务器的安全状况：

- 使用具有内部访问权限的 VPC 终端节点，这样只有您的 VPC 或 VPC 连接的环境（例如本地数据中心或 VPN）中的客户端才能访问服务器。 AWS Direct Connect
- 要允许客户端通过互联网访问端点并保护您的服务器，请使用具有面向互联网访问权限的 VPC 端点。然后，修改 VPC 的安全组，使其仅允许来自托管用户客户端的某些 IP 地址的流量。
- 如果您需要基于密码的身份验证，并且在服务器上使用自定义身份提供商，则最佳做法是，您的密码策略可以防止用户创建弱密码并限制失败的登录尝试次数。
- AWS Transfer Family 是一项托管服务，因此它不提供 shell 访问权限。您无法直接访问底层 SFTP 服务器以在 Transfer Family 服务器上运行 OS 本机命令。
- 在具有内部访问权限的 VPC 端点前使用网络负载均衡器。将负载均衡器上的侦听器端口从端口 22 更改为其他端口。这可以降低但不能消除端口扫描器和机器人探测服务器的风险，因为端口 22 最常用于扫描。有关详细信息，请参阅博客文章网络负载均衡器现在支持安全组。

> **Note**
>
> 如果您使用 Network Load Balancer，则 AWS Transfer Family CloudWatch 日志会显示 NLB 的 IP 地址，而不是实际的客户端 IP 地址。

配置 SFTP、FTPS 或 FTP 服务器端点

您可以使用该 AWS Transfer Family 服务创建文件传输服务器。以下文件传输协议可用：

- Secure Shell (SSH) 文件传输协议 (SFTP) — 通过 SSH 的文件传输 有关更多信息，请参阅 the section called “创建启用 SFTP 的服务器”。

> **Note**
>
> 我们提供了创建 SFTP Transfer Family 服务器的 AWS CDK 示例。该示例使用 TypeScript，可 GitHub 在此处找到。

- 文件传输协议安全 (FTPS) _— 使用 TLS 加密的文件传输功能 有关更多信息，请参阅 the section called “创建启用 FTPS 的服务器”。
- 文件传输协议 (FTP) — 未加密的文件传输功能 有关更多信息，请参阅 the section called “创建启用 FTP 的服务器”。
- 适用性声明 2 (AS2) — 用于传输结构化 business-to-business 数据的文件传输。有关更多信息，请参阅 the section called “配置 AS2”。对于 AS2，您可以快速创建 AWS CloudFormation 堆栈以进行演示。有关此过程的说明，请参阅使用模板创建演示 Transfer Family AS2 堆栈。

您可以创建具有多个协议的服务器。

> **Note**
>
> 如果您为同一个服务器端点启用了多个协议，并且想要通过多个协议使用相同的用户名提供访问权限，则只要在身份提供商中设置了该协议的特定凭据，就可以这样做。对于 FTP，建议保留与 SFTP 和 FTPS 不同的凭证。这是因为，与 SFTP 和 FTPS 不同，FTP 以明文形式传输凭证。通过将 FTP 凭证与 SFTP 或 FTPS 隔离开来，如果共享或公开 FTP 凭证，则使用 SFTP 或 FTPS 的工作负载会保持安全。

创建服务器时，您可以选择特定的服务器 AWS 区域 来执行分配给该服务器的用户的文件操作请求。除了为服务器分配一个或多个协议外，您还可以分配以下身份提供商类型之一：

- 使用 SSH 密钥托管的服务。有关更多信息，请参阅 与服务托管用户合作。
- AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD)。 此方法允许你整合 Microsoft Active Directory 群组以提供对 Transfer Family 服务器的访问权限。有关更多信息，请参阅 使用 Di AWS rectory Service 身份提供商。
- 一种自定义方法。自定义身份提供商方法使用 AWS Lambda 或 Amazon API Gateway，使您能够集成目录用于身份验证和授权用户。服务自动分配一个标识符，唯一标识您的服务器。有关更多信息，请参阅 使用自定义身份提供程序。Transfer Family 提供了 AWS CloudFormation 模板，您可以使用这些模板来快速部署使用自定义身份提供商的服务器。
  - 用于身份验证的 Lambda 函数描述了使用 Lambda 函数进行身份验证的 CloudFormation 模板。
  - 使用 API Gateway 方法进行身份验证描述了使用 Amazon API Gateway 方法进行身份验证的 CloudFormation模板。

您还可以使用默认服务器端点为服务器分配端点类型（可公开访问或 VPC 托管）和主机名，或者使用 Amazon Route 53 服务或使用您选择的域名系统 (DNS) 服务为服务器分配自定义主机名。服务器主机名在创建时 AWS 区域 必须是唯一的。

此外，您可以分配 Amazon CloudWatch CloudWatch 日志角色将事件推送到您的日志，选择包含可供服务器使用的加密算法的安全策略，并以键值对的标签形式向服务器添加元数据。

> **Important**
>
> 实例化的服务器和数据传输会产生费用。有关定价以及用于估算使用 AWS Pricing Calculator Transfer Family 的成本的信息，请参阅AWS Transfer Family 定价。

创建启用 SFTP 的服务器

Secure Shell (SSH) 文件传输协议 (SFTP) 是一种用于通过互联网安全传输数据的网络协议。该协议支持 SSH 的完整安全和身份验证功能。它被广泛应用于金融服务、医疗保健、零售和广告等各行各业的业务合作伙伴之间交换数据，包括敏感信息。

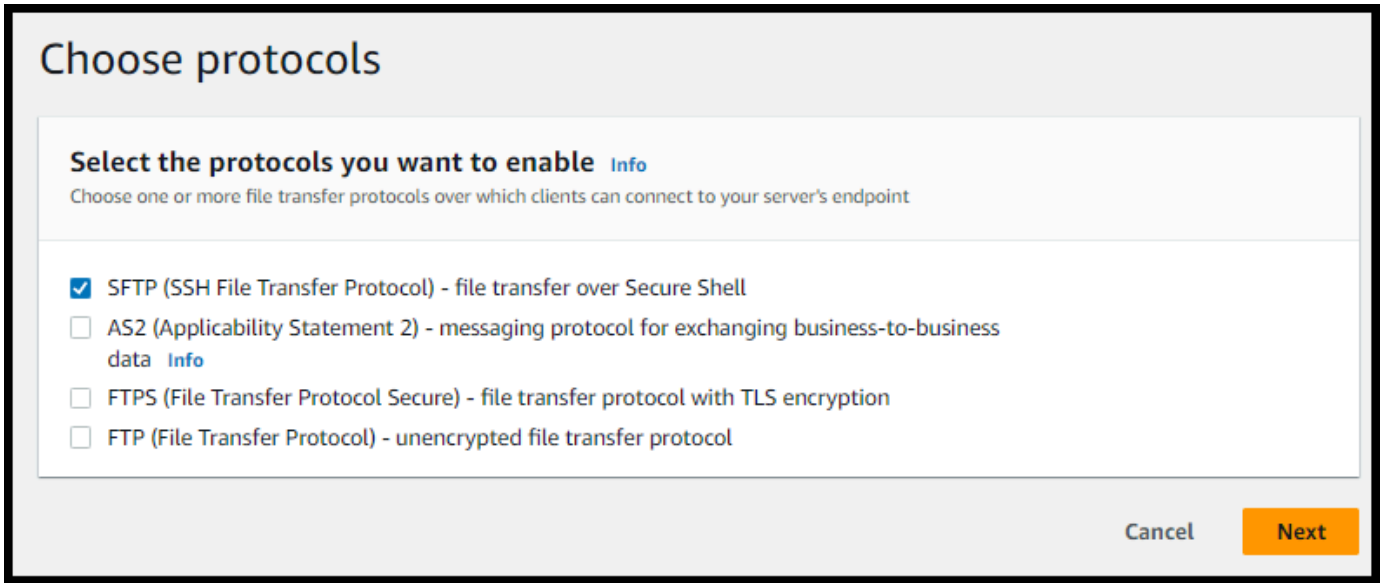
> **Note**
>
> Transfer Family 的 SFTP 服务器通过端口 22 运行。对于 VPC 托管的端点，SFTP Transfer Family 服务器也可以通过端口 2222 运行。有关更多信息，请参阅 在虚拟私有云中创建服务器。

另请参阅

- 我们提供了创建 SFTP Transfer Family 服务器的 AWS CDK 示例。该示例使用 TypeScript，可 GitHub 在此处找到。
- 有关如何在 VPC 内部署 Transfer Family 服务器的演练，请参阅使用 IP 允许列表保护您的 AWS Transfer Family 服务器。

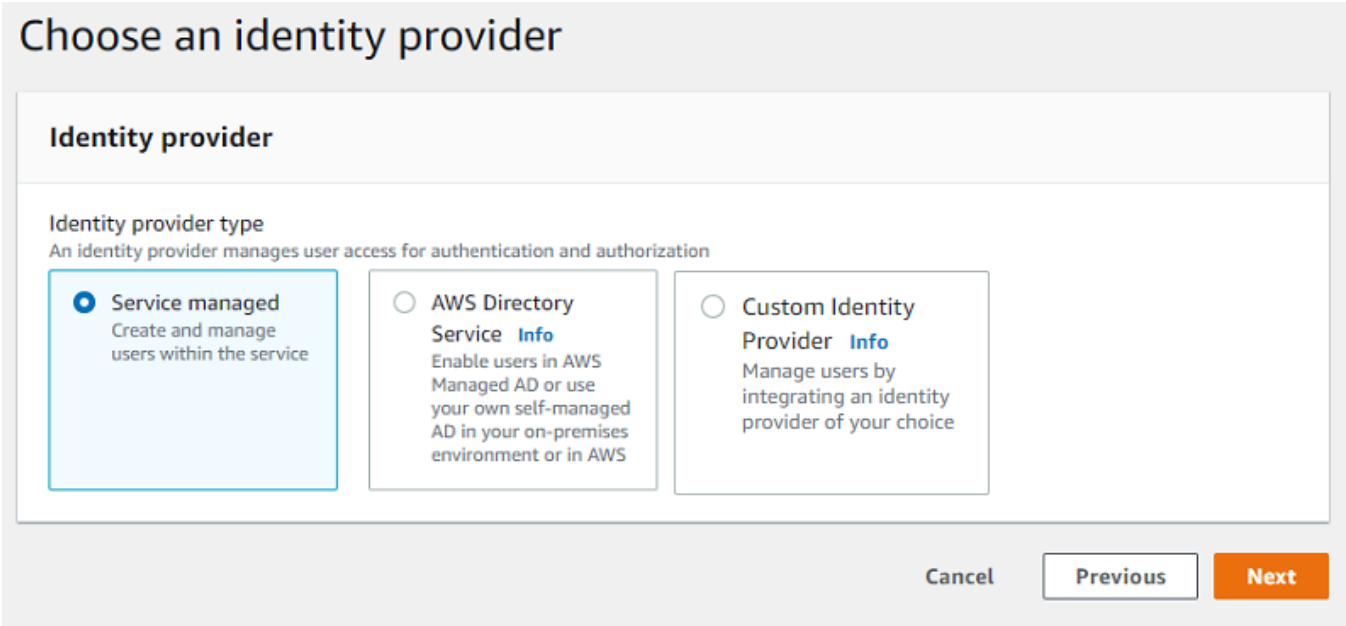
创建启用 SFTP 的服务器

1. 打开 AWS Transfer Family 控制台 https://console.aws.amazon.com/transfer/ 并从导航窗格中选择 “服务器”，然后选择 “创建服务器”。
2. 在选择协议中，选择 SFTP，然后选择下一步。



3. 在选择身份提供商中，选择要用于管理用户访问权限的身份提供商。您有以下选项：

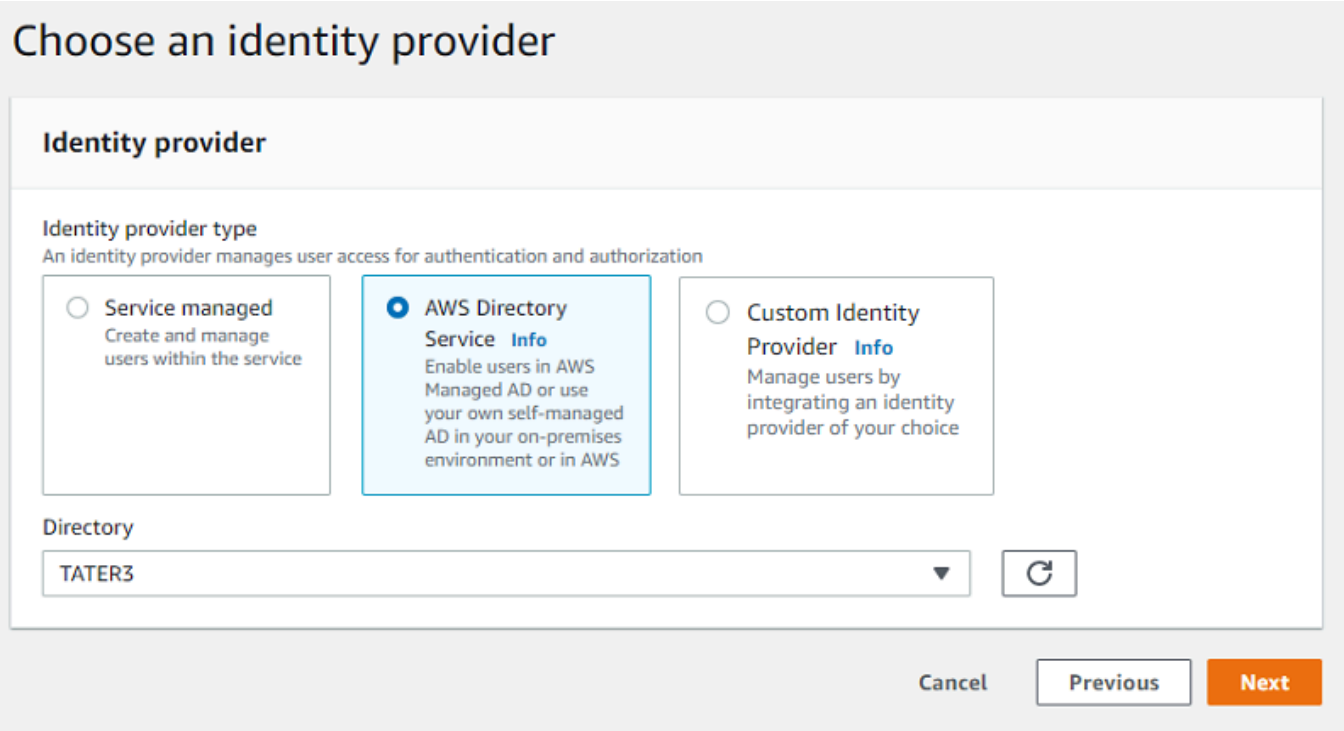
   - 服务托管-您将用户身份和密钥存储在中 AWS Transfer Family。



   - AWS Directory Service for Microsoft Active Directory— 您提供用于访问终端节点的 AWS Directory Service 目录。这样，您就可以使用存储在 Activity Directory 中的凭证对用户进行身份验证。要了解有关与 AWS Managed Microsoft AD 身份提供商合作的更多信息，请参阅使用 Di AWS rectory Service 身份提供商。

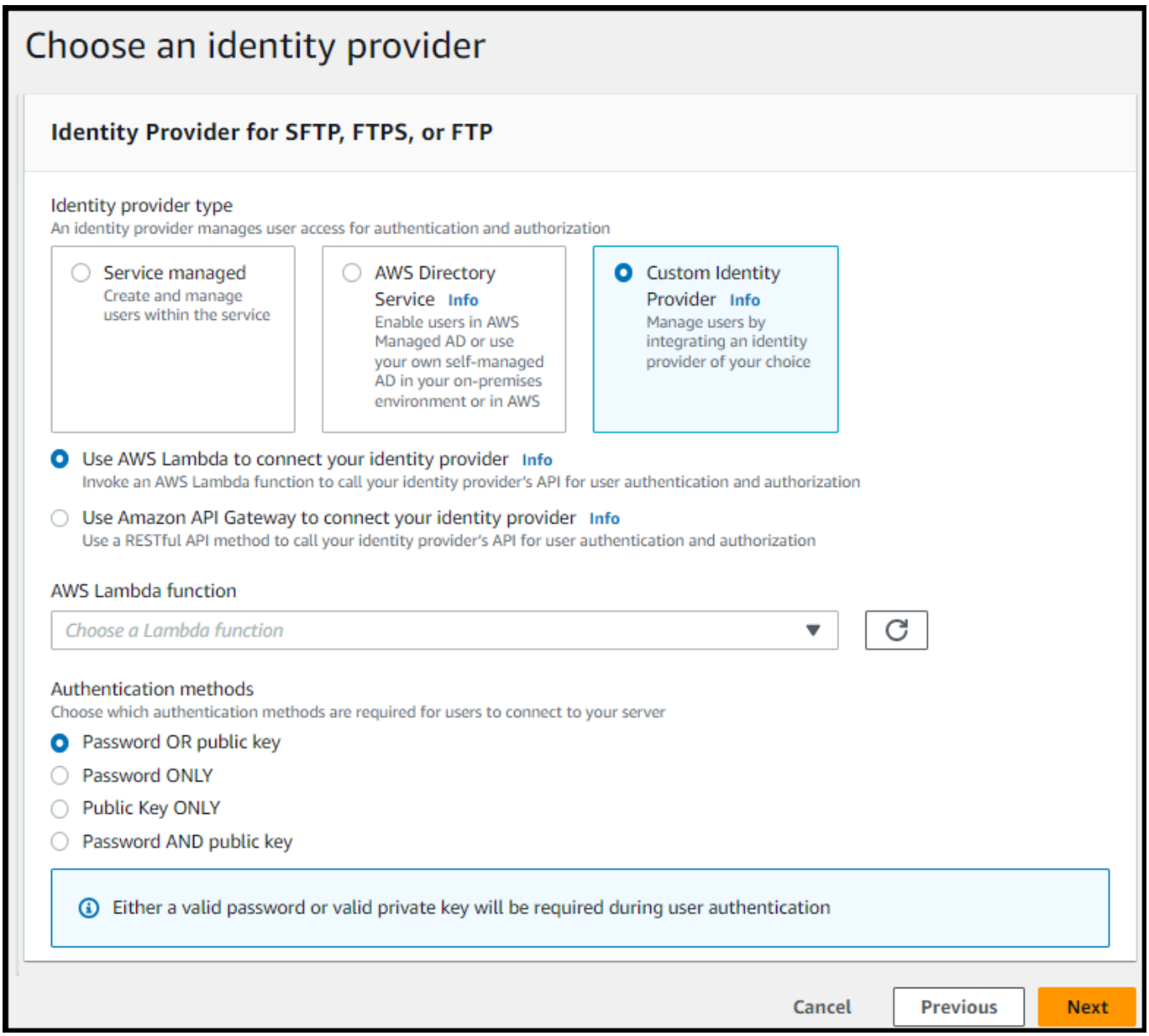
> **Note**
>
> - 不支持跨账户目录和共享目录。 AWS Managed Microsoft AD

- 要设置以 Directory Service 作为身份提供者的服务器，您需要添加一些 AWS Directory Service 权限。有关更多信息，请参阅 开始使用之前 AWS Directory Service for Microsoft Active Directory。



- 自定义身份提供商 — 请选择以下任一选项：
  - AWS Lambda 用于连接您的身份提供商-您可以使用由 Lambda 函数支持的现有身份提供商。您提供 Lambda 函数名称。有关更多信息，请参阅 AWS Lambda 用于整合您的身份提供商。
  - 使用 Amazon API Gateway 连接您的身份提供商 — 您可以创建由 Lambda 函数支持的 API 网关方法以用作身份提供商。您提供一个 Amazon API Gateway URL 和一个调用角色。有关更多信息，请参阅 使用 Amazon API Gateway 整合您的身份提供程序。

  对于任一选项，您还可以指定如何进行身份验证。
  - 密码或密钥-用户可以使用其密码或密钥进行身份验证。这是默认值。
  - 仅限密码-用户必须提供密码才能连接。
  - 仅限密钥 — 用户必须提供私钥才能连接。
  - 密码和密钥 — 用户必须同时提供私钥和密码才能连接。服务器首先检查密钥，如果密钥有效，系统会提示输入密码。如果提供的私有密钥与存储的公有密钥不匹配，则身份验证失败。



4. 选择下一步。
5. 在选择端点中，执行以下操作：

   a. 对于端点类型，选择可公开访问的端点类型。有关 VPC 托管的端点，请参阅[在虚拟私有云中创建服务器](#)。

   b. （可选）对于自定义主机名，选择无。

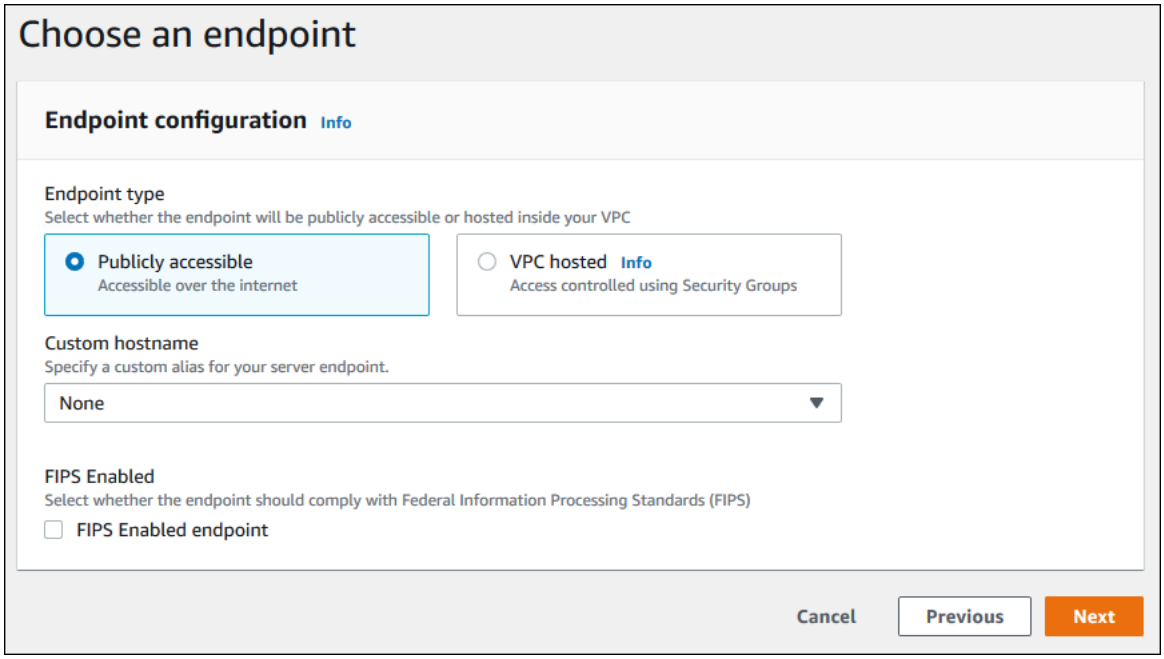
      您将获得由提供的服务器主机名 AWS Transfer Family。服务器主机名使用格式 *serverId*.server.transfer.*regionId*.amazonaws.com。

对于自定义主机名，您可以为服务器端点指定自定义别名。要了解有关使用自定义主机名的更多信息，请参阅使用自定义主机名。

c. （可选）对于启用 FIPS，请选中启用 FIPS 端点复选框以确保端点符合联邦信息处理标准 (FIPS)。

> **Note**
>
> 启用 FIPS 的端点仅在北美 AWS 地区可用。有关可用区域，请参阅AWS 一般参考中的AWS Transfer Family 端点和限额。有关 FIPS 的更多信息，请参阅联邦信息处理标准 (FIPS) 140-2。

d. 选择下一步。



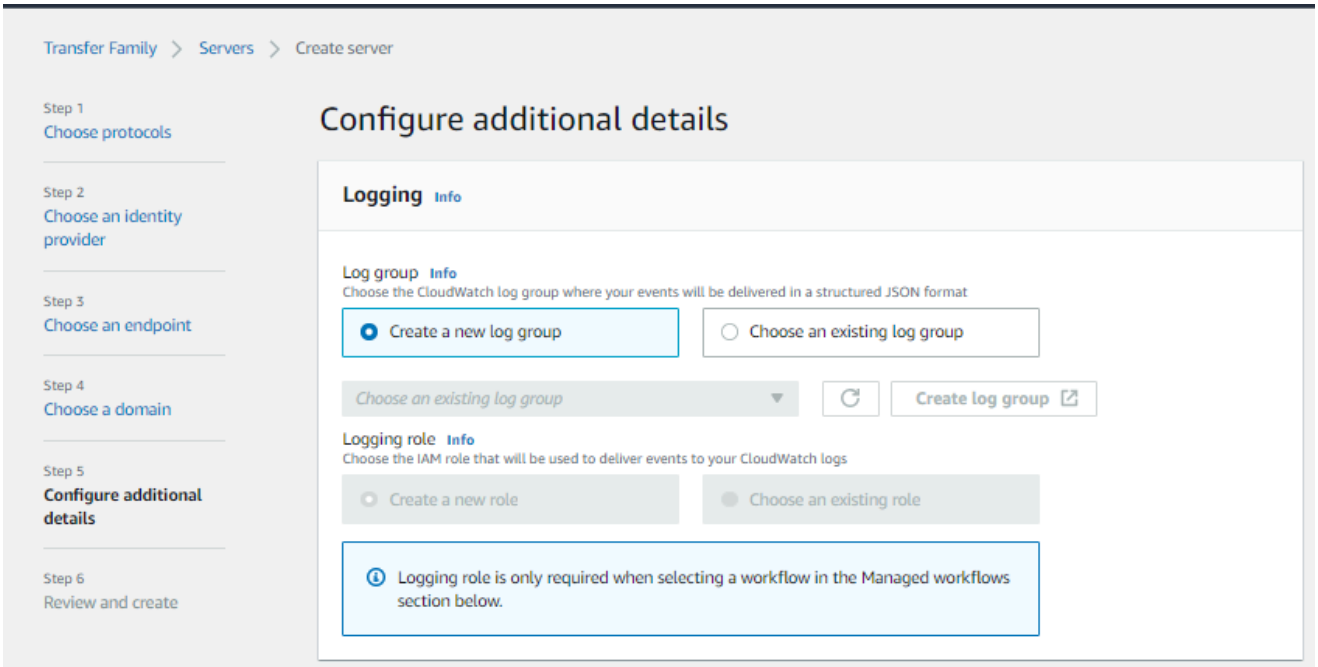
6. 在“选择域”页面上，选择要用于通过所选协议 AWS 存储和访问数据的存储服务：

   - 选择 Amazon S3，通过所选协议将您的文件作为对象存储和访问。
   - 选择 Amazon EFS，通过所选协议在 Amazon EFS 文件系统中存储和访问您的文件。

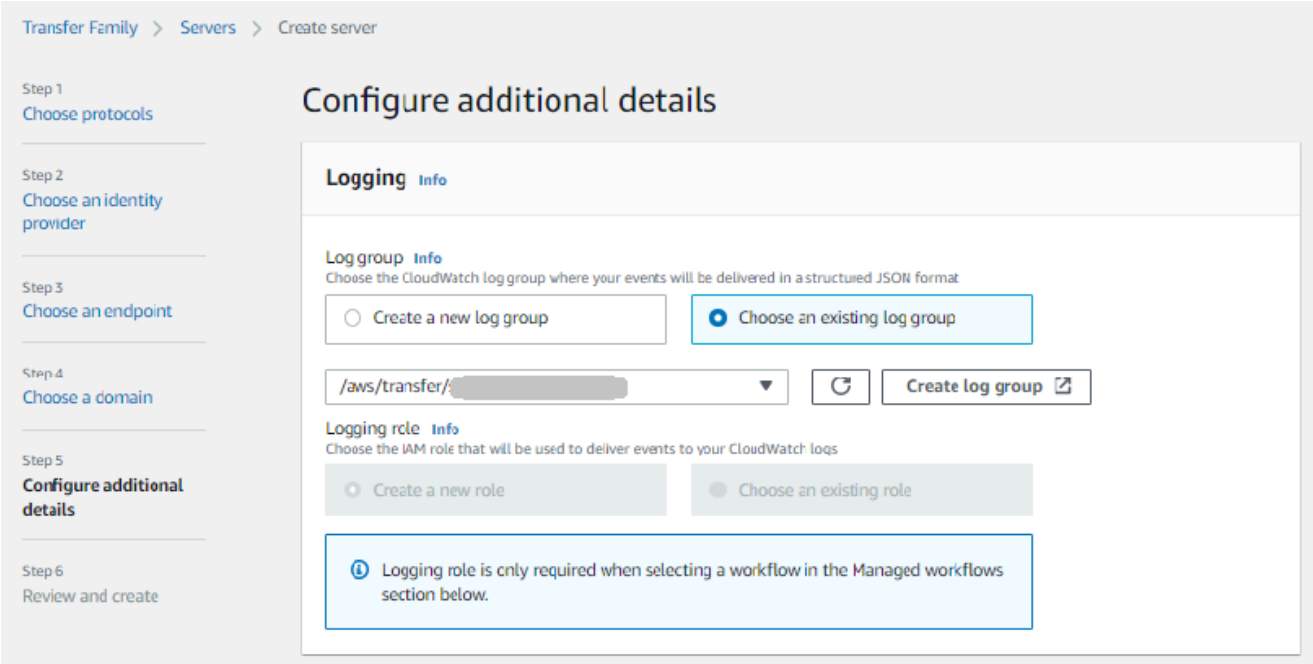
   选择下一步。

7. 在配置其他详细信息中，执行以下操作：

a. 对于日志记录，指定现有日志组或创建新日志组（默认选项）。

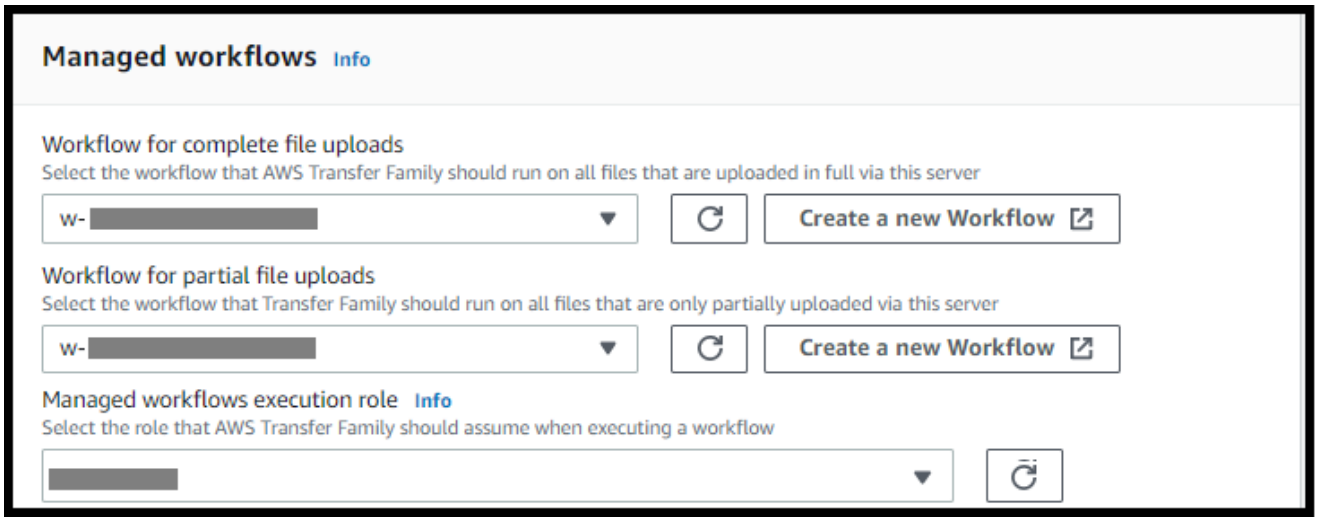


如果您选择现有日志组，则必须选择与您的日志组关联的日志组 AWS 账户。



如果选择 “创建日志组”，则 CloudWatch 控制台 (https://console.aws.amazon.com/cloudwatch/) 将打开 “创建日志组” 页面。有关详细信息，请参阅在 Log CloudWatch s 中创建日志组。

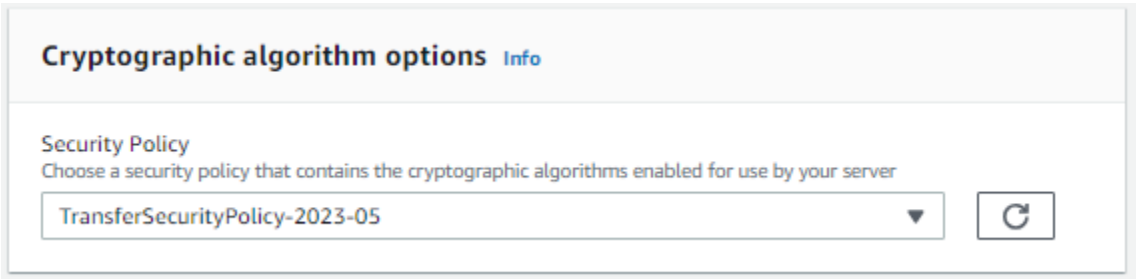
b. （可选）对于托管工作流程，请选择 Transfer Family 在执行工作流程时应承担的工作流程 ID（和相应的角色）。您可以选择一个工作流程在完成上传后执行，选择另一个工作流程在部分上传时执行。要了解有关使用托管工作流程处理文件的更多信息，请参阅AWS Transfer Family 托管工作流程。



c. 对于加密算法选项，请选择包含允许服务器使用的加密算法的安全策略。

> **Note**
>
> 默认情况下：
>
> - 如果未选择启用 FIPS 的端点，则`TransferSecurityPolicy-2020-06`安全策略将附加至您的服务器。
> - 如果已选择启用 FIPS 的端点，则`TransferSecurityPolicy-FIPS-2020-06`安全策略将附加至您的服务器。

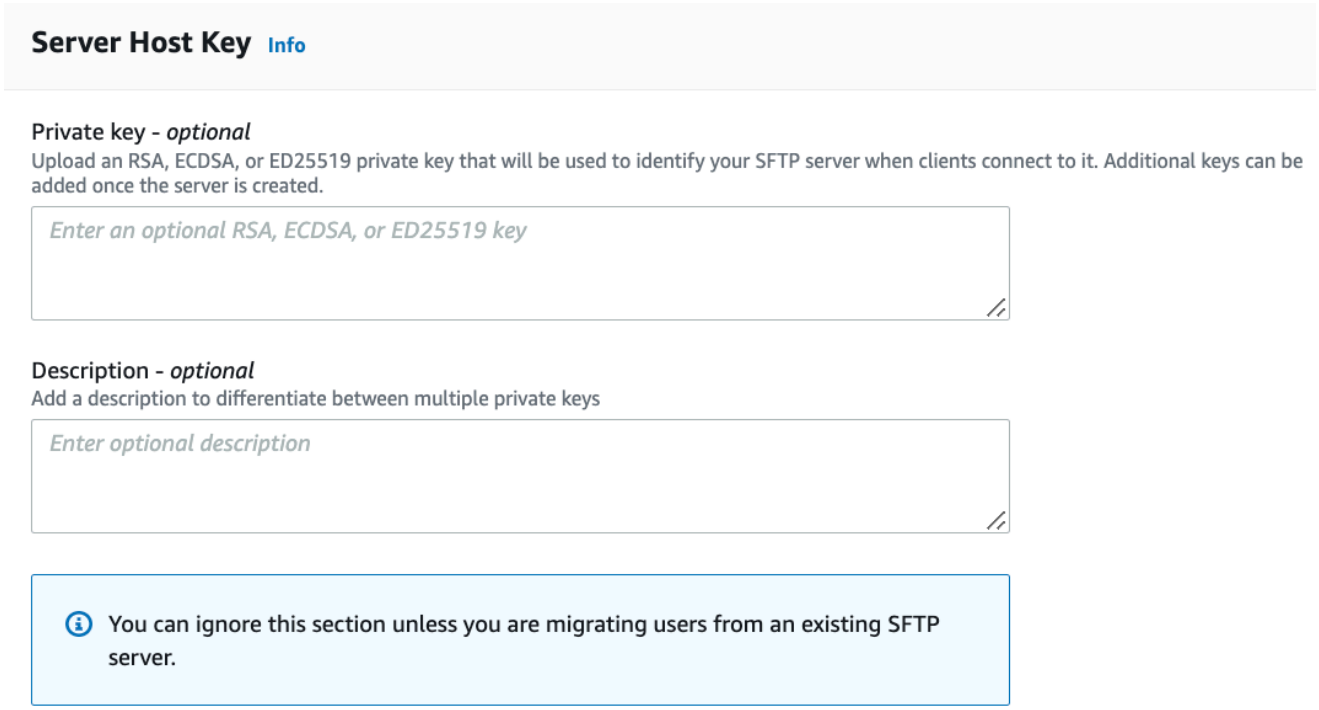
有关安全策略的更多信息，请参阅AWS Transfer Family 服务器的安全策略。



d. （可选）对于 Server Host Key，输入 RSA、ED25519 或 ECDSA 私有密钥，该私有密钥将用于在客户端通过 SFTP 连接到服务器时标识服务器。您还可以添加描述以区分多个主机密钥。

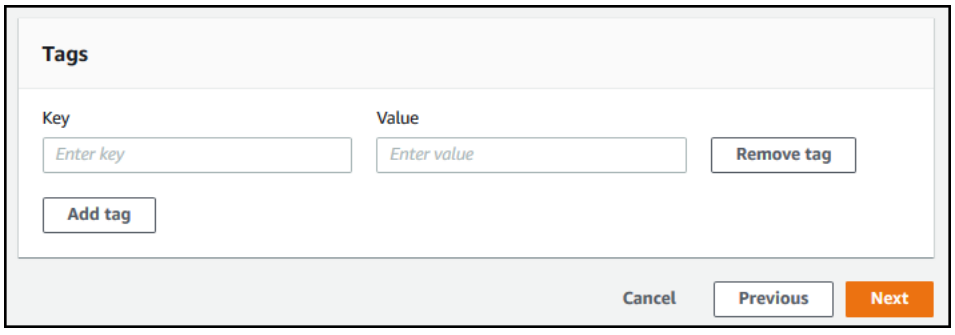
创建服务器后，您可以添加其他主机密钥。如果您想轮换密钥或想要使用不同类型的密钥（例如 RSA 密钥和 ECDSA 密钥），则拥有多个主机密钥非常有用。

> **Note**
>
> 服务器主机密钥部分仅用于从启用 SFTP 的现有服务器迁移用户。

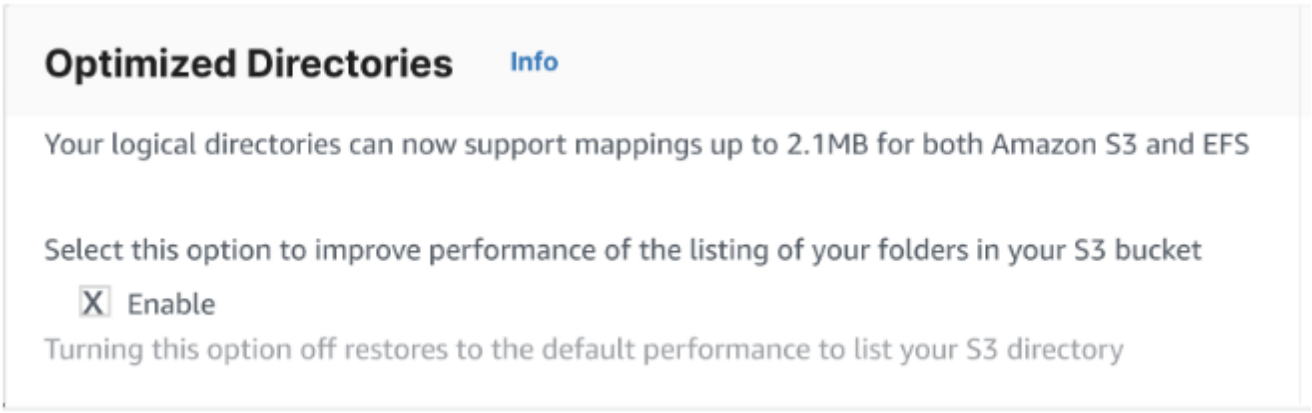


e. （可选）对于标签，在密钥和值中，输入一个或多个标签作为键值对，然后选择添加标签。

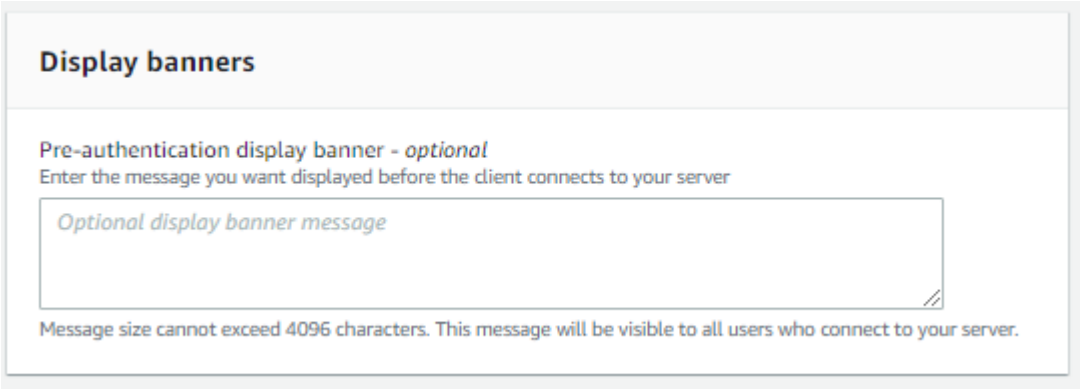
f. 选择下一步。



g. 您可以优化 Amazon S3 目录的性能。例如，假设您进入主目录，并且有 10,000 个子目录。换句话说，您的 S3 存储桶有 10,000 个文件夹。在这种情况下，如果您运行 `ls` (list) 命令，则列表操作需要六到八分钟。但是，如果您优化目录，则此操作只需要几秒钟。

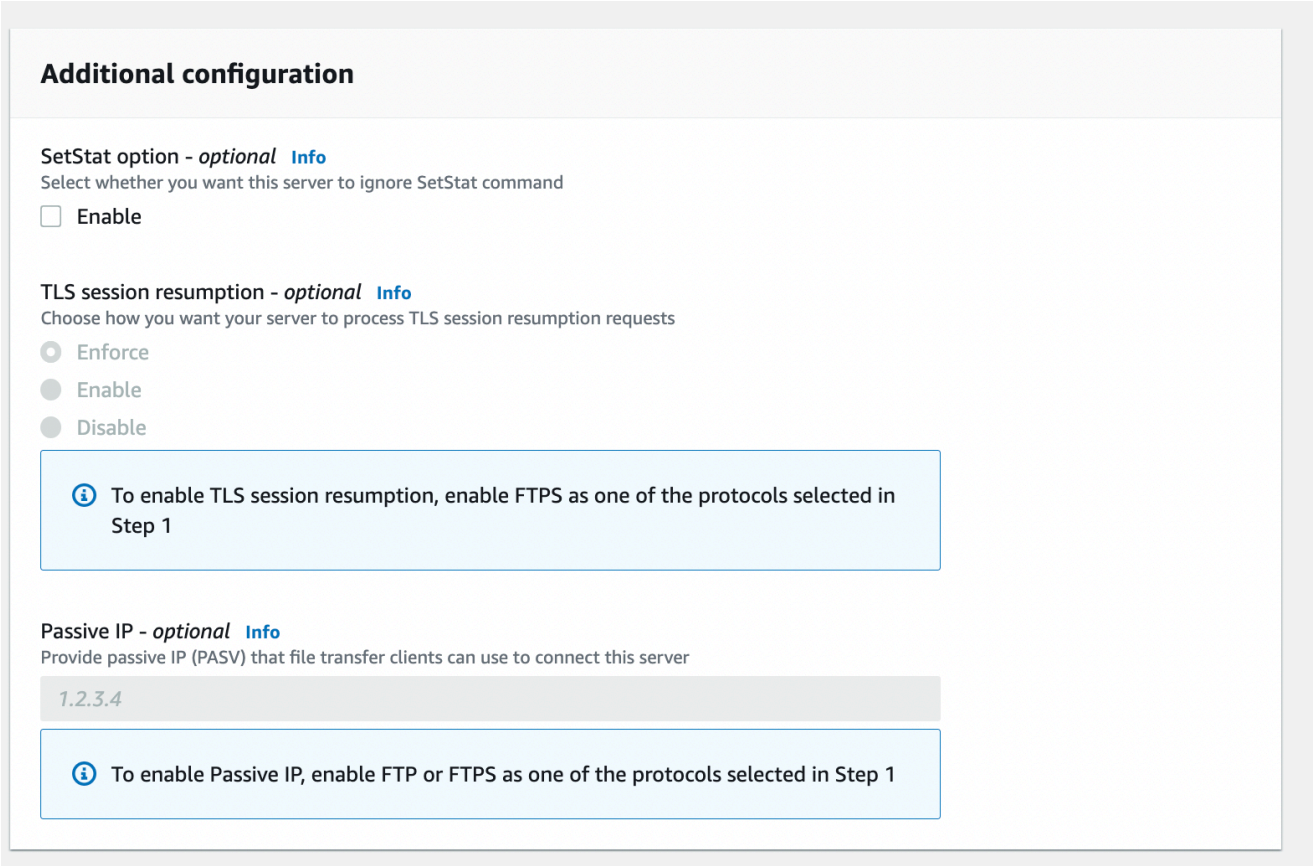


h. （可选）配置 AWS Transfer Family 服务器以向最终用户显示自定义消息，例如组织政策或条款和条件。对于显示横幅，在预身份验证显示横幅文本框中，输入要在用户进行身份验证之前向其显示的短信。



i. （可选）您可以配置以下其他选项。

- SetStat 选项：启用此选项可忽略客户端尝试对您上传到 Amazon S3 存储桶的文件使用`SETSTAT`时生成的错误。有关更多详细信息，请参阅中的`SetStatOption`文档ProtocolDetails。
- TLS 会话恢复：仅当您启用 FTPS 作为该服务器的协议之一时，此选项才可用。
- 被动 IP：仅当您启用 FTPS 或 FTP 作为该服务器的协议之一时，此选项才可用。



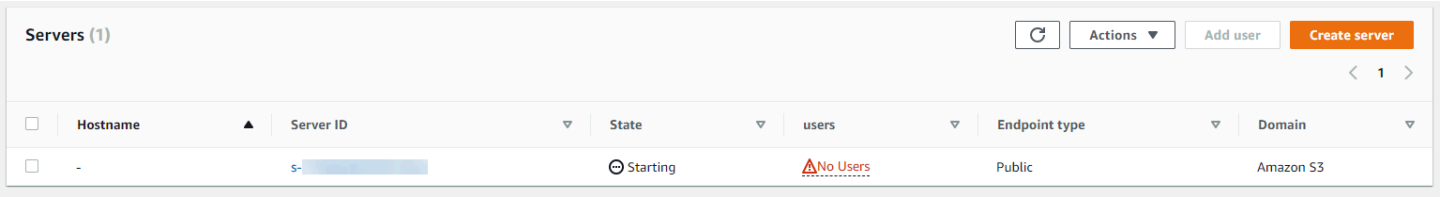
8. 在审核和创建页面上，审核您的选择。

   - 如果要编辑其中任何一个，请选择该步骤旁边的编辑。

     > **Note**
     >
     > 在选择编辑的步骤之后，您必须审核每个步骤。

   - 如果没有任何更改，请选择创建服务器来创建您的服务器。您将转至如下所示的 Servers (服务器) 页面，其中列出了您的新服务器。

您的新服务器状态更改为在线可能需要几分钟时间。到时候，您的服务器可以执行用户的文件操作。



创建启用 FTPS 的服务器

安全文件传输协议 (FTPS) 是 FTP 的扩展。它使用传输层安全性协议 (TLS)/安全套接字层 (SSL) 加密协议对流量进行加密。FTPS 允许同时或独立地对控制和数据通道连接进行加密。

创建启用 FTPS 的服务器

1. 打开 AWS Transfer Family 控制台 https://console.aws.amazon.com/transfer/ 并从导航窗格中选择 “服务器”，然后选择 “创建服务器”。
2. 在选择协议中，选择 FTPS。

   对于服务器证书，请选择存储在 AWS Certificate Manager (ACM) 中的证书，该证书将用于在客户端通过 FTPS 连接到服务器时标识服务器，然后选择下一步。

   要请求新的公有证书，请参阅AWS Certificate Manager 用户指南中的请求公有证书。

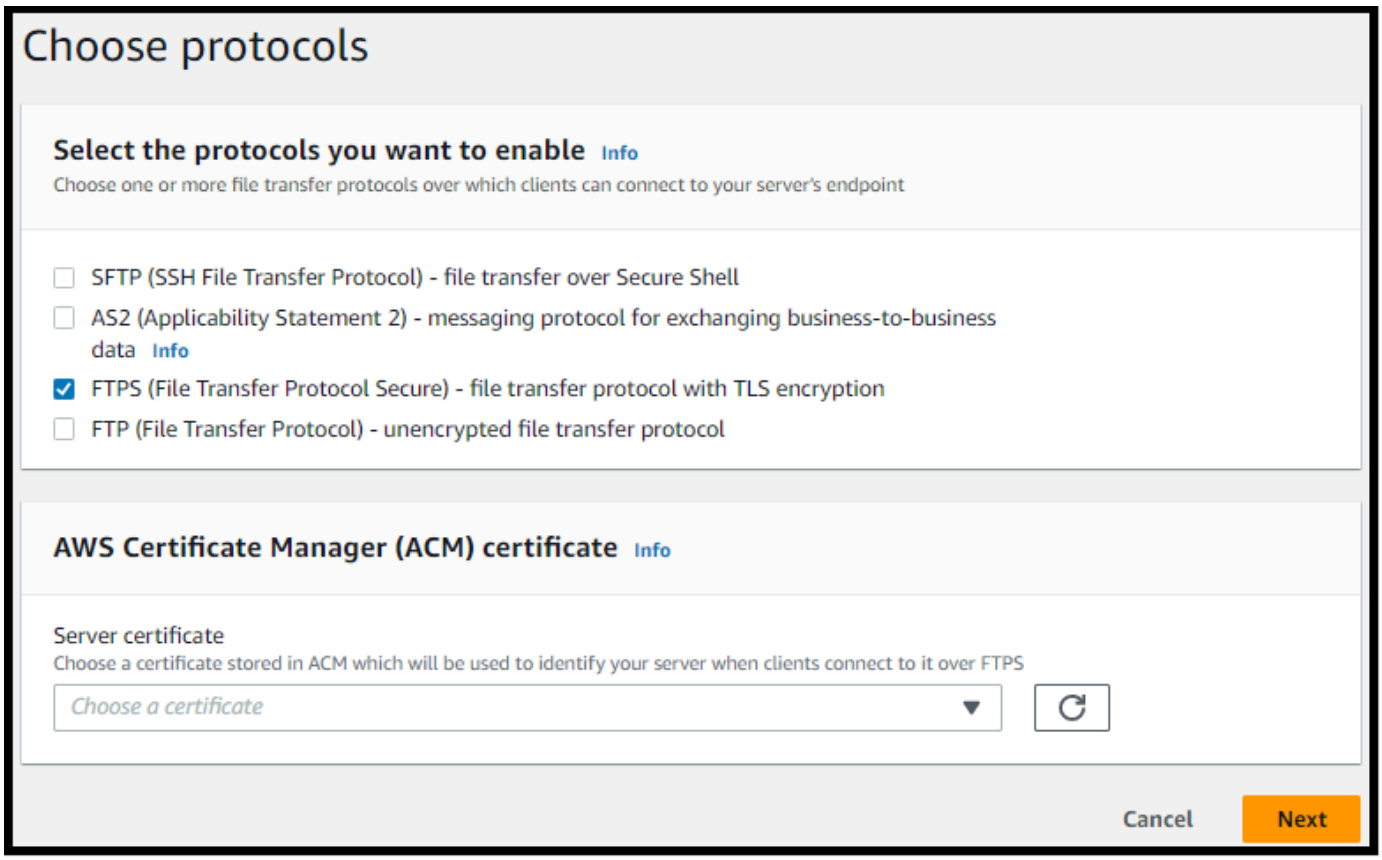
   要将现有证书导入到 ACM 中，请参阅AWS Certificate Manager 用户指南中的将证书导入到 ACM。

   要请求私有证书以通过私有 IP 地址使用 FTPS，请参阅AWS Certificate Manager 用户指南中的请求私有证书。

   支持具有以下加密算法和密钥大小的证书：

   - 2048 位 RSA (RSA_2048)
   - 4096 位 RSA (RSA_4096)
   - Elliptic Prime Curve 256 位 (EC_prime256v1)
   - Elliptic Prime Curve 384 位 (EC_secp384r1)
   - Elliptic Prime Curve 521 位 (EC_secp521r1)

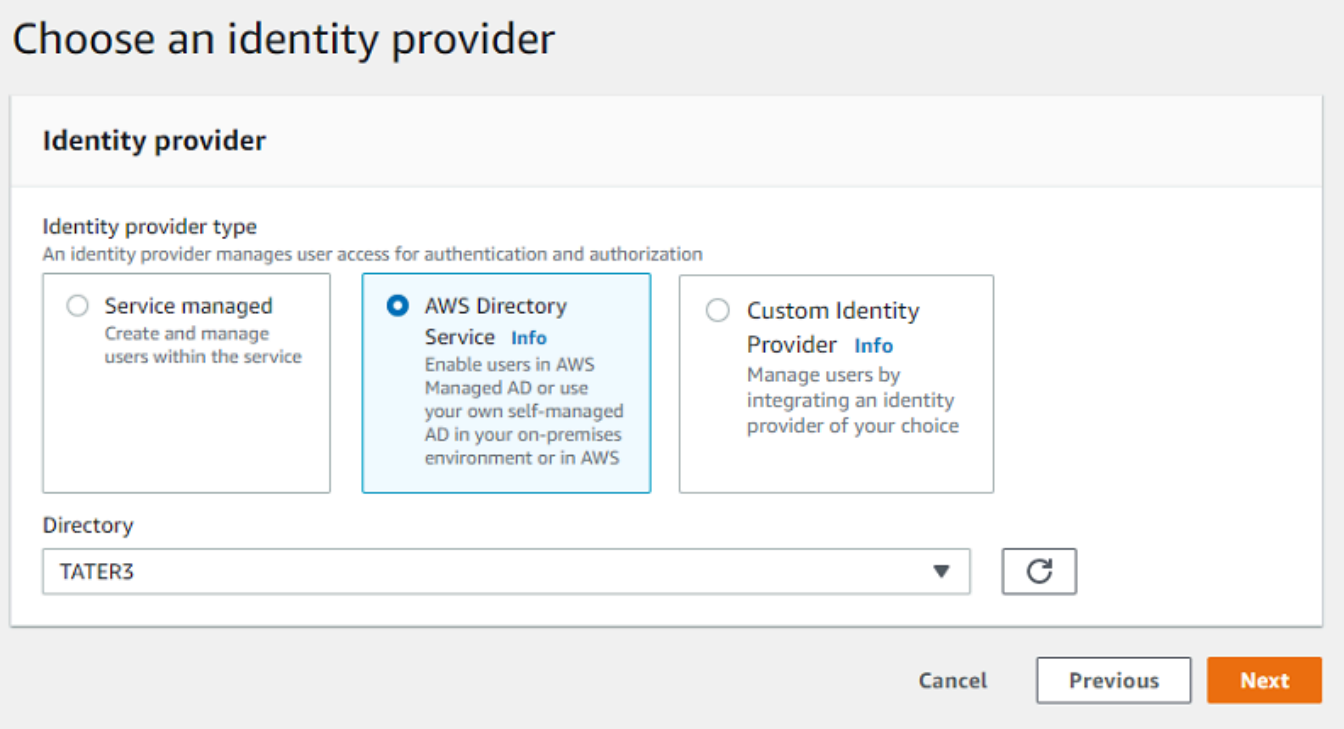
   > **Note**
   >
   > 证书必须是指定了 FQDN 或 IP 地址且具有有关颁发者的信息的有效 SSL/TLS X.509 版本 3 证书。



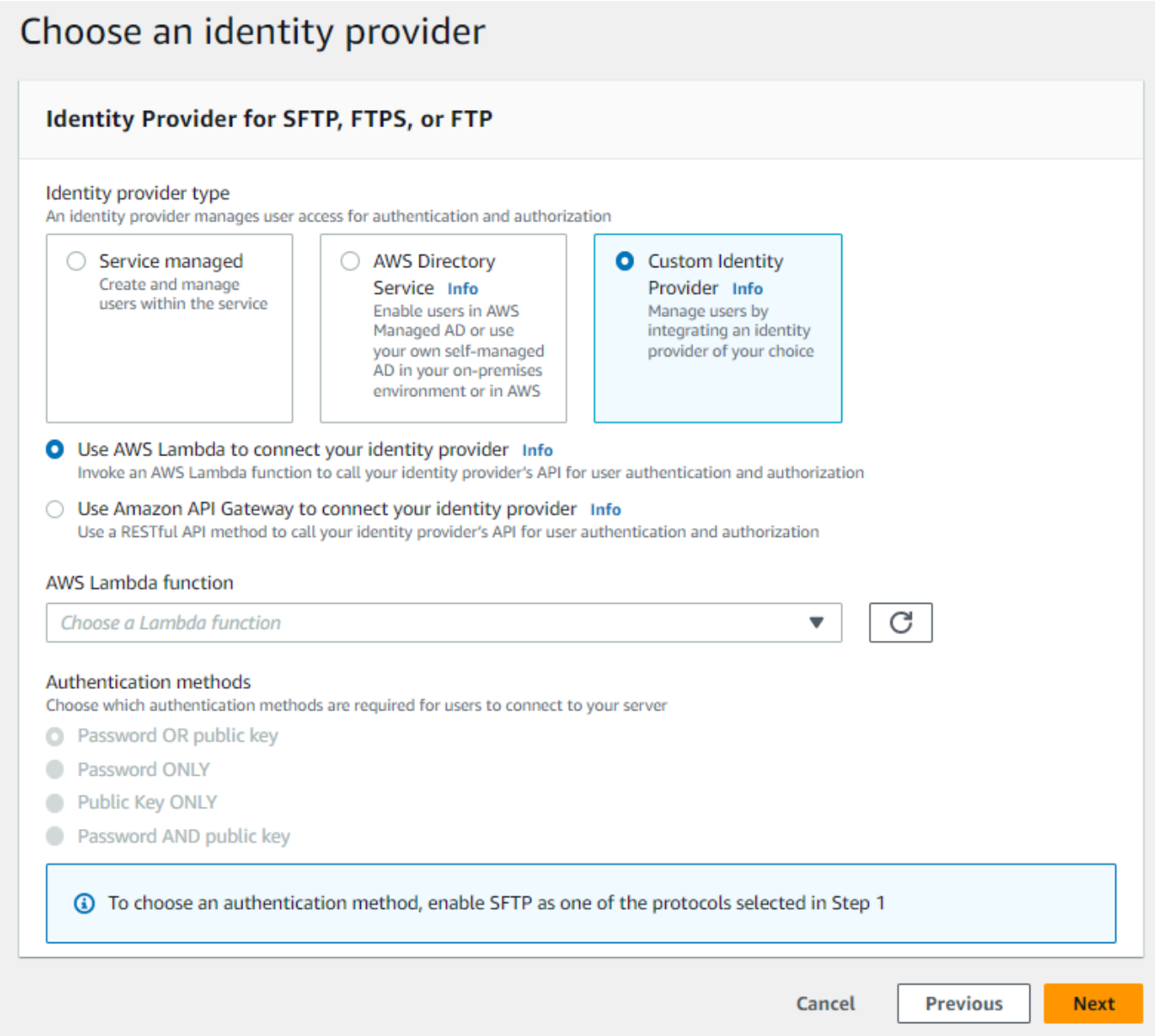
3. 在选择身份提供商中，选择要用于管理用户访问权限的身份提供商。您有以下选项：

   - AWS Directory Service for Microsoft Active Directory— 您提供用于访问终端节点的 AWS Directory Service 目录。这样，您就可以使用存储在 Activity Directory 中的凭证对用户进行身份验证。要了解有关与 AWS Managed Microsoft AD 身份提供商合作的更多信息，请参阅使用 Di AWS rectory Service 身份提供商。

     > **Note**
     >
     > - 不支持跨账户目录和共享目录。 AWS Managed Microsoft AD
     > - 要设置以 Directory Service 作为身份提供者的服务器，您需要添加一些 AWS Directory Service 权限。有关更多信息，请参阅 开始使用之前 AWS Directory Service for Microsoft Active Directory。



- 自定义身份提供商 — 请选择以下任一选项：
  - AWS Lambda 用于连接您的身份提供商-您可以使用由 Lambda 函数支持的现有身份提供商。您提供 Lambda 函数名称。有关更多信息，请参阅 AWS Lambda 用于整合您的身份提供商。
  - 使用 Amazon API Gateway 连接您的身份提供商 — 您可以创建由 Lambda 函数支持的 API 网关方法以用作身份提供商。您提供一个 Amazon API Gateway URL 和一个调用角色。有关更多信息，请参阅 使用 Amazon API Gateway 整合您的身份提供程序。



4. 选择下一步。
5. 在选择端点中，执行以下操作：

> **Note**
>
> Transfer Family 的 FTPS 服务器通过端口 21（控制通道）和端口范围 8192–8200（数据通道）运行。

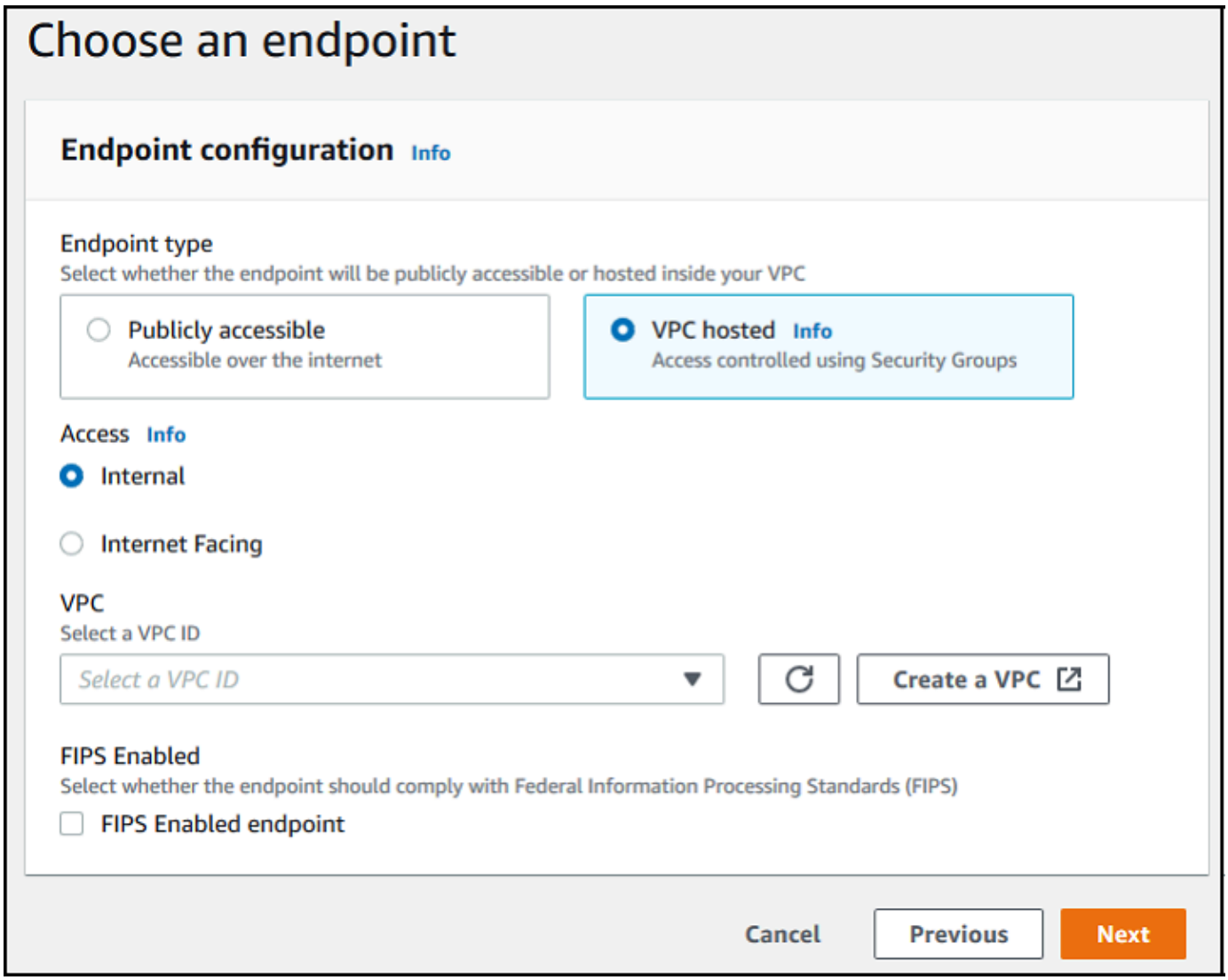
a. 对于端点类型，选择托管服务器端点的 VPC 托管端点类型。有关设置 VPC 主机端点的信息，请参阅[在虚拟私有云中创建服务器]。

> **Note**
>
> 不支持可公共访问的端点。

b. （可选）对于启用 FIPS，请选中启用 FIPS 端点复选框以确保端点符合联邦信息处理标准 (FIPS)。

> **Note**
>
> 启用 FIPS 的端点仅在北美 AWS 地区可用。有关可用区域，请参阅AWS 一般参考中的AWS Transfer Family 端点和限额。有关 FIPS 的更多信息，请参阅联邦信息处理标准 (FIPS) 140-2。

c. 选择下一步。



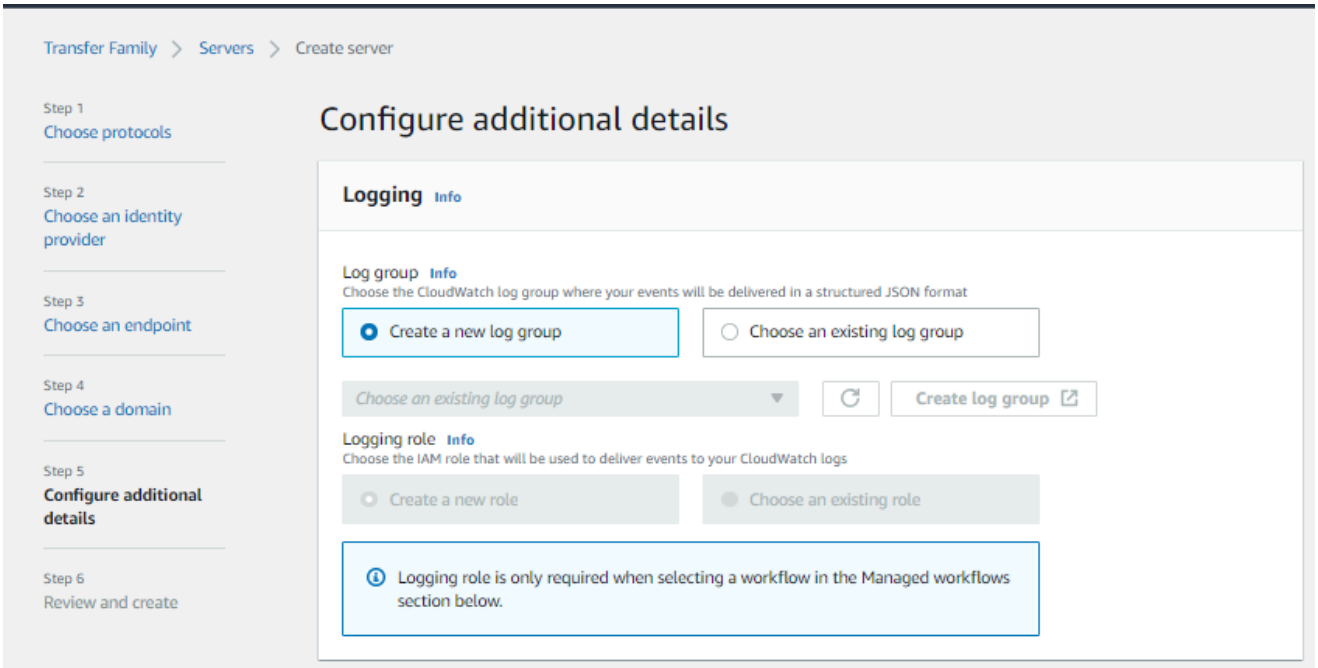
6. 在“选择域”页面上，选择要用于通过所选协议 AWS 存储和访问数据的存储服务：

   - 选择 Amazon S3，通过所选协议将您的文件作为对象存储和访问。
   - 选择 Amazon EFS，通过所选协议在 Amazon EFS 文件系统中存储和访问您的文件。

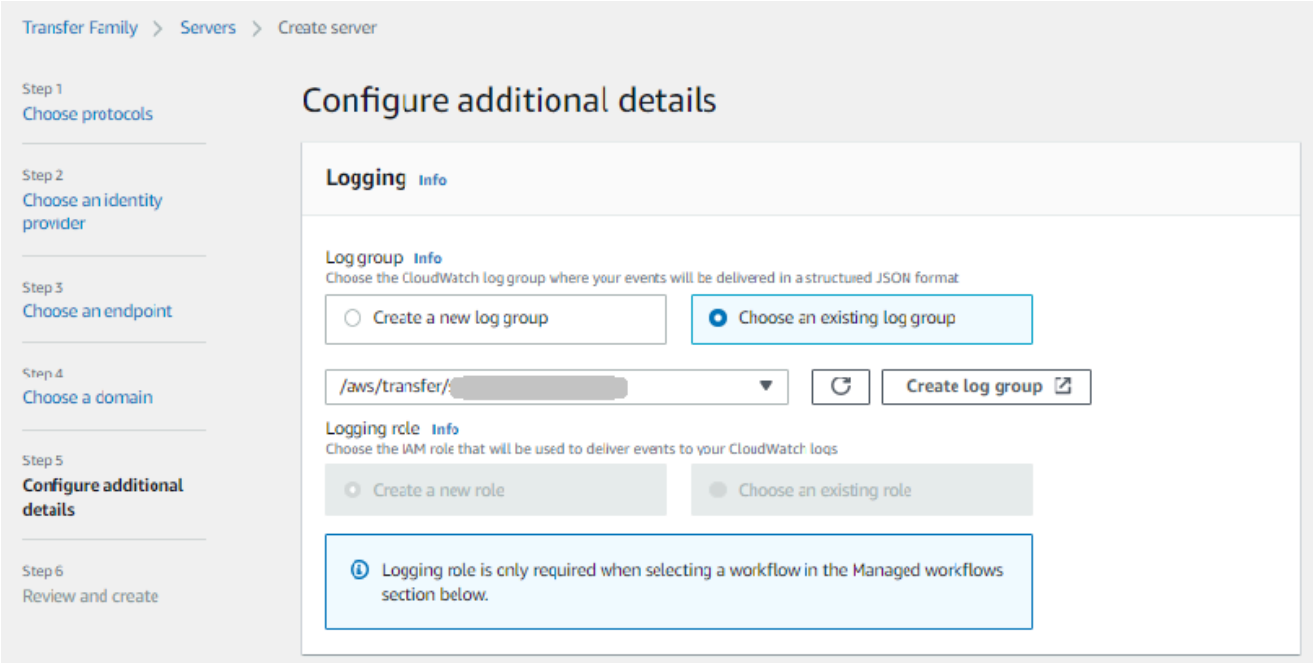
   选择下一步。

7. 在配置其他详细信息中，执行以下操作：

   a. 对于日志记录，指定现有日志组或创建新日志组（默认选项）。



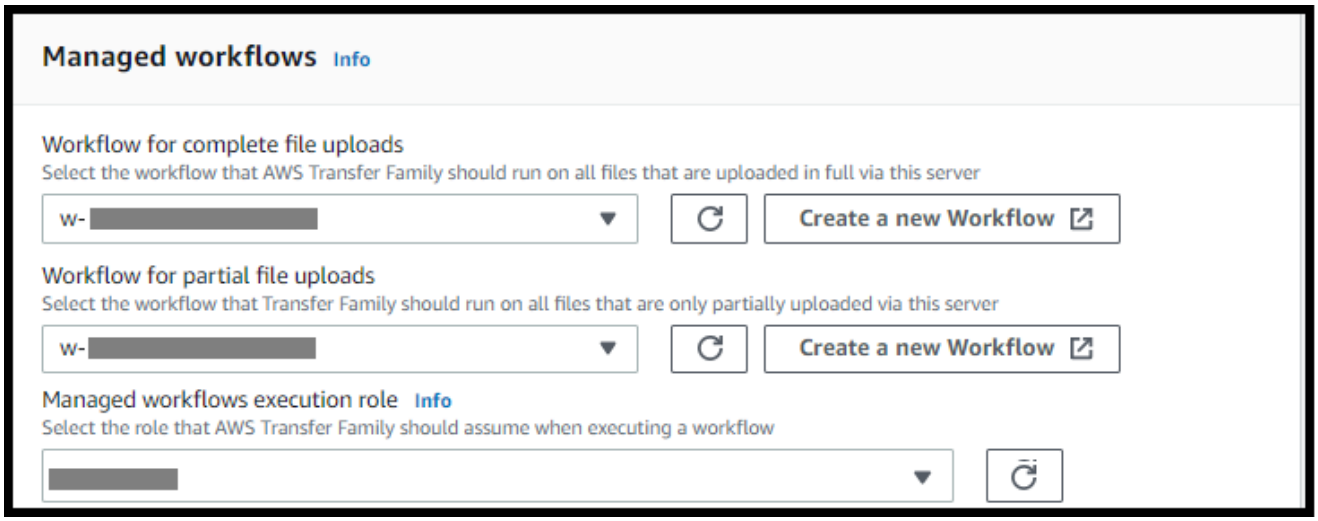
如果您选择现有日志组，则必须选择与您的日志组关联的日志组 AWS 账户。



如果选择 “创建日志组”，则 CloudWatch 控制台 (https://console.aws.amazon.com/cloudwatch/) 将打开 “创建日志组” 页面。有关详细信息，请参阅在 Log CloudWatch s 中创建日志组。

b. （可选）对于托管工作流程，请选择 Transfer Family 在执行工作流程时应承担的工作流程 ID（和相应的角色）。您可以选择一个工作流程在完成上传后执行，选择另一个工作流程在

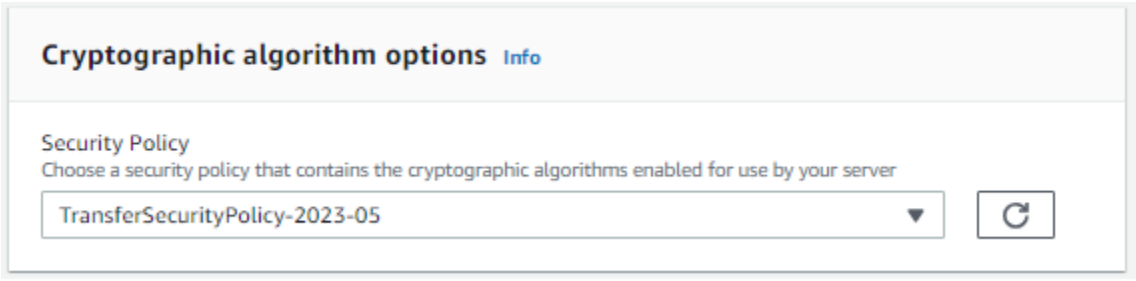
部分上传时执行。要了解有关使用托管工作流程处理文件的更多信息，请参阅AWS Transfer Family 托管工作流程。



c. 对于加密算法选项，请选择包含允许服务器使用的加密算法的安全策略。

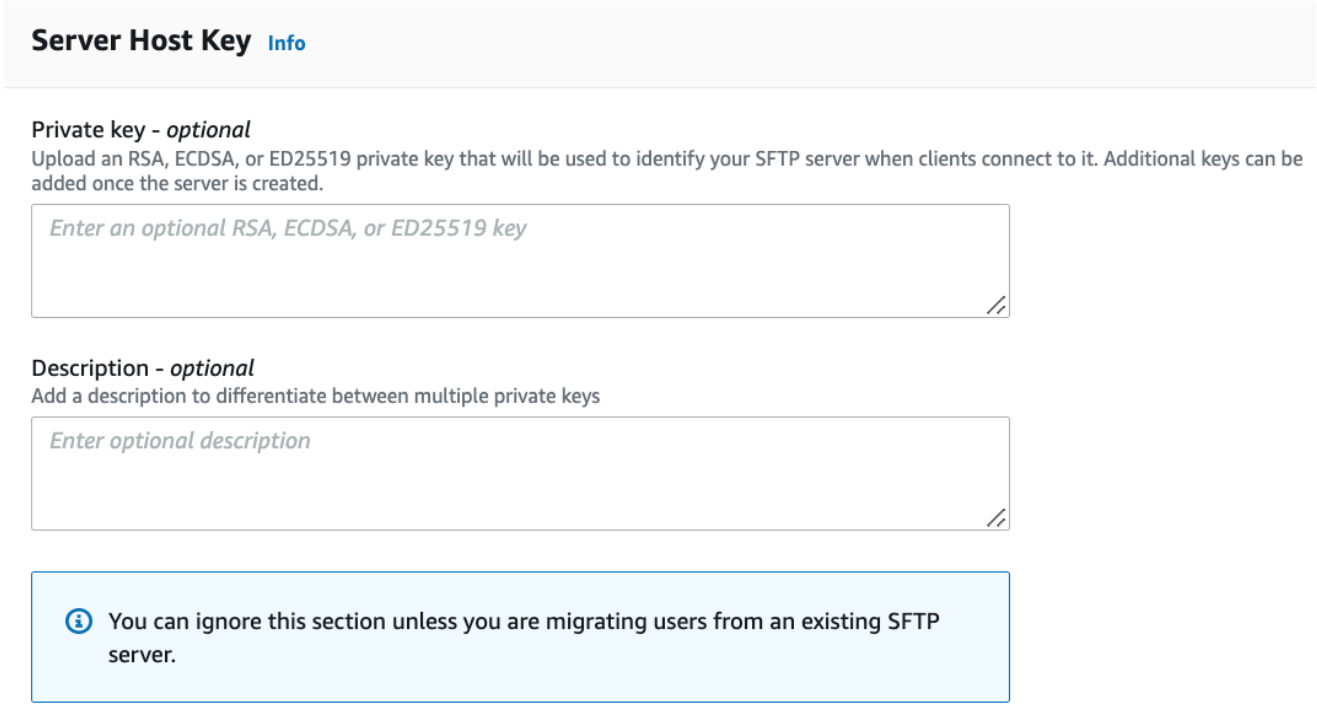
> **Note**
>
> 默认情况下：
>
> - 如果未选择启用 FIPS 的端点，则`TransferSecurityPolicy-2020-06`安全策略将附加至您的服务器。
> - 如果已选择启用 FIPS 的端点，则`TransferSecurityPolicy-FIPS-2020-06`安全策略将附加至您的服务器。

有关安全策略的更多信息，请参阅AWS Transfer Family 服务器的安全策略。



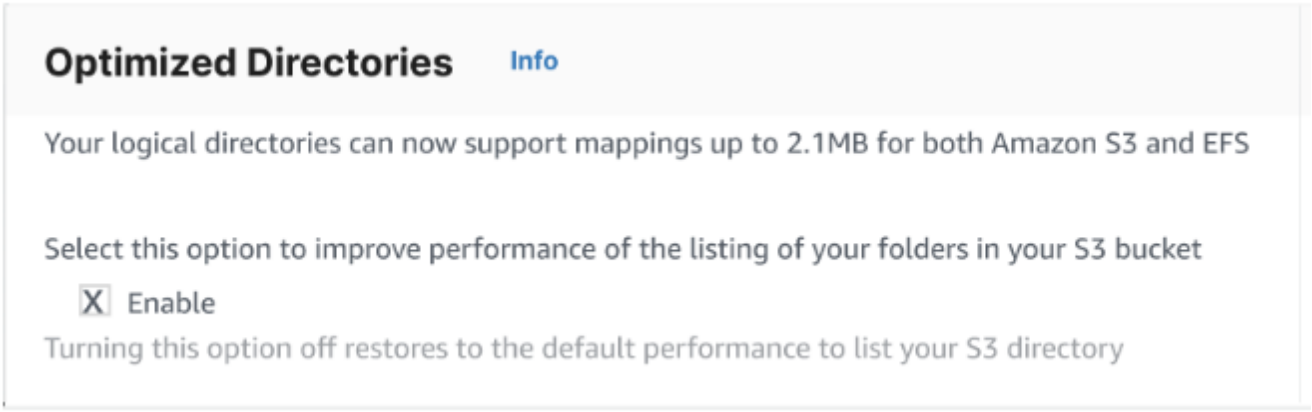
d. 对于服务器主机密钥，请将其留空。

> **Note**
>
> 服务器主机密钥部分仅用于从启用 SFTP 的现有服务器迁移用户。

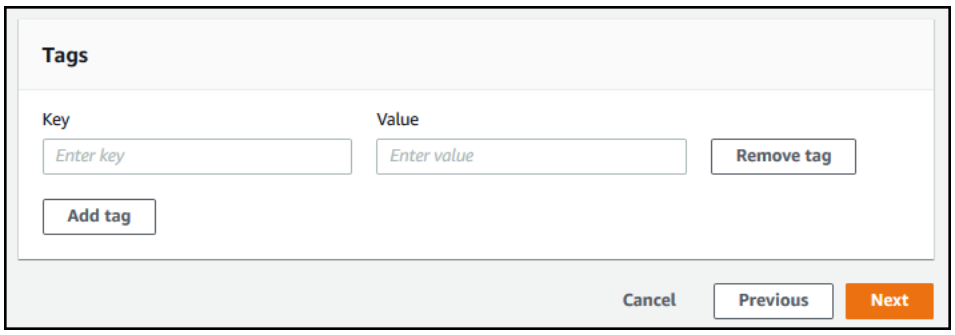


e. （可选）对于标签，在密钥和值中，输入一个或多个标签作为键值对，然后选择添加标签。

f. 您可以优化 Amazon S3 目录的性能。例如，假设您进入主目录，并且有 10,000 个子目录。换句话说，您的 S3 存储桶有 10,000 个文件夹。在这种情况下，如果您运行 `ls` (list) 命令，则列表操作需要六到八分钟。但是，如果您优化目录，则此操作只需要几秒钟。

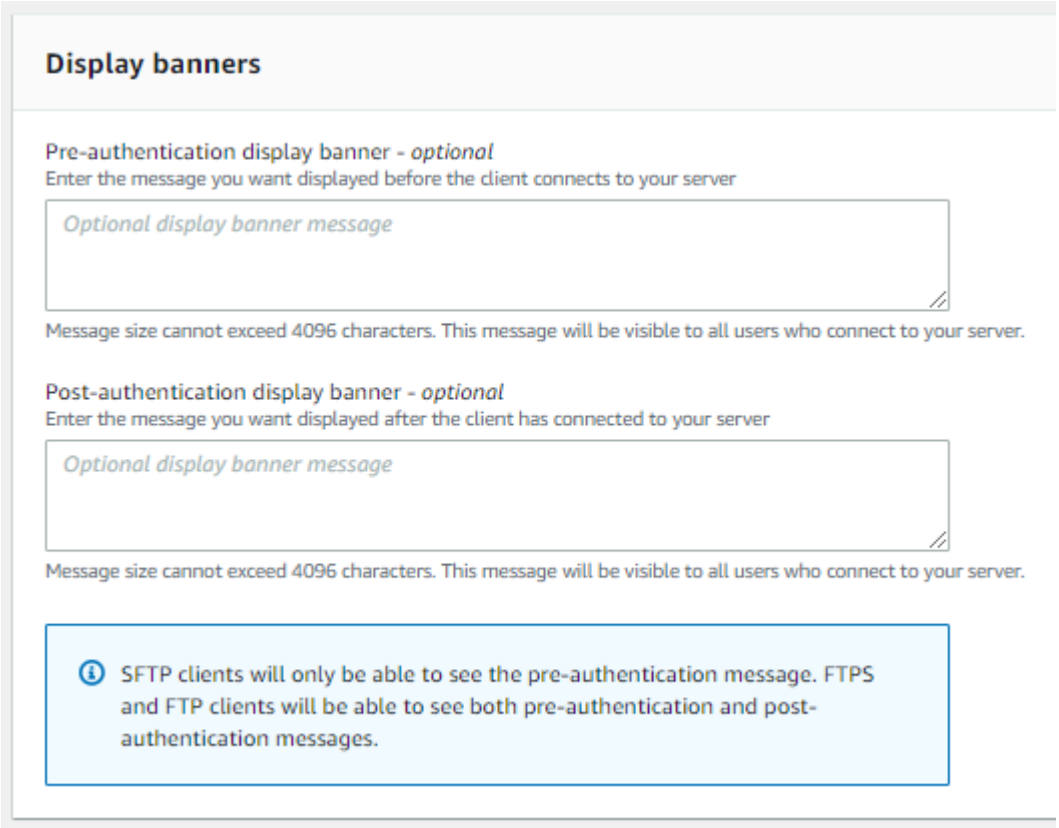


g. 选择下一步。



h. （可选）您可以将 AWS Transfer Family 服务器配置为向最终用户显示自定义消息，例如组织政策或条款和条件。您还可以向成功通过身份验证的用户显示自定义的每日消息 (MOTD)。

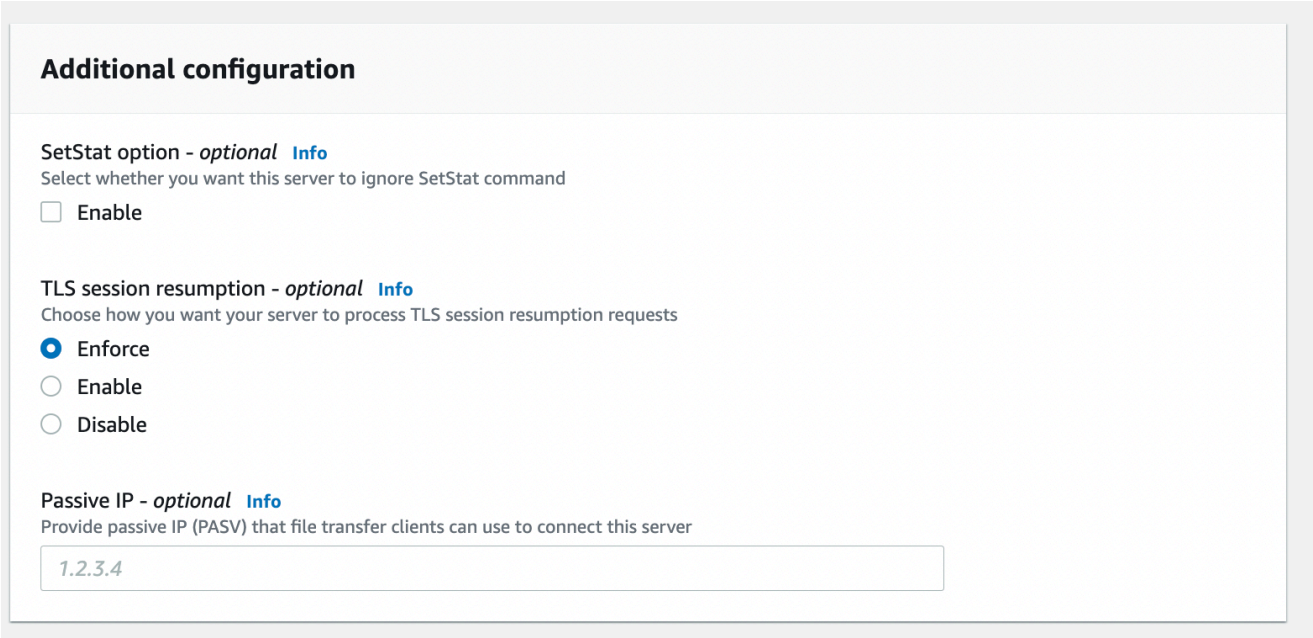
对于显示横幅，在预身份验证显示横幅文本框中，输入要在用户进行身份验证之前向他们显示的短信，然后在后身份验证显示横幅文本框中，输入要在用户成功进行身份验证后向他们显示的文本。



i. （可选）您可以配置以下其他选项。

- SetStat 选项：启用此选项可忽略客户端尝试对您上传到 Amazon S3 存储桶的文件使用SETSTAT时生成的错误。有关其他详细信息，请参阅ProtocolDetails主题中的SetStatOption文档。

- TLS 会话恢复：提供一种机制来恢复或共享 FTPS 会话的控制和数据连接之间协商的私有密钥。有关其他详细信息，请参阅ProtocolDetails主题中的`TlsSessionResumptionMode`文档。
- 被动 IP：表示 FTP 和 FTPS 协议的被动模式。输入一个 IPv4 地址，例如防火墙、路由器或负载均衡器的公有 IP 地址。有关其他详细信息，请参阅ProtocolDetails主题中的`PassiveIp`文档。

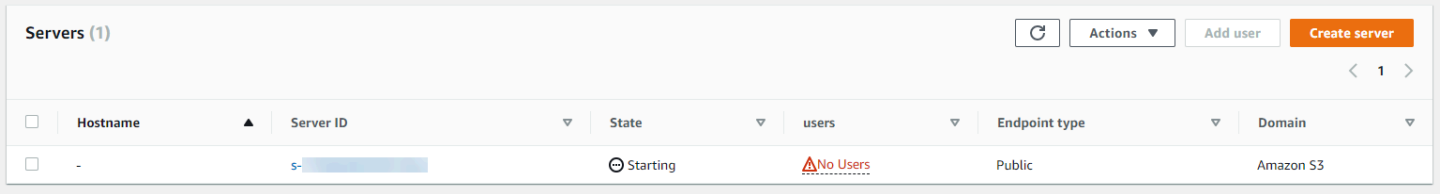


8. 在审核和创建页面上，审核您的选择。
   - 如果要编辑其中任何一个，请选择该步骤旁边的编辑。

     > **Note**
     >
     > 在选择编辑的步骤之后，您必须审核每个步骤。

   - 如果没有任何更改，请选择创建服务器来创建您的服务器。您将转至如下所示的 Servers (服务器) 页面，其中列出了您的新服务器。

您的新服务器状态更改为在线可能需要几分钟时间。到时候，您的服务器可以执行用户的文件操作。



后续步骤：对于下一步，请继续前往使用自定义身份提供程序设置用户。

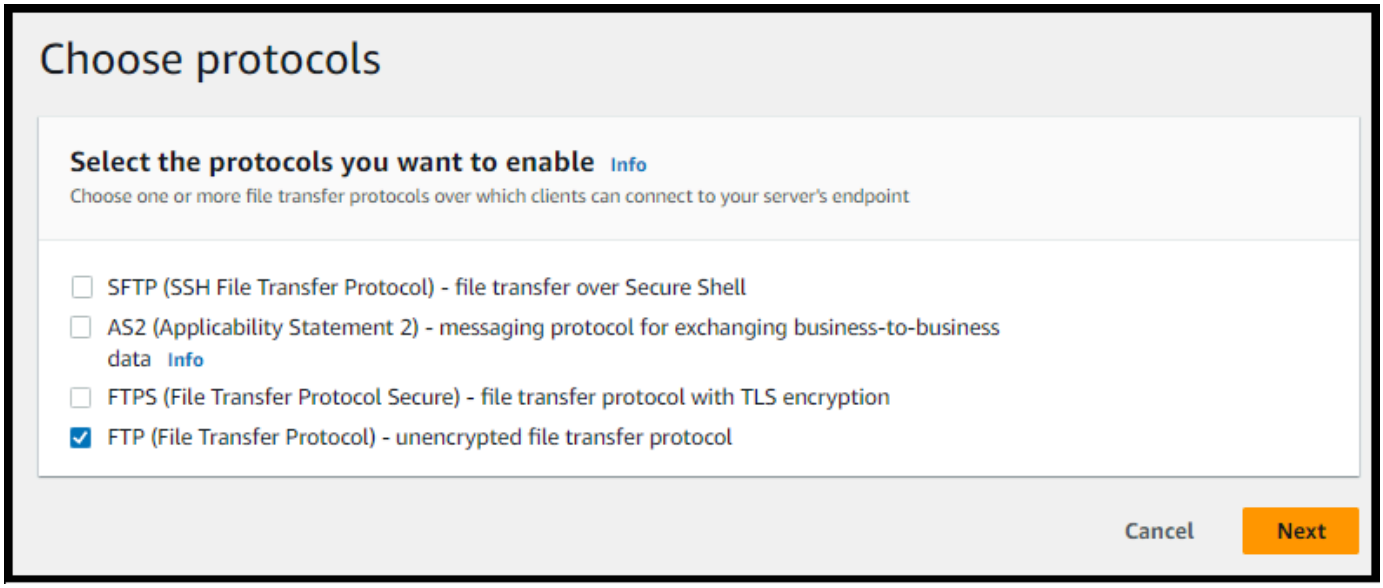
创建启用 FTP 的服务器

文件传输协议 (FTP) 是一种用于数据传输的网络协议。FTP 使用单独的通道进行控制和数据传输。控制通道保持打开状态，直至终止或不活动超时状态。数据通道在传输期间处于活动状态。FTP 使用明文且不支持流量加密。

> **Note**
>
> 启用 FTP 时，必须为托管 VPC 的终端节点选择内部访问选项。如果您需要服务器让数据通过公共网络，则必须使用安全协议，例如 SFTP 或 FTPS。

创建启用 FTP 的服务器

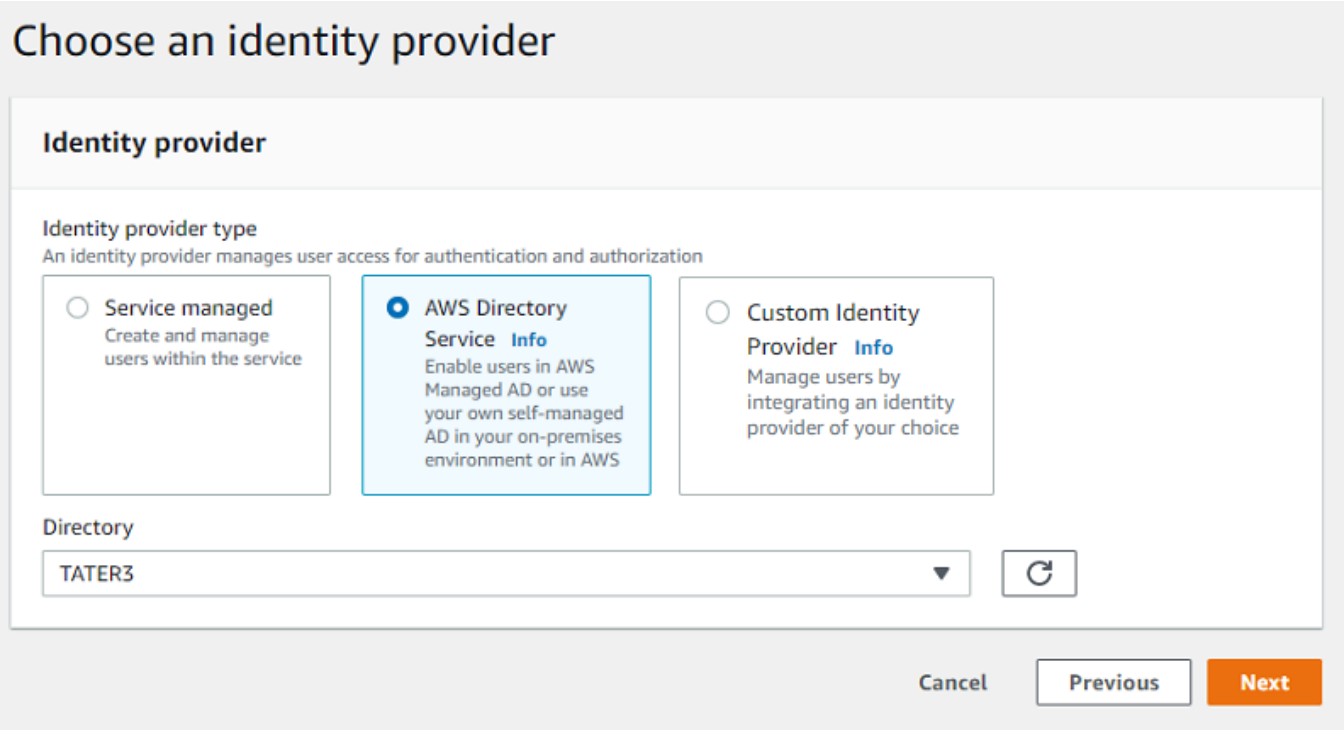
1. 打开 AWS Transfer Family 控制台 https://console.aws.amazon.com/transfer/ 并从导航窗格中选择 “服务器”，然后选择 “创建服务器”。
2. 在选择协议中，选择 FTP，然后选择下一步。



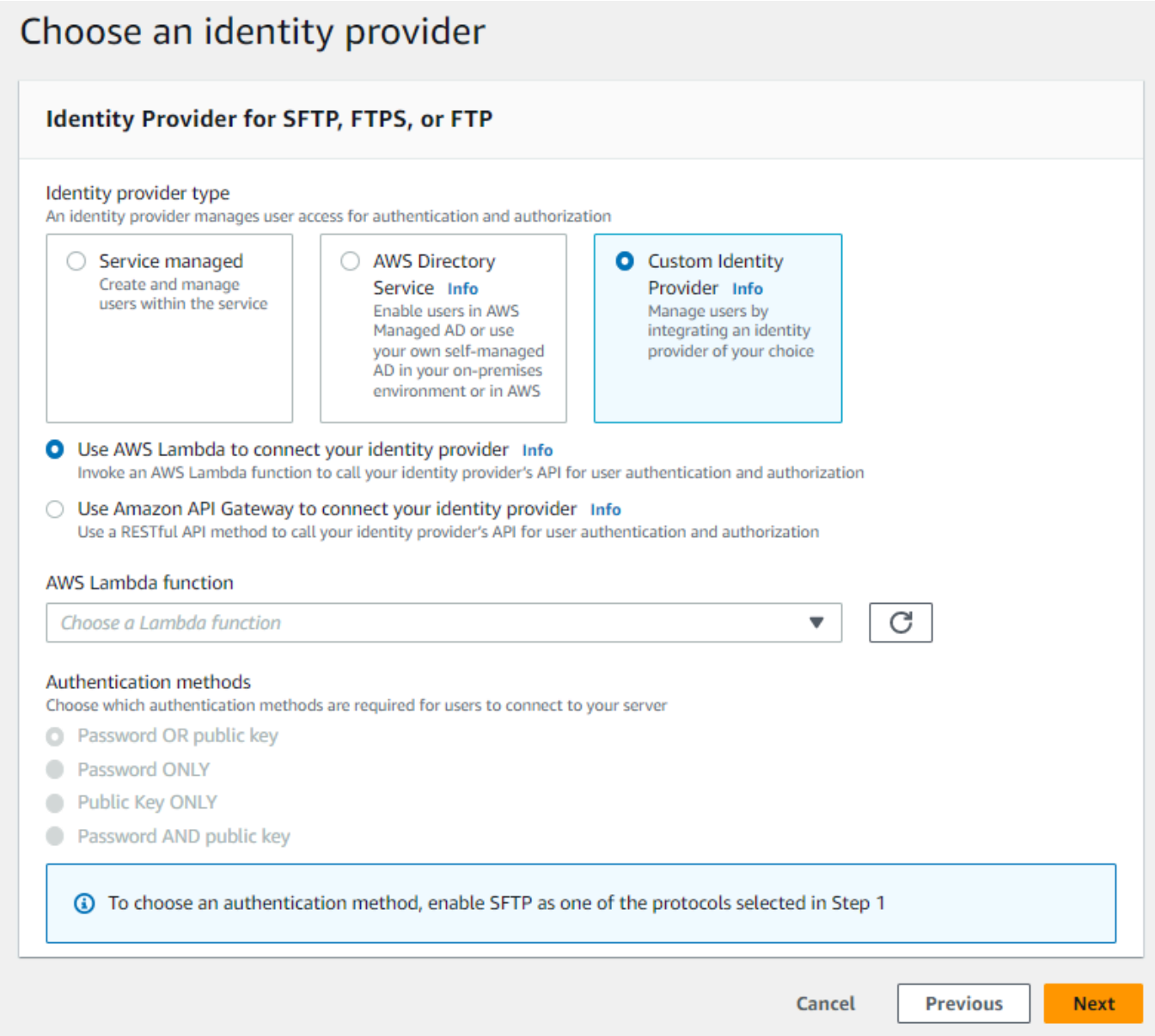
3. 在选择身份提供商中，选择要用于管理用户访问权限的身份提供商。您有以下选项：

- AWS Directory Service for Microsoft Active Directory— 您提供用于访问终端节点的 AWS Directory Service 目录。这样，您就可以使用存储在 Activity Directory 中的凭证对用户进行身份验证。要了解有关与 AWS Managed Microsoft AD 身份提供商合作的更多信息，请参阅使用 Di AWS rectory Service 身份提供商。

> **Note**
>
> - 不支持跨账户目录和共享目录。 AWS Managed Microsoft AD
> - 要设置以 Directory Service 作为身份提供者的服务器，您需要添加一些 AWS Directory Service 权限。有关更多信息，请参阅 开始使用之前 AWS Directory Service for Microsoft Active Directory。



- 自定义身份提供商 — 请选择以下任一选项：
  - AWS Lambda 用于连接您的身份提供商-您可以使用由 Lambda 函数支持的现有身份提供商。您提供 Lambda 函数名称。有关更多信息，请参阅 AWS Lambda 用于整合您的身份提供商。
  - 使用 Amazon API Gateway 连接您的身份提供商 — 您可以创建由 Lambda 函数支持的 API 网关方法以用作身份提供商。您提供一个 Amazon API Gateway URL 和一个调用角色。有关更多信息，请参阅 使用 Amazon API Gateway 整合您的身份提供程序。



4. 选择下一步。
5. 在选择端点中，执行以下操作：

> **Note**
>
> Transfer Family 的 FTP 服务器通过端口 21（控制通道）和端口范围 8192–8200（数据通道）运行。

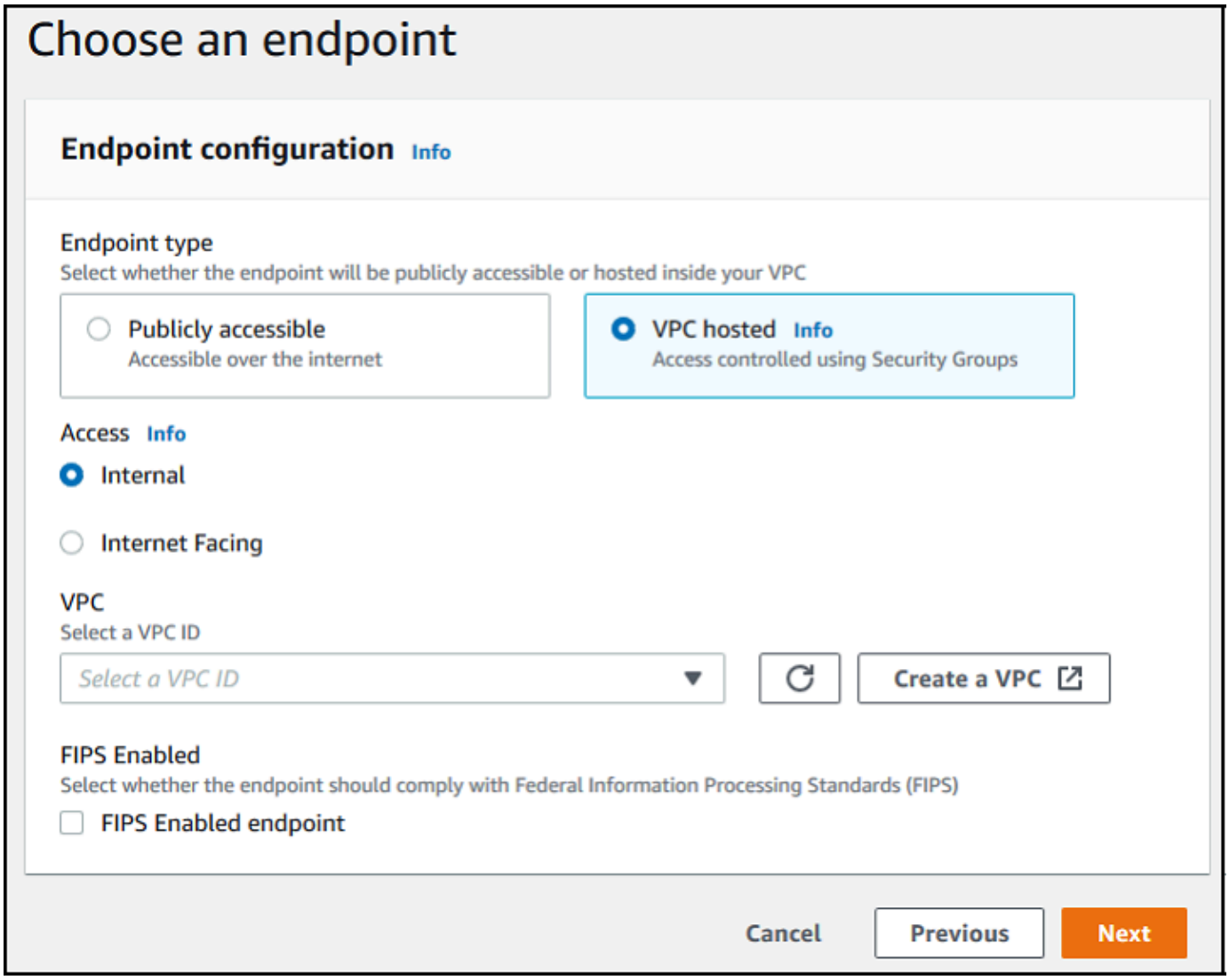
a. 对于端点类型，选择托管服务器端点的 VPC 托管。有关设置 VPC 主机端点的信息，请参阅在虚拟私有云中创建服务器。

> **Note**
>
> 不支持可公共访问的端点。

b. 对于启用 FIPS，请清除启用 FIPS 端点复选框。

> **Note**
>
> FTP 服务器不支持启用 FIPS 的端点。

c. 选择下一步。



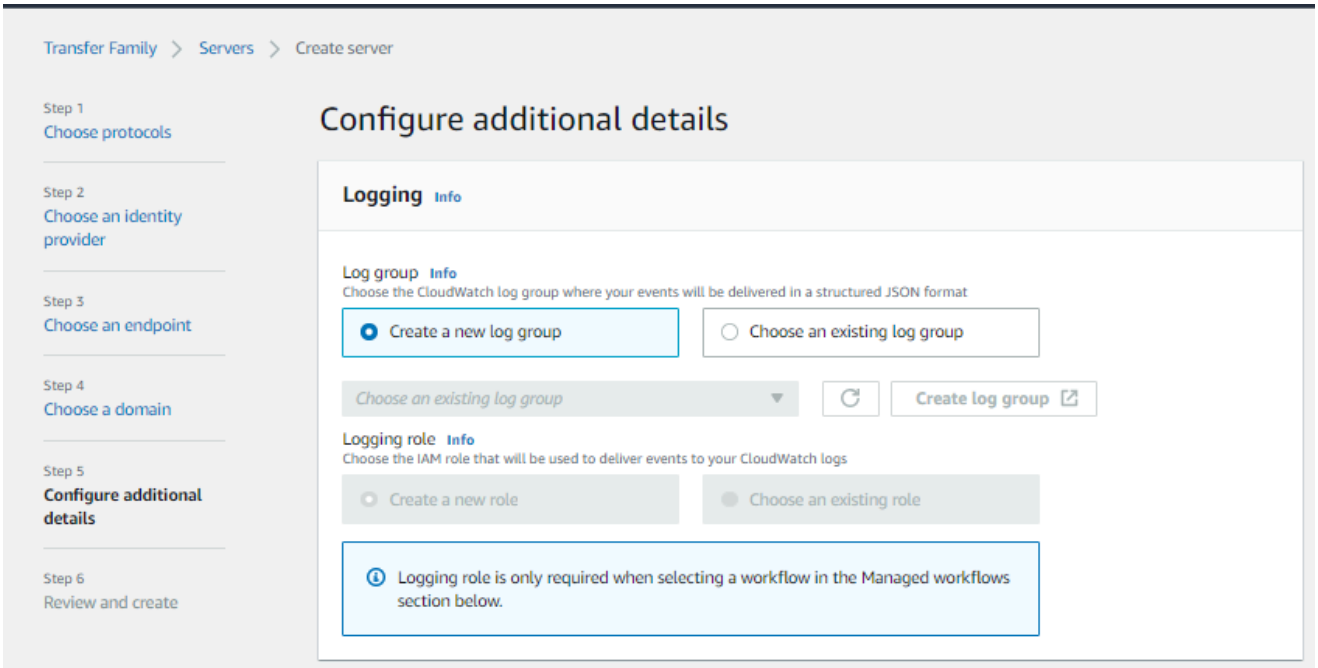
6. 在“选择域”页面上，选择要用于通过所选协议 AWS 存储和访问数据的存储服务。

- 选择 Amazon S3，通过所选协议将您的文件作为对象存储和访问。
- 选择 Amazon EFS，通过所选协议在 Amazon EFS 文件系统中存储和访问您的文件。

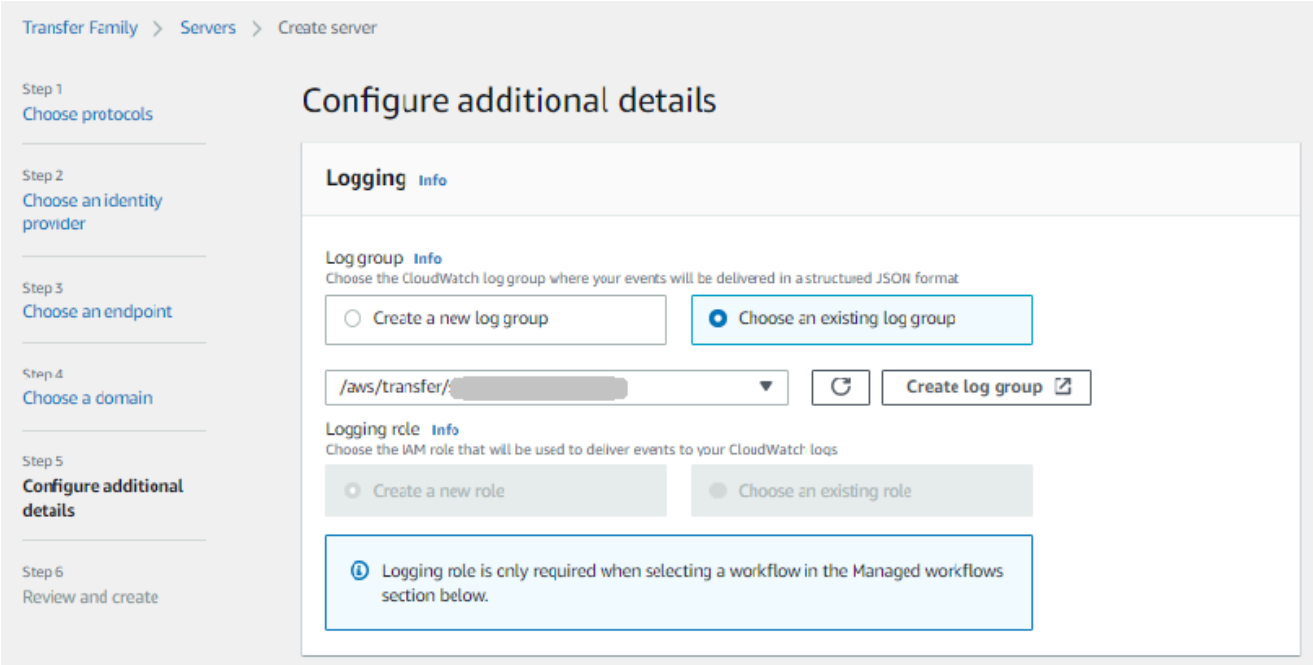
选择下一步。

7. 在配置其他详细信息中，执行以下操作：

   a. 对于日志记录，指定现有日志组或创建新日志组（默认选项）。

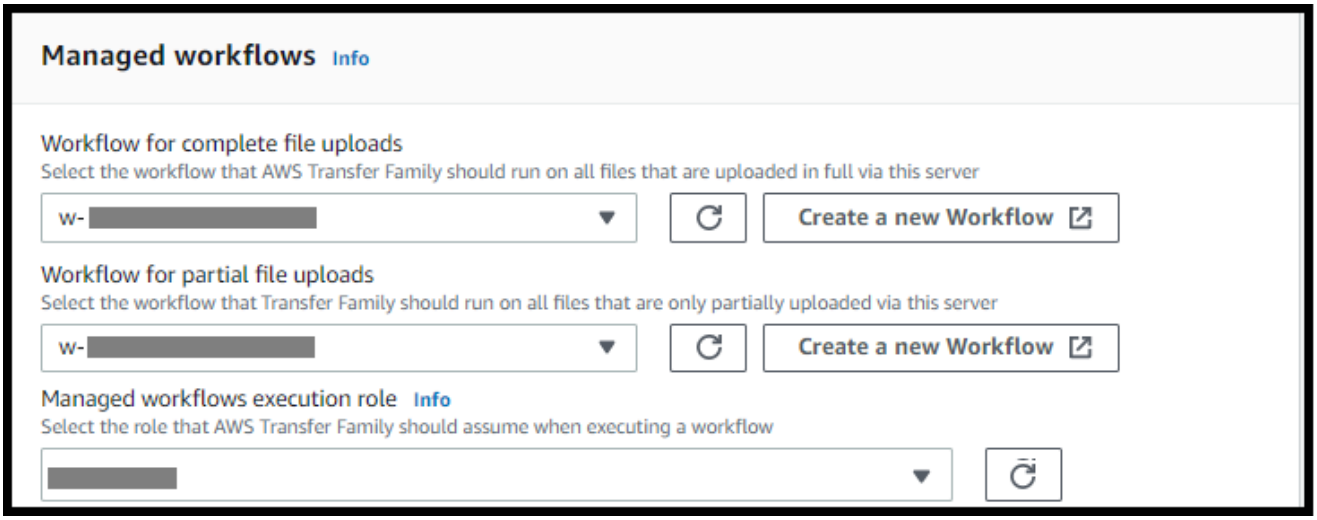


   如果您选择现有日志组，则必须选择与您的日志组关联的日志组 AWS 账户。



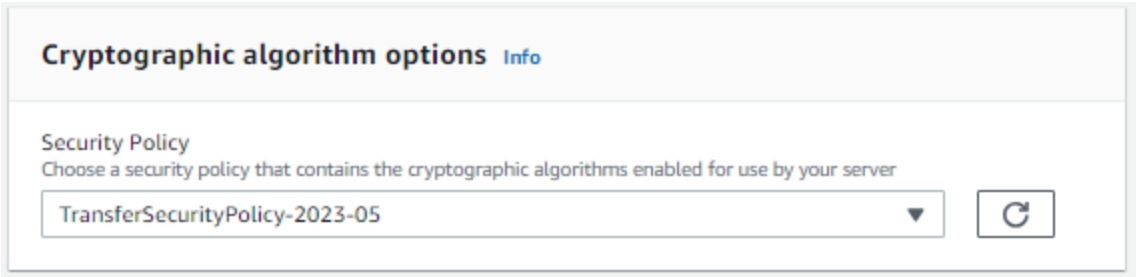
如果选择 “创建日志组”，则 CloudWatch 控制台 (https://console.aws.amazon.com/cloudwatch/) 将打开 “创建日志组” 页面。有关详细信息，请参阅在 Log CloudWatch s 中创建日志组。

b. （可选）对于托管工作流程，请选择 Transfer Family 在执行工作流程时应承担的工作流程 ID（和相应的角色）。您可以选择一个工作流程在完成上传后执行，选择另一个工作流程在部分上传时执行。要了解有关使用托管工作流程处理文件的更多信息，请参阅AWS Transfer Family 托管工作流程。



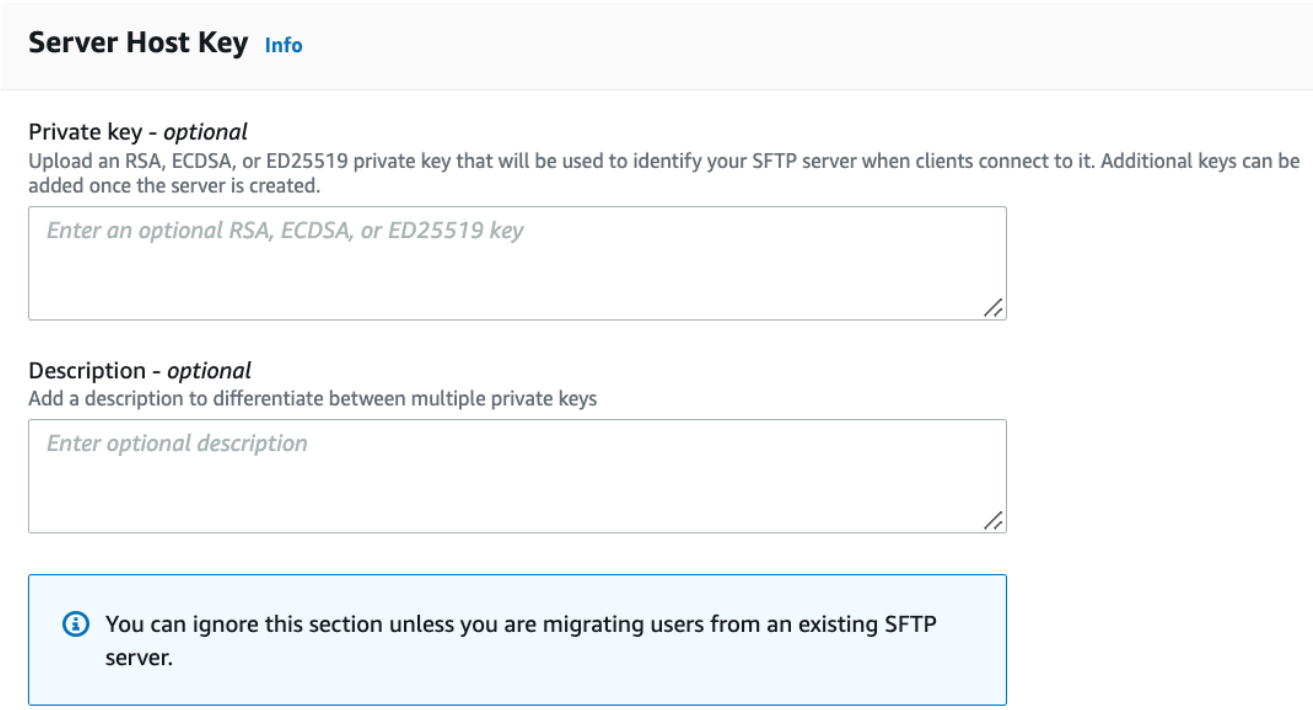
c. 对于加密算法选项，请选择包含允许服务器使用的加密算法的安全策略。

> **Note**
>
> Transfer Family 会将最新的安全策略分配给你的 FTP 服务器。但是，由于 FTP 协议不使用任何加密，因此 FTP 服务器不使用任何安全策略算法。除非您的服务器也使用 FTPS 或 SFTP 协议，否则安全策略将保持未使用状态。



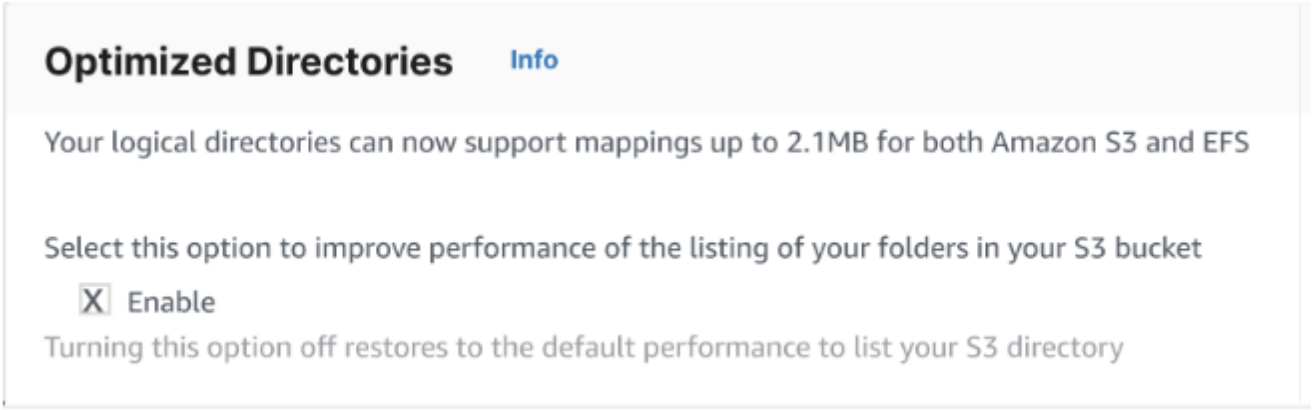
d. 对于服务器主机密钥，请将其留空。

> **Note**
>
> 服务器主机密钥部分仅用于从启用 SFTP 的现有服务器迁移用户。

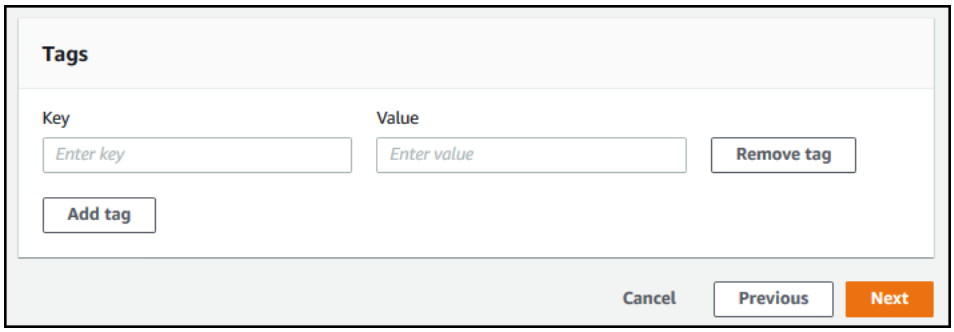


e. （可选）对于标签，在密钥和值中，输入一个或多个标签作为键值对，然后选择添加标签。

f. 您可以优化 Amazon S3 目录的性能。例如，假设您进入主目录，并且有 10,000 个子目录。换句话说，您的 S3 存储桶有 10,000 个文件夹。在这种情况下，如果您运行 `ls` (list) 命令，则列表操作需要六到八分钟。但是，如果您优化目录，则此操作只需要几秒钟。

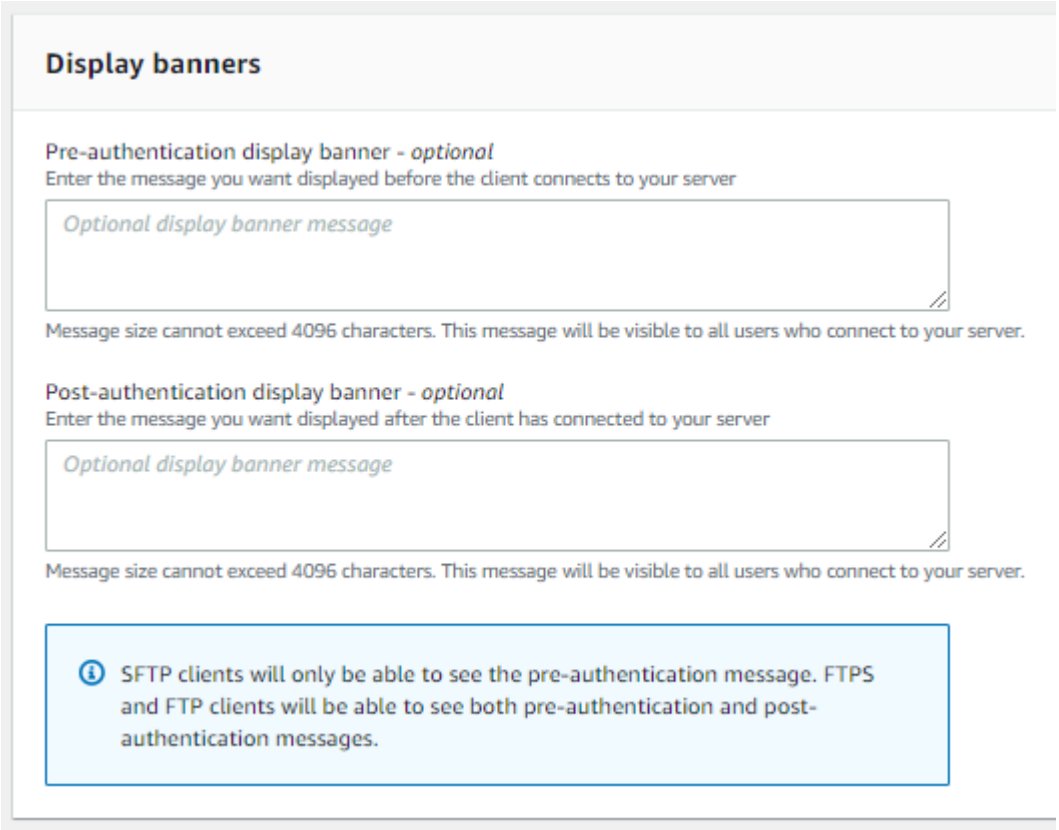


g. 选择下一步。



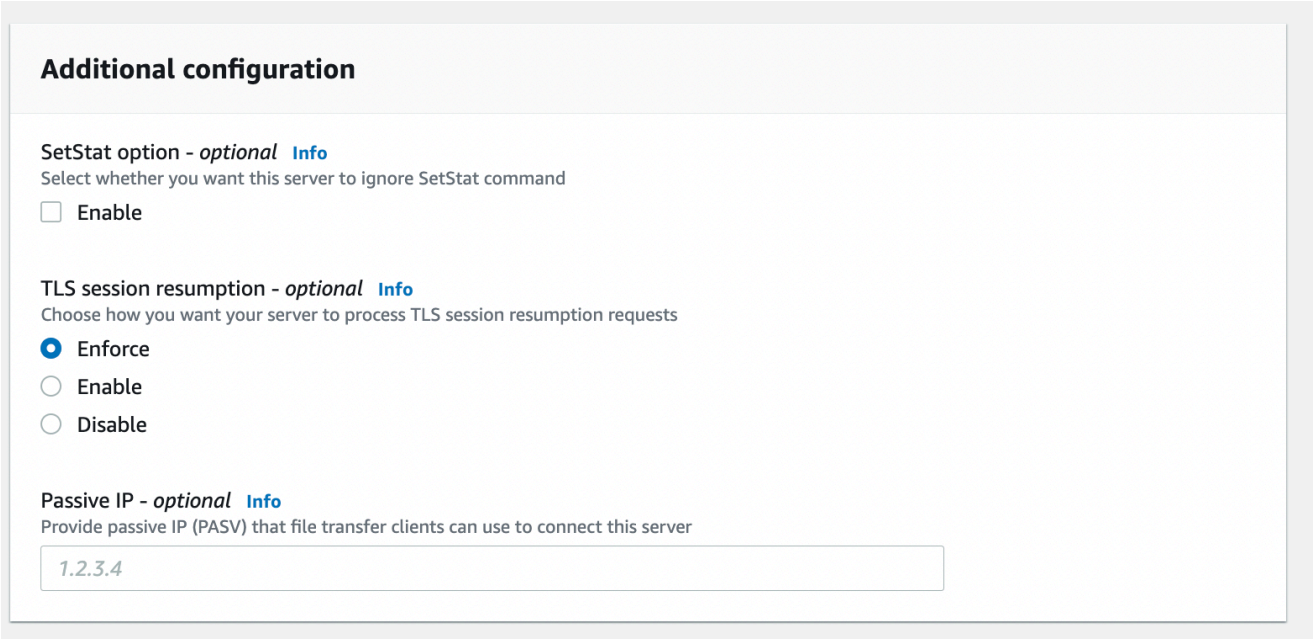
h. （可选）您可以将 AWS Transfer Family 服务器配置为向最终用户显示自定义消息，例如组织政策或条款和条件。您还可以向成功通过身份验证的用户显示自定义的每日消息 (MOTD)。

对于显示横幅，在预身份验证显示横幅文本框中，输入要在用户进行身份验证之前向他们显示的短信，然后在后身份验证显示横幅文本框中，输入要在用户成功进行身份验证后向他们显示的文本。



i. （可选）您可以配置以下其他选项。

- SetStat 选项：启用此选项可忽略客户端尝试对您上传到 Amazon S3 存储桶的文件使用SETSTAT时生成的错误。有关其他详细信息，请参阅ProtocolDetails主题中的SetStatOption文档。
- TLS 会话恢复：提供一种机制来恢复或共享 FTPS 会话的控制和数据连接之间协商的私有密钥。有关其他详细信息，请参阅ProtocolDetails主题中的TlsSessionResumptionMode文档。
- 被动 IP：表示 FTP 和 FTPS 协议的被动模式。输入一个 IPv4 地址，例如防火墙、路由器或负载均衡器的公有 IP 地址。有关其他详细信息，请参阅ProtocolDetails主题中的PassiveIp文档。



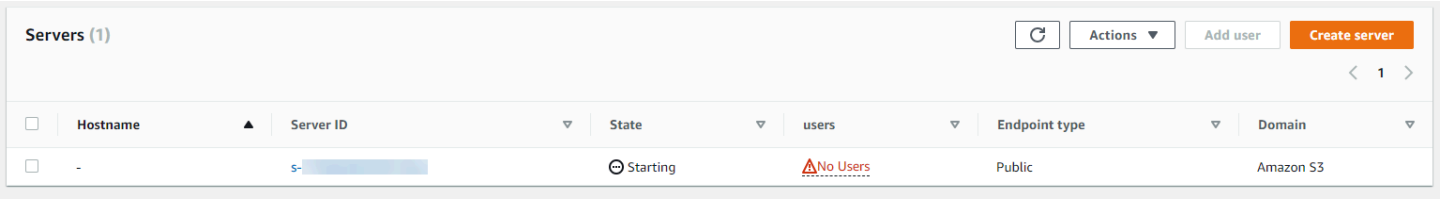
8. 在审核和创建页面上，审核您的选择。

   - 如果要编辑其中任何一个，请选择该步骤旁边的编辑。

     > **Note**
     >
     > 在选择编辑的步骤之后，您必须审核每个步骤。

   - 如果没有任何更改，请选择创建服务器来创建您的服务器。您将转至如下所示的 Servers (服务器) 页面，其中列出了您的新服务器。

您的新服务器状态更改为在线可能需要几分钟时间。到时候，您的服务器可以执行用户的文件操作。



后续步骤 — 对于下一步，请继续前往使用自定义身份提供程序设置用户。

在虚拟私有云中创建服务器

您可以将服务器的端点托管在虚拟私有云 (VPC) 中，用于在不通过公共互联网的情况下向 Amazon S3 存储桶或 Amazon EFS 文件系统传输数据和从 Amazon S3 存储桶或 Amazon EFS 文件系统中传输数据。

> **Note**
>
> 2021 年 5 月 19 日之后，如果您的账户在 2021 年 5 月 19 日之前尚未在`EndpointType=VPC_ENDPOINT` 账户中使用 AWS 创建服务器，则您将无法创建服务器。如果您在 2021 年 2 月 21 日当天或之前已经在 `EndpointType=VPC_ENDPOINT` 账户中使用 AWS 创建了服务器，则不会受到影响。在此日期之后，使用 `EndpointType = VPC`。有关更多信息，请参阅 the section called “停止使用 VPC_ENDPOINT”。

如果您使用 Amazon Virtual Private Cloud (Amazon VPC) 托管您的 AWS 资源，则可以在您的 VPC 和服务器之间建立私有连接。然后，您可以使用此服务器通过客户端将数据传输到您的 Amazon S3 存储桶或从您的 Amazon S3 存储桶中传输数据，而无需使用公有 IP 地址或需要互联网网关。

使用 Amazon VPC，您可以在自定义虚拟网络中启动 AWS 资源。可以使用 VPC 控制您的网络设置，例如 IP 地址范围、子网、路由表和网络网关。有关 VPC 的更多信息，请参阅 Amazon VPC 用户指南中的什么是 Amazon VPC？。

在下一部分中，查找如何创建 VPC 并将其连接到服务器的说明。作为概述，您可以按如下方式执行此操作：

1. 使用 VPC 端点设置服务器。
2. 使用 VPC 内的客户端通过 VPC 端点连接到您的服务器。这样，您就可以使用AWS Transfer Family通过客户端传输存储在 Amazon S3 存储桶中的数据。即使网络已与公共互联网断开连接，您也可以执行此传输。
3. 此外，如果您选择将服务器的端点设为面向互联网，则可以将弹性 IP 地址与您的端点相关联。这样做可以让 VPC 之外的客户端连接到您的服务器。您可以使用 VPC 安全组以控制请求仅来自允许地址的经过身份验证的用户的访问权限。

主题

- 更改 SFTP 服务器的端点类型
- 停止使用 VPC_ENDPOINT
- 将AWS Transfer Family服务器端点类型从 VPC_ENDPOINT 更新为 VPC

创建仅在您的 VPC 内访问的服务器端点

在以下步骤中，您将创建仅由您的 VPC 内的资源访问的服务器端点。

在 VPC 内创建服务器端点

1. 打开AWS Transfer Family控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 从导航窗格中，选择服务器，然后选择创建服务器。
3. 在选择协议中，选择一个或多个协议，然后选择下一步。有关协议的更多信息，请参阅 步骤 2：创建启用 SFTP 的服务器。
4. 在选择身份提供商中，选择服务托管以在 AWS Transfer Family 中存储用户身份和密钥，然后选择下一步。

   > **Note**
   >
   > 此过程使用服务托管选项。如果您选择自定义，则提供 Amazon API Gateway 端点和 AWS Identity and Access ManagementIAM 角色来访问端点。执行此操作后，您可以集成目录服务，用于对用户进行身份验证和授权。要了解有关使用自定义身份提供商的更多信息，请参阅使用自定义身份提供程序。

5. 在选择端点中，执行以下操作：

   > **Note**
   >
   > Transfer Family 的 FTP 和 FTPS 服务器通过端口 21（控制通道）和端口范围 8192–8200（数据通道）运行。

   a. 对于端点类型，选择托管服务器端点的 VPC 托管端点类型。

   b. 对于访问，请选择内部，使您的端点仅可由使用端点的私有 IP 地址的客户端访问。

> **Note**
>
> 有关面向互联网选项的详细信息，请参阅为服务器创建面向互联网的端点。在 VPC 中创建的仅用于内部访问的服务器不支持自定义主机名。

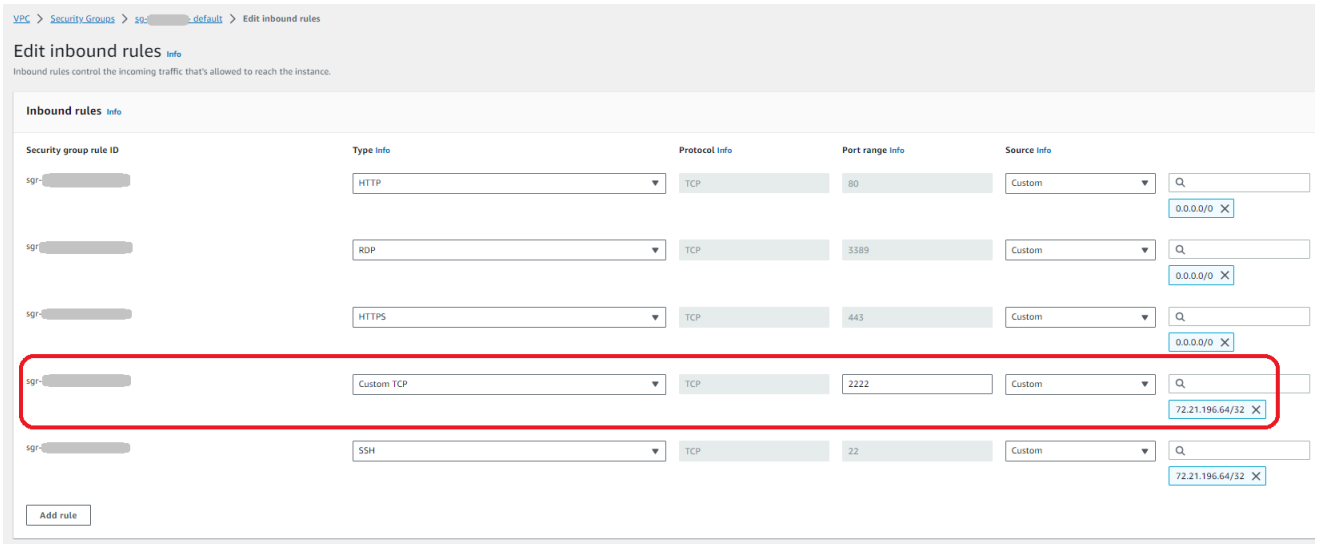
c. 对于 VPC，选择现有 VPC ID 或选择创建 VPC 以创建新的 VPC。

d. 在可用区部分，最多选择三个可用区和关联的子网。

e. 在安全组部分，选择一个或多个现有安全组 ID 或选择创建安全组来创建新的安全组。有关安全组的更多信息，请参阅 Amazon Virtual Private Cloud 用户指南中 VPC 的安全组。要创建安全组，请参阅 Amazon Virtual Private Cloud 用户指南中的创建安全组。

> **Note**
>
> 您的 VPC 会自动带有默认的安全组。如果您在启动服务器时没有指定其他安全组或组，我们会将默认安全组与您的服务器相关联。

对于安全组的入站规则，您可以将 SSH 流量配置为使用端口 22、2222 或两者兼而有之。默认配置端口 22。要使用端口 2222，您需要向安全组添加入站规则。对于类型，选择 “自定义 TCP”，然后在 **2222** “端口范围” 中输入；对于源，输入与 SSH 端口 22 规则相同的 CIDR 范围。



f. （可选）对于启用 FIPS，请选中启用 FIPS 的端点复选框以确保端点符合联邦信息处理标准 (FIPS)。

> **Note**
>
> 启用 FIPS 的端点仅在北美AWS地区可用。有关可用区域，请参阅AWS 一般参考中的AWS Transfer Family端点和限额。有关 FIPS 的更多信息，请参阅联邦信息处理标准 (FIPS) 140-2。

g. 请选择 Next（下一步）。

6. 在配置其他详细信息中，执行以下操作：

a. 要进行CloudWatch 日志记录，请选择以下选项之一以启用 Amazon CloudWatch 记录您的用户活动：

- 创建一个新角色，允许 Transfer Family 自动创建 IAM 角色，前提是您拥有创建新角色的相应权限。创建的 IAM 角色被称为`AWSTransferLoggingAccess`。
- 选择现有角色以从您的帐户中选择现有 IAM 角色。在日志记录角色下，选择该角色。此 IAM 角色应包括将服务设置为`transfer.amazonaws.com`的信任策略。

  有关 CloudWatch 日志记录的更多信息，请参阅配置 CloudWatch 日志记录角色。

> **Note**
>
> - 如果您未指定日志记录角色， CloudWatch 则无法在中查看最终用户活动。
> - 如果您不想设置 CloudWatch 日志记录角色，请选择选择现有角色，但不要选择日志记录角色。

b. 对于加密算法选项，请选择包含允许服务器使用的加密算法的安全策略。

> **Note**
>
> 默认情况下，除非选择不同的服务器，否则`TransferSecurityPolicy-2020-06`安全策略将连接到服务器。

有关安全策略的更多信息，请参阅AWS Transfer Family 服务器的安全策略。

c. （可选）对于 服务器主机密钥，输入 RSA、ED25519 或 ECDSA 私有密钥，该私有密钥将用于在客户端通过 SFTP 连接到服务器时识别服务器。

> **Note**
>
> 该部分仅适用于从启用 SFTP 的现有服务器迁移用户。

d. （可选）对于标签，在密钥和值中，输入一个或多个标签作为键值对，然后选择添加标签。

e. 请选择 Next（下一步）。

7. 在审核和创建页面上，审核您的选择。如果您：

   - 要编辑其中任何一个，请选择该步骤旁边的编辑。

   > **Note**
   >
   > 在选择编辑的步骤之后，您将需要查看每个步骤。

   - 如果没有更改，请选择创建服务器来创建您的服务器。您将转至如下所示的 Servers (服务器) 页面，其中列出了您的新服务器。

您的新服务器状态更改为在线可能需要几分钟时间。到时候，您的服务器可以执行用户的文件操作。

为服务器创建面向互联网的端点

在以下过程中，创建服务器端点。只有在您的 VPC 默认安全组中允许其源 IP 地址的客户端才能通过互联网访问此端点。此外，通过使用弹性 IP 地址使您的端点面向互联网，您的客户可以使用弹性 IP 地址来允许在其防火墙中访问您的端点。

> **Note**
>
> 在面向互联网的 VPC 托管端点上，只能使用 SFTP 和 FTPS。

创建面向互联网的端点

1. 打开AWS Transfer Family控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 从导航窗格中，选择服务器，然后选择创建服务器。
3. 在选择协议中，选择一个或多个协议，然后选择下一步。有关协议的更多信息，请参阅 步骤 2：创建启用 SFTP 的服务器。

4. 在选择身份提供商中，选择服务托管以在 AWS Transfer Family 中存储用户身份和密钥，然后选择下一步。

> **Note**
>
> 此过程使用服务托管选项。如果您选择自定义，则提供 Amazon API Gateway 端点和 AWS Identity and Access ManagementIAM 角色来访问端点。执行此操作后，您可以集成目录服务，用于对用户进行身份验证和授权。要了解有关使用自定义身份提供商的更多信息，请参阅使用自定义身份提供程序。

5. 在选择端点中，执行以下操作：

   a. 对于端点类型，选择托管服务器端点的 VPC 托管端点类型。

   b. 对于访问，请选择面向互联网，使客户端可以通过互联网访问您的端点。

   > **Note**
   >
   > 选择面向互联网时，可以在每个子网或多个子网中选择一个现有的弹性 IP 地址。或者您可以前往 VPC 控制台 (https://console.aws.amazon.com/vpc/) 分配一个或多个新的弹性 IP 地址。这些地址可以归AWS所有，也可以归您所有。您无法将已在使用的弹性 IP 地址与您的端点相关联。

   c. （可选）对于自定义主机名，请选择以下选项之一：

   > **Note**
   >
   > AWS GovCloud (US)需要直接通过弹性 IP 地址进行连接的客户，或者在商用 Route 53 中创建指向其 EIP 的主机名记录。有关将 Route 53 用于 GovCloud 终端节点的更多信息，请参阅AWS GovCloud (US)用户指南中的使用您的AWS GovCloud (US)资源设置 Amazon Route 53。

   - Amazon Route 53 DNS 别名— 如果要使用的主机名已注册到 Route 53。然后，您可以输入主机名。
   - 其他 DNS— 如果要使用的主机名已注册到另一个 DNS 提供商。然后，您可以输入主机名。

- 无 — 使用服务器的端点，而不是使用自定义主机名。服务器主机名使用格式 *server-id*.server.transfer.*region*.amazonaws.com。

> **Note**
>
> 对于中的客户AWS GovCloud (US)，选择 “无” 不会以这种格式创建主机名。

要了解有关使用自定义主机名的更多信息，请参阅使用自定义主机名。

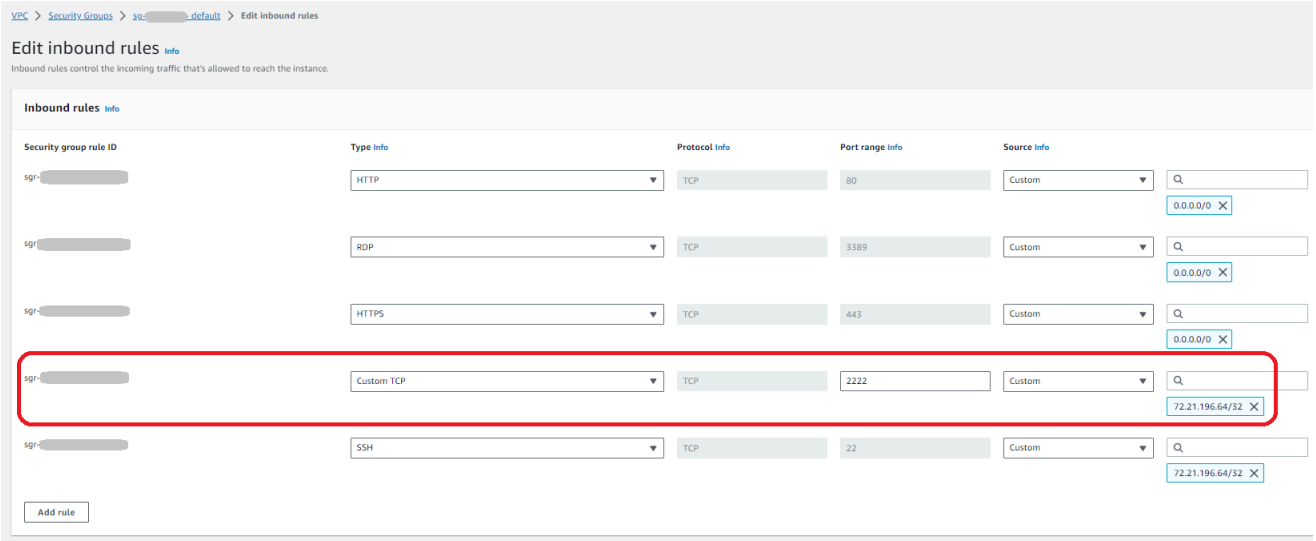
d. 对于 VPC，选择现有 VPC ID 或选择创建 VPC 以创建新的 VPC。

e. 在可用区部分，最多选择三个可用区和关联的子网。对于 IPv4 地址，为每个子网选择一个弹性 IP 地址。这是您的客户端可用来允许在其防火墙中访问您的端点的 IP 地址。

f. 在安全组部分，选择一个或多个现有安全组 ID 或选择创建安全组来创建新的安全组。有关安全组的更多信息，请参阅 Amazon Virtual Private Cloud 用户指南中 VPC 的安全组。要创建安全组，请参阅 Amazon Virtual Private Cloud 用户指南中的创建安全组。

> **Note**
>
> 您的 VPC 会自动带有默认的安全组。如果您在启动服务器时没有指定其他安全组或组，我们会将默认安全组与您的服务器相关联。

对于安全组的入站规则，您可以将 SSH 流量配置为使用端口 22、2222 或两者兼而有之。默认配置端口 22。要使用端口 2222，您需要向安全组添加入站规则。对于类型，选择 “自定义 TCP”，然后在 **2222** “端口范围” 中输入；对于源，输入与 SSH 端口 22 规则相同的 CIDR 范围。



g. （可选）对于启用 FIPS，请选中启用 FIPS 的端点复选框以确保端点符合联邦信息处理标准 (FIPS)。

> **Note**
>
> 启用 FIPS 的端点仅在北美AWS地区可用。有关可用区域，请参阅AWS 一般参考中的AWS Transfer Family端点和限额。有关 FIPS 的更多信息，请参阅联邦信息处理标准 (FIPS) 140-2。

h. 请选择 Next（下一步）。

6. 在配置其他详细信息中，执行以下操作：

a. 要进行CloudWatch 日志记录，请选择以下选项之一以启用 Amazon CloudWatch 记录您的用户活动：

- 创建一个新角色，允许 Transfer Family 自动创建 IAM 角色，前提是您拥有创建新角色的相应权限。创建的 IAM 角色被称为`AWSTransferLoggingAccess`。
- 选择现有角色以从您的帐户中选择现有 IAM 角色。在日志记录角色下，选择该角色。此 IAM 角色应包括将服务设置为`transfer.amazonaws.com`的信任策略。

  有关 CloudWatch 日志记录的更多信息，请参阅配置 CloudWatch 日志记录角色。

> **Note**
>
> - 如果您未指定日志记录角色， CloudWatch 则无法在中查看最终用户活动。

- 如果您不想设置 CloudWatch 日志记录角色，请选择选择现有角色，但不要选择日志记录角色。

b. 对于加密算法选项，请选择包含允许服务器使用的加密算法的安全策略。

> **Note**
>
> 默认情况下，除非选择不同的服务器，否则`TransferSecurityPolicy-2020-06`安全策略将连接到服务器。

有关安全策略的更多信息，请参阅AWS Transfer Family 服务器的安全策略。

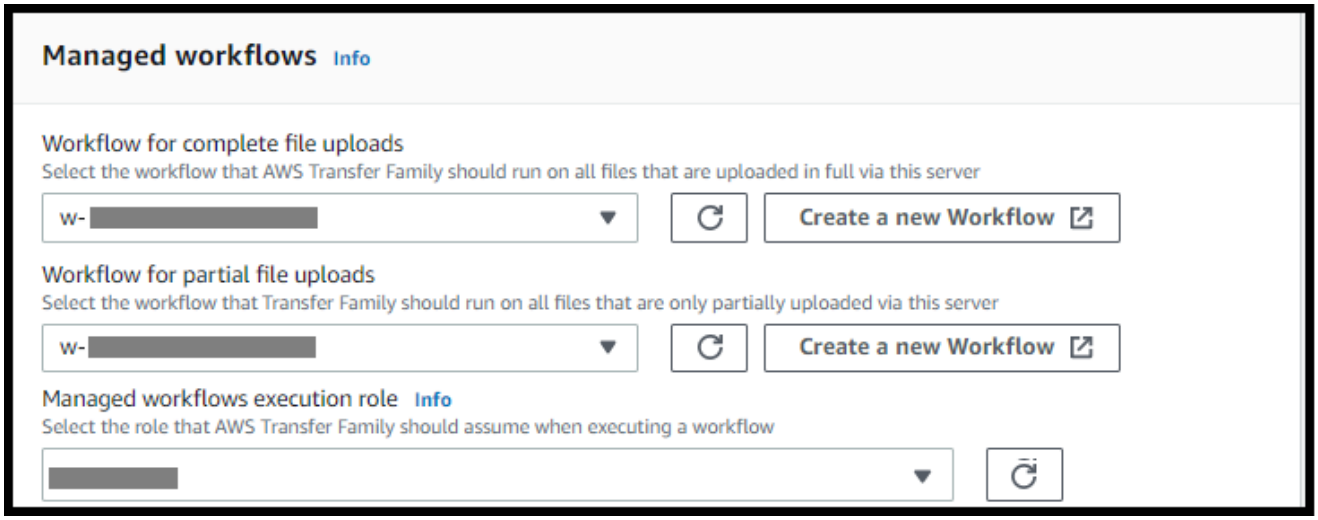
c. （可选）对于 服务器主机密钥，输入 RSA、ED25519 或 ECDSA 私有密钥，该私有密钥将用于在客户端通过 SFTP 连接到服务器时识别服务器。

> **Note**
>
> 该部分仅适用于从启用 SFTP 的现有服务器迁移用户。

d. （可选）对于标签，在密钥和值中，输入一个或多个标签作为键值对，然后选择添加标签。

e. 请选择 Next（下一步）。

f. （可选）对于托管工作流程，请选择 Transfer Family 在执行工作流程时应承担的工作流程 ID（和相应的角色）。您可以选择一个工作流程在完成上传后执行，选择另一个工作流程在部分上传时执行。要了解有关使用托管工作流程处理文件的更多信息，请参阅AWS Transfer Family 托管工作流程。



7. 在审核和创建页面上，审核您的选择。如果您：

   - 要编辑其中任何一个，请选择该步骤旁边的编辑。

     > **Note**
     >
     > 在选择编辑的步骤之后，您将需要查看每个步骤。

   - 如果没有更改，请选择创建服务器来创建您的服务器。您将转至如下所示的 Servers (服务器) 页面，其中列出了您的新服务器。

您可以选择服务器 ID，以查看您已创建的服务器的详细设置。填充公有 IPv4 地址列后，您提供的弹性 IP 地址将成功关联到服务器的端点。

> **Note**
>
> 当 VPC 中的服务器处于联机状态时，只能通过 UpdateServerAPI 修改子网。必须停止服务器才能添加或更改服务器端点的弹性 IP 地址。

更改 SFTP 服务器的端点类型

如果您现有的服务器可通过互联网访问（即，具有公有端点类型），您可以将其端点更改为 VPC 端点。

> **Note**
>
> 如果您在 VPC 中有一台显示为VPC_ENDPOINT的现有服务器，建议您将其修改为新的 VPC 端点类型。有了这种新的端点类型，您就不再需要使用网络负载均衡器 (NLB) 将弹性 IP 地址与服务器的端点关联起来。此外，您还可以使用 VPC 安全组来限制对服务器端点的访问。不过，您可以按需继续使用VPC_ENDPOINT端点类型。

以下过程假设您具有使用当前公有端点类型或较旧VPC_ENDPOINT类型的服务器。

更改服务器的端点类型

1. 打开AWS Transfer Family控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在导航窗格中，选择 Servers (服务器)。

3. 选中要更改其端点类型的服务器的复选框。

> **Important**
>
> 您必须先停止服务器，然后才能更改其终端节点。

4. 对于操作，请选择停止。
5. 在出现的确认对话框中，通过选择停止来确认您要停止服务器。

> **Note**
>
> 在继续下一步之前，在 端点详细信息 中，等待服务器的 状态 更改为 离线；这可能需要几分钟时间。您可能必须在服务器页面上选择刷新才能查看状态更改。
> 服务器离线之前，您无法进行任何编辑。

6. 在端点详细信息中，选择编辑。
7. 在编辑端点配置中，执行以下操作：

   a. 对于端点类型，选择 VPC 托管。

   b. 对于访问权限，请选择下列选项之一：

   - 内部，使您的端点只能由使用端点的私有 IP 地址的客户端访问。
   - 面向互联网，使客户端可以通过公共互联网访问您的端点。

   > **Note**
   >
   > 选择面向互联网时，可以在每个子网或多个子网中选择一个现有的弹性 IP 地址。或者，您可以前往 VPC 控制台 (https://console.aws.amazon.com/vpc/) 分配一个或多个新的弹性 IP 地址。这些地址可以归AWS所有，也可以归您所有。您无法将已在使用的弹性 IP 地址与您的端点相关联。

   c. （仅适用于面向互联网的访问权限是可选的）对于自定义主机名，请选择以下选项之一：

   - Amazon Route 53 DNS 别名— 如果要使用的主机名已注册到 Route 53。然后，您可以输入主机名。
   - 其他 DNS— 如果要使用的主机名已注册到另一个 DNS 提供商。然后，您可以输入主机名。

- 无 — 使用服务器的端点，而不是使用自定义主机名。服务器主机名使用格式 *serverId*.server.transfer.*regionId*.amazonaws.com。

  要了解有关使用自定义主机名的更多信息，请参阅使用自定义主机名。

d. 对于 VPC，选择现有 VPC ID 或选择创建 VPC 以创建新的 VPC。

e. 在可用区部分，最多选择三个可用区和关联的子网。如果选择面向互联网，则还要为每个子网选择一个弹性 IP 地址。

> **Note**
>
> 如果您想要最多三个可用区，但可用区域不足，则在 VPC 控制台中创建它们 (https://console.aws.amazon.com/vpc/)。
> 如果您修改子网或弹性 IP 地址，则服务器需要几分钟才能更新。在服务器更新完成之前，您无法保存更改。

f. 选择保存。

8. 在操作中，选择启动，然后等待服务器状态更改为在线；这可能需要几分钟。

> **Note**
>
> 如果您将公有端点类型更改为 VPC 端点类型，请注意您的服务器的端点类型已更改为 VPC。

默认安全组已附加到端点。要更改或添加其他安全组，请参阅创建安全组。

停止使用 VPC_ENDPOINT

AWS Transfer Family将停止使用EndpointType=VPC_ENDPOINT为新AWS帐户创建服务器的功能。截至 2021 年 5 月 19 日，不拥有端点类型为AWS的AWS Transfer Family服务器的VPC_ENDPOINT账户将无法使用EndpointType=VPC_ENDPOINT创建新服务器。如果您已经拥有使用该VPC_ENDPOINT端点类型的服务器，建议您EndpointType=VPC尽快开始使用。有关详细信息，请参阅将您的AWS Transfer Family服务器端点类型从 VPC_ENDPOINT 更新为 VPC。

我们在 2020 年初推出了新的VPC端点类型。有关更多信息，AWS Transfer Family请参阅 SFTP 支持 VPC 安全组和弹性 IP 地址。这个新的端点功能更丰富，更具成本效益，而且不 PrivateLink 收费。有关更多信息，请参阅AWS PrivateLink 定价。

此端点类型在功能上等同于以前的端点类型 (VPC_ENDPOINT)。您可以将弹性 IP 地址直接附加到端点，使其面向互联网，并使用安全组进行源 IP 筛选。有关更多信息，请参阅使用 IP 允许列表来保护您的 SFTP 服务器AWS Transfer Family安全博客文章。

您也可以在共享 VPC 环境中托管此端点。有关更多信息，请参阅AWS Transfer Family现在支持共享服务 VPC 环境。

除了 SFTP 之外，您还可以使用 VPC EndpointType来启用 FTPS 和 FTP。我们不打算将这些功能和 FTPS/FTP 支持添加到EndpointType=VPC_ENDPOINT。我们还从AWS Transfer Family控制台中删除了此端点类型选项。

您可以使用 Transfer Family 控制台、AWS CLI、API、开发工具包或AWS CloudFormation更改服务器的端点类型。要更改服务器的端点类型，请参阅将AWS Transfer Family服务器端点类型从 VPC_ENDPOINT 更新为 VPC。

如有任何疑问，请联系AWS Support或您的AWS客户团队。

> **Note**
>
> 我们不打算在 EndpointType =VPC_ENDPOINT 中添加这些功能以及 FTPS 或 FTP 支持。我们不再在AWS Transfer Family控制台上将其作为选项提供。

如果您还有其他问题，可以通过AWS Support或您的客户团队联系我们。

将AWS Transfer Family服务器端点类型从 VPC_ENDPOINT 更新为 VPC

您可以使用AWS Management Console、AWS CloudFormation、或 Transfer Family API 将服务器EndpointType从VPC_ENDPOINT更新为VPC。以下各部分提供了使用每种方法更新服务器端点类型的详细过程和示例。如果您在多个AWS区域和多个AWS账户中拥有服务器，则可以使用下一部分中提供的示例脚本进行修改，使用需要更新的VPC_ENDPOINT类型来识别服务器。

主题

使用VPC_ENDPOINT端点类型识别服务器

您可以使用VPC_ENDPOINT来识别哪些服务器正在使用AWS Management Console。

识别通过控制台使用VPC_ENDPOINT端点类型的服务器

1. 打开AWS Transfer Family控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在导航窗格中选择服务器，显示该区域中您账户中的服务器列表。
3. 按端点类型对服务器列表进行排序，以查看所有使用VPC_ENDPOINT的服务器。

识别跨多个VPC_ENDPOINT区域和账户使用AWS的服务器

如果您在多个AWS区域和多个AWS帐户中拥有服务器，则可以使用以下示例脚本进行修改，以使用VPC_ENDPOINT端点类型标识服务器。该示例脚本使用 Amazon EC2 DescribeRegions和 Transfer Family ListServers API 调用来获取所有使用服务器的服务器 ID 和区域的列表VPC_ENDPOINT。如果您有许多AWS账户，如果您使用身份提供商的会话配置文件进行身份验证，则可以使用具有只读审计员访问权限的 IAM 角色遍历您的账户。

1. 以下是一个简单示例。

```
import boto3

profile = input("Enter the name of the AWS account you'll be working in: ")
session = boto3.Session(profile_name=profile)

ec2 = session.client("ec2")

regions = ec2.describe_regions()

for region in regions['Regions']:
    region_name = region['RegionName']
    if region_name=='ap-northeast-3': #https://github.com/boto/boto3/issues/1943
        continue
    transfer = session.client("transfer", region_name=region_name)
    servers = transfer.list_servers()
    for server in servers['Servers']:
       if server['EndpointType']=='VPC_ENDPOINT':
           print(server['ServerId'], region_name)
```

2. 获得要更新的服务器列表后，您可以使用以下各部分中描述的方法之一将EndpointType更新为VPC。

使用AWS Management Console更新服务端点类型

1. 打开AWS Transfer Family控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在导航窗格中，选择 Servers (服务器)。
3. 选中要更改其端点类型的服务器的复选框。

> **Important**
>
> 您必须先停止服务器，然后才能更改其终端节点。

4. 对于操作，请选择停止。
5. 在出现的确认对话框中，通过选择停止来确认您要停止服务器。

> **Note**
>
> 在继续下一步之前，请等待服务器的状态变为离线；这可能需要几分钟。您可能必须在服务器页面上选择刷新才能查看状态更改。

6. 状态更改为离线后，选择服务器以显示服务器详细信息页面。
7. 在端点详细信息部分中，选择编辑。
8. 为端点类型选择 VPC 托管。
9. 选择保存
10. 在操作中，选择启动，然后等待服务器状态更改为在线；这可能需要几分钟。

使用AWS CloudFormation更新服务器端点类型

本部分介绍如何使用AWS CloudFormation将服务器EndpointType更新为VPC。对于已使用AWS CloudFormation部署的 Transfer Family 服务器，请使用以下步骤。在此示例中，用于部署 Transfer Family 服务器的原始AWS CloudFormation模板如下所示：

```
AWSTemplateFormatVersion: '2010-09-09'
Description: 'Create AWS Transfer Server with VPC_ENDPOINT endpoint type'
Parameters:
  SecurityGroupId:
    Type: AWS::EC2::SecurityGroup::Id
  SubnetIds:
    Type: List<AWS::EC2::Subnet::Id>
  VpcId:
```

```
    Type: AWS::EC2::VPC::Id
Resources:
  TransferServer:
    Type: AWS::Transfer::Server
    Properties:
      Domain: S3
      EndpointDetails:
        VpcEndpointId: !Ref VPCEndpoint
      EndpointType: VPC_ENDPOINT
      IdentityProviderType: SERVICE_MANAGED
      Protocols:
        - SFTP
  VPCEndpoint:
    Type: AWS::EC2::VPCEndpoint
    Properties:
      ServiceName: com.amazonaws.us-east-1.transfer.server
      SecurityGroupIds:
        - !Ref SecurityGroupId
      SubnetIds:
        - !Select [0, !Ref SubnetIds]
        - !Select [1, !Ref SubnetIds]
        - !Select [2, !Ref SubnetIds]
      VpcEndpointType: Interface
      VpcId: !Ref VpcId
```

模板已更新，其中包含以下更改：

- EndpointType已更改为VPC。
- AWS::EC2::VPCEndpoint资源已删除。
- SecurityGroupId、SubnetIds、和VpcId已移至EndpointDetails资源AWS::Transfer::Server部分，
- VpcEndpointId的EndpointDetails属性已删除。

更新后的模板如下所示：

```
AWSTemplateFormatVersion: '2010-09-09'
Description: 'Create AWS Transfer Server with VPC endpoint type'
Parameters:
  SecurityGroupId:
    Type: AWS::EC2::SecurityGroup::Id
  SubnetIds:
```

```
    Type: List<AWS::EC2::Subnet::Id>
  VpcId:
    Type: AWS::EC2::VPC::Id
Resources:
  TransferServer:
    Type: AWS::Transfer::Server
    Properties:
      Domain: S3
      EndpointDetails:
        SecurityGroupIds:
          - !Ref SecurityGroupId
        SubnetIds:
          - !Select [0, !Ref SubnetIds]
          - !Select [1, !Ref SubnetIds]
          - !Select [2, !Ref SubnetIds]
        VpcId: !Ref VpcId
      EndpointType: VPC
      IdentityProviderType: SERVICE_MANAGED
      Protocols:
        - SFTP
```

AWS CloudFormation要更新使用部署的 Transfer Family 服务器的端点类型

1. 使用以下步骤停止要更新的服务器。

   a. 打开AWS Transfer Family控制台，网址为 https://console.aws.amazon.com/transfer/。

   b. 在导航窗格中，选择 Servers (服务器)。

   c. 选中要更改其端点类型的服务器的复选框。

   > **Important**
   >
   > 您必须先停止服务器，然后才能更改其终端节点。

   d. 对于操作，请选择停止。

   e. 在出现的确认对话框中，通过选择停止来确认您要停止服务器。

   > **Note**
   >
   > 在继续下一步之前，请等待服务器的状态变为离线；这可能需要几分钟。您可能必须在服务器页面上选择刷新才能查看状态更改。

2. 更新堆 CloudFormation 栈

   a. 打开 AWS CloudFormation 控制台，地址：https://console.aws.amazon.com/cloudformation。

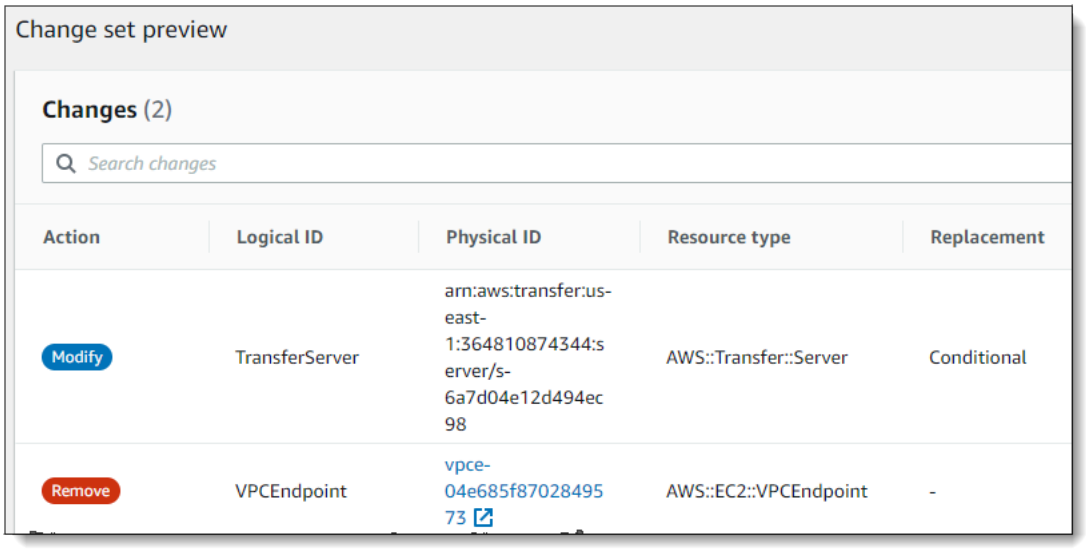
   b. 选择用于创建 Transfer Family 服务器的堆栈。

   c. 选择更新。

   d. 选择替换当前模板

   e. 上传新模板。 CloudFormation 更改集可帮助您在实施模板更改之前了解模板更改将如何影响正在运行的资源。在此示例中，将修改 Transfer 服务器资源，并删除 VPCEndpoint 资源。VPC 端点类型服务器代表您创建 VPC 端点，替换原始`VPCEndpoint`资源。

      上传新模板后，更改集将与以下所示类似：



   f. 更新堆栈。

3. 堆栈更新完成后，导航至 Transfer Family 管理控制台 https://console.aws.amazon.com/transfer/。

4. 重新启动服务器。选择在AWS CloudFormation中更新的服务器，然后从操作菜单中选择启动。

EndpointType 使用 API 更新服务器

您可以使用 describe-server AWS CLI命令或 UpdateServerAPI 命令。以下示例脚本停止 Transfer Family 服务器、更新 EndpointType、移除 VPC_ENDPOINT 并启动服务器。

```
import boto3
import time
```

```
profile = input("Enter the name of the AWS account you'll be working in: ")
region_name = input("Enter the AWS Region you're working in: ")
server_id = input("Enter the AWS Transfer Server Id: ")

session = boto3.Session(profile_name=profile)

ec2 = session.client("ec2", region_name=region_name)
transfer = session.client("transfer", region_name=region_name)

group_ids=[]

transfer_description = transfer.describe_server(ServerId=server_id)
if transfer_description['Server']['EndpointType']=='VPC_ENDPOINT':
    transfer_vpc_endpoint = transfer_description['Server']['EndpointDetails']
['VpcEndpointId']
    transfer_vpc_endpoint_descriptions =
 ec2.describe_vpc_endpoints(VpcEndpointIds=[transfer_vpc_endpoint])
    for transfer_vpc_endpoint_description in
 transfer_vpc_endpoint_descriptions['VpcEndpoints']:
        subnet_ids=transfer_vpc_endpoint_description['SubnetIds']
        group_id_list=transfer_vpc_endpoint_description['Groups']
        vpc_id=transfer_vpc_endpoint_description['VpcId']
        for group_id in group_id_list:
             group_ids.append(group_id['GroupId'])
    if transfer_description['Server']['State']=='ONLINE':
        transfer_stop = transfer.stop_server(ServerId=server_id)
        print(transfer_stop)
        time.sleep(300) #safe
        transfer_update =
 transfer.update_server(ServerId=server_id,EndpointType='VPC',EndpointDetails={'SecurityGroupId
        print(transfer_update)
        time.sleep(10)
        transfer_start = transfer.start_server(ServerId=server_id)
        print(transfer_start)
        delete_vpc_endpoint =
 ec2.delete_vpc_endpoints(VpcEndpointIds=[transfer_vpc_endpoint])
```

使用自定义主机名

服务器主机名是用户连接到服务器时在客户端中输入的主机名。在您使用 AWS Transfer Family 时，可以使用已为服务器主机名注册的自定义域。例如，您可以使用类似于 `mysftpserver.mysubdomain.domain.com` 的自定义主机名。

要将流量从注册的自定义域重定向到您的服务器端点，您可以使用 Amazon Route 53 或任何域名系统 (DNS) 提供商。Route 53 是AWS Transfer Family原生支持的 DNS 服务。

主题

在控制台上，您可以选择下列选项之一来设置自定义主机名：

- Amazon Route 53 DNS 别名— 如果要使用的主机名已注册到 Route 53。然后，您可以输入主机名。
- 其他 DNS— 如果要使用的主机名已注册到另一个 DNS 提供商。然后，您可以输入主机名。
- 无 — 使用服务器的端点，而不是使用自定义主机名。

在创建新的服务器或编辑现有服务器的配置时设置此选项。有关创建新的服务器的更多信息，请参阅步骤 2：创建启用 SFTP 的服务器。有关编辑现有服务器配置的更多信息，请参阅编辑服务器详细信息。

有关将您自己的域用于服务器主机名以及 AWS Transfer Family 如何使用 Route 53 的更多详细信息，请参阅以下部分。

使用 Amazon Route 53 作为 DNS 提供商

当您创建服务器时，您可以使用 Amazon Route 53 作为您的 DNS 提供程序。在将一个域用于 Route 53 之前，请先注册该域。有关更多信息，请参阅 Amazon Route 53 开发人员指南中的域注册工作原理。

在您使用 Route 53 向服务器提供 DNS 路由时，AWS Transfer Family 将使用您输入的自定义主机名来提取其托管区域。当 AWS Transfer Family 提取托管区域时，可能会发生以下三种情况：

1. 如果您刚开始使用 Route 53 并且没有托管区域，AWS Transfer Family 会添加一个新的托管区域和一条 CNAME 记录。此 CNAME 记录的值为服务器的端点主机名。CNAME 为备用域名。
2. 如果您在 Route 53 中有一个不带任何 CNAME 记录的托管区域，则 AWS Transfer Family 会向该托管区域添加一条 CNAME 记录。
3. 如果服务检测到该托管区域中已有一条 CNAME 记录，则会显示一个错误，指示 CNAME 记录已存在。在此情况下，请将 CNAME 记录的值更改为服务器的主机名。

> **Note**
>
> 如果此步骤是服务器创建工作流程的一部分，则表示您的服务器已成功创建，并且您的自定义主机名设置为无。

有关 Route 53 中托管区域的更多信息，请参阅 Amazon Route 53 开发人员指南中的托管区域。

使用其他 DNS 提供商

在创建服务器时，您还可以使用 Amazon Route 53 之外的 DNS 提供商。如果您使用替代 DNS 提供商，请确保域中的流量被定向到服务器端点。

为此，请将域设置为服务器的端点主机名。端点主机名在控制台中如下所示：

serverid.server.transfer.*region*.amazonaws.com

> **Note**
>
> 如果您的服务器有 VPC 端点，则主机名的格式与上述格式不同。要查找您的 VPC 端点，请在服务器的详细信息页面上选择 VPC，然后在 VPC 控制面板上选择 VPC 端点 ID。端点是列出的第一个 DNS 名称。

非控制台创建的服务器的自定义主机名

使用AWS Cloud Development Kit (AWS CDK)、AWS CloudFormation 或 CLI 创建服务器时，如果您希望该服务器具有自定义主机名，则必须添加标记。当您使用控制台创建 Transfer Family 服务器时，会自动完成标记。

> **Note**
>
> 您还需要创建 DNS 记录，以将流量从您的域名重定向到服务器端点。有关详细信息，请参阅《Amazon Route 53 开发者指南》中的处理记录。

使用以下密钥作为您的自定义主机名：

- 添加`transfer:customHostname`以在控制台中显示自定义主机名。

- 如果您使用 Route 53 作为 DNS 提供商，请添加`transfer:route53HostedZoneId`。此标签将自定义主机名链接到您的 Route 53 托管区 ID。

要添加自定义主机名，请发出以下 CLI 命令。

```
aws transfer tag-resource --arn arn:aws:transfer:region:AWS ##:server/server-ID --tags
 Key=transfer:customHostname,Value="custom-host-name"
```

例如：

```
aws transfer tag-resource --arn arn:aws:transfer:us-east-1:111122223333:server/
s-1234567890abcdef0 --tags Key=transfer:customHostname,Value="abc.example.com"
```

如果您使用的是 Route 53，请发出以下命令将您的自定义主机名链接到您的 Route 53 托管区 ID。

```
aws transfer tag-resource --arn server-ARN:server/server-ID --tags
 Key=transfer:route53HostedZoneId,Value=HOSTED-ZONE-ID
```

例如：

```
aws transfer tag-resource --arn arn:aws:transfer:us-east-1:111122223333:server/
s-1234567890abcdef0 --tags Key=transfer:route53HostedZoneId,Value=ABCDE1111222233334444
```

假设上一个命令中的示例值，运行以下命令来查看标签：

```
aws transfer list-tags-for-resource --arn arn:aws:transfer:us-
east-1:111122223333:server/s-1234567890abcdef0
```

```
"Tags": [
   {
      "Key": "transfer:route53HostedZoneId",
      "Value": "/hostedzone/ABCDE1111222233334444"
   },
   {
      "Key": "transfer:customHostname",
      "Value": "abc.example.com"
   }
 ]
```

> **Note**
>
> 您的公共、托管区域及其 ID 可在 Amazon Route 53 上获取。
> 登录到 AWS Management Console，然后通过以下网址打开 Route 53 控制台：https://console.aws.amazon.com/route53/。

使用客户端通过服务器端点传输文件

通过在客户端中指定传输操作，您可以通过 AWS Transfer Family 服务传输文件。 AWS Transfer Family 支持以下客户端：

- 我们支持 SFTP 协议的第 3 版。
- OpenSSH (macOS 和 Linux)

> **Note**
>
> 此客户端仅适用于启用了 Secure Shell (SSH) 文件传输协议 (SFTP) 的服务器。

- WinSCP（仅 Microsoft Windows）
- Cyberduck（Windows、macOS 和 Linux）
- FileZilla （Windows、macOS 和 Linux）

以下限制适用于每个客户端：

- 每个连接的并发、多路复用、SFTP 会话的最大数量为 10。
- Amazon S3 和 Amazon EFS（由于 NFSv4 协议）要求文件名采用 UTF-8 编码。使用不同的编码可能会导致意想不到的结果。对于 Amazon S3，请参阅对象密钥命名指南。
- 对于 安全文件传输协议 (FTPS)，仅支持显式模式。不支持隐式模式。
- 对于文件传输协议 (FTP) 和 FTPS，仅支持被动模式。
- 对于 FTP 和 FTPS，仅支持流模式。
- 对于 FTP 和 FTPS，仅支持图像/二进制模式。
- 对于 FTP 和 FTPS，数据连接的 TLS-PROT C（未受保护）TLS 是默认值，但是 AWS Transfer Family FTPS 协议不支持端口 C。因此，对于 FTPS，您需要发出 PROT P，您的数据操作才能被接受。

- 如果您使用 Amazon S3 作为服务器存储，并且您的客户端包含使用多个连接进行单次传输的选项，请务必禁用该选项。否则，上传大文件可能会突然失败。请注意，如果您使用 Amazon EFS 作为存储后端，EFS 确实支持多个连接进行单次传输。

以下是 FTP 和 FTPS 的可用命令列表：

可用命令					
ABOR	FEAT	MLST	PASS	RETR	STOR
AUTH	LANG	MKD	PASV	RMD	STOU
CDUP	LIST	MODE	PBSZ	RNFR	STRU
CWD	MDTM	NLST	PROT	RNTO	SYST
DELE	MFMT	NOOP	PWD	SIZE	TYPE
EPSV	MLSD	OPTS	QUIT	STAT	USER

> **Note**
>
> 不支持 APPE。

对于 SFTP，目前不支持在使用 Amazon Elastic File System (Amazon EFS) 的服务器上使用逻辑主目录的用户执行以下操作。

SFTP 命令不受支持			
SSH_FXP_READLINK	SSH_FXP_SYMLINK	SSH_FXP_STAT（当请求的文件是符号链接时）	SSH_FXP_REALPATH（当请求的路径包含任何符号链接组件时）

生成公有-私有密钥对

在传输文件之前，必须有可用的公有-私有密钥对。如果您之前没有生成过密钥对，请参阅为服务托管用户生成 SSH 密钥。

主题

可用的 SFTP/FTPS/FTP 命令

下表描述了 SFTP AWS Transfer Family、FTPS 和 FTP 协议的可用命令。

> **Note**
>
> 该表提到了仅支持存储桶和对象的 Amazon S3 的文件和目录：无层次结构。但是，您可以在对象键名称中使用前缀来暗示层次结构，并以类似于文件夹的方式组织数据。Amazon Simple Storage Service 用户指南中的使用对象元数据中描述了该行为。

SFTP/FTPS/FTP 命令

命令	Amazon S3	Amazon EFS
`cd`	支持	支持
`chgrp`	不支持	支持（`root`或仅支持`owner`）
`chmod`	不支持	支持（`root`仅限）
`chmtime`	不支持	支持

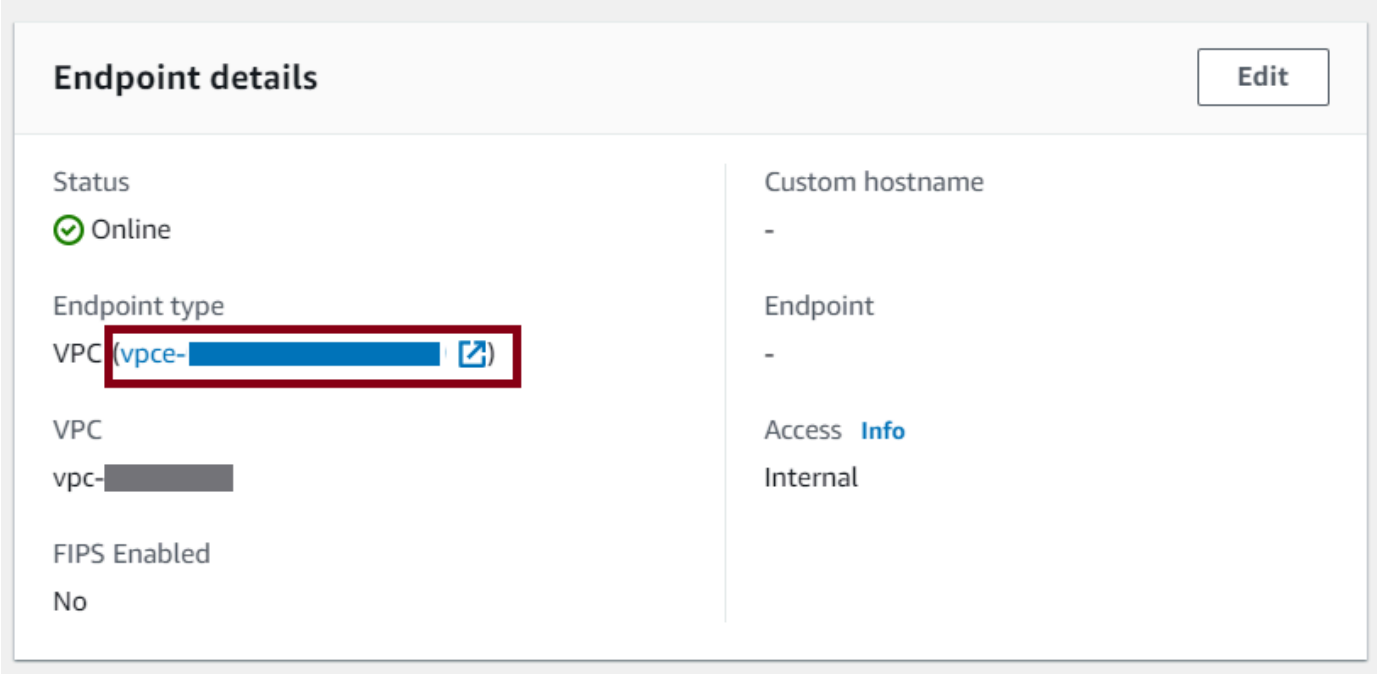
命令	Amazon S3	Amazon EFS
chown	不支持	支持（root仅限）
get	支持	支持（包括解析符号链接）
ln -s	不支持	支持
ls/dir	支持	支持
mkdir	支持	支持
put	支持	支持
pwd	支持	支持
rename	仅支持文件	支持 **Note** 不支持会覆盖现有文件或目录的重命名。
rm	支持	支持
rmdir	支持（仅限空目录）	支持
version	支持	支持

查找您的 Amazon VPC 端点

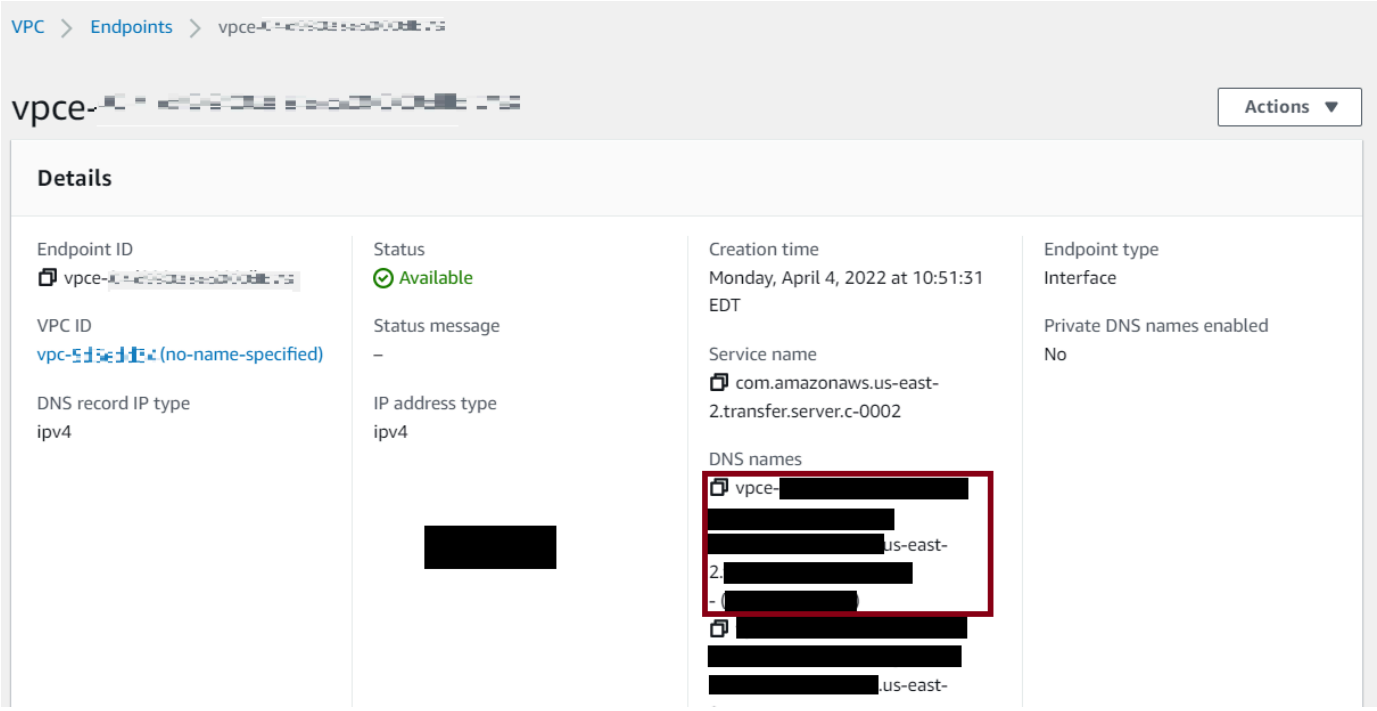
如果您的 Transfer Family 服务器的端点类型是 VPC，则识别用于传输文件的端点并不简单。在这种情况下，使用以下过程查找您的 Amazon VPC 端点。

查找您的 Amazon VPC 端点

1. 导航到您的服务器详细信息页面。
2. 在端点详细信息窗格中，选择 VPC。



3. 在 Amazon VPC 控制面板中，选择 VPC 端点 ID。
4. 在 DNS 名称列表中，您的服务器端点是第一个列出的端点。



避免setstat错误

一些 SFTP 文件传输客户端可以在上传文件时尝试使用命令（例如 SETSTAT）更改远程文件的属性，包括时间戳和权限。但是，这些命令与 Amazon S3 等对象存储系统不兼容。由于这种不兼容性，即使文件以其他方式成功上传，从这些客户端上传文件也可能导致错误。

- 当您调用CreateServer或 UpdateServerAPI 时，使用该ProtocolDetails选项SetStatOption可以忽略当客户端尝试对要上传到 S3 存储桶的文件使用 SETSTAT 时生成的错误。
- 将该值设置为 ENABLE_NO_OP 以使 Transfer Family 服务器忽略 SETSTAT 命令，并上传文件而无需对您的 SFTP 客户端进行任何更改。
- 请注意，虽然该SetStatOptionENABLE_NO_OP设置忽略了错误，但它确实会在日志中 CloudWatch 生成一个日志条目，因此您可以确定客户端何时进行 SETSTAT 调用。

有关此选项的 API 详细信息，请参阅ProtocolDetails。

使用 OpenSSH

按照下文中的说明，使用 OpenSSH 从命令行传输文件。

> **Note**
>
> 此客户端仅适用于启用 SFTP 的服务器。

AWS Transfer Family 使用 OpenSSH 命令行实用程序传输文件

1. 在 Linux、macOS 或 Windows 上，打开命令终端。
2. 在提示符中，输入以下命令：

   ```
   sftp -i transfer-key sftp_user@service_endpoint
   ```

   在前面的命令中，*sftp_user* 是用户名，*transfer-key* 是 SSH 私有密钥。此处*service_endpoint*是服务器的终端节点，如所选服务器的 AWS Transfer Family 控制台中所示。

> **Note**
>
> 此命令使用默认ssh_config文件中的设置。除非您之前编辑过此文件，否则 SFTP 使用端口 22。您可以通过在命令中添加**-P**标志来指定其他端口（例如 2222），如下所示。
>
> ```
> sftp -P 2222 -i transfer-key sftp_user@service_endpoint
> ```
>
> 或者，如果您一直想使用端口 2222，则可以更新ssh_config文件中的默认端口。

此时应显示 sftp 提示符。

3. （可选）要查看用户的主目录，请在sftp提示符下输入以下命令：

   pwd

4. 要将文件从您的文件系统上传到 Transfer Family 服务器，请使用put命令。例如，要上传hello.txt（假设该文件位于文件系统的当前目录中），请在sftp提示符下运行以下命令：

   ```
   put hello.txt
   ```

   此时将显示类似于下文的消息，指示文件传输正在进行或者已完成。

   ```
   Uploading hello.txt to /my-bucket/home/sftp_user/hello.txt

   hello.txt 100% 127 0.1KB/s 00:00
   ```

> **Note**
>
> 在您的服务器创建之后，环境中的 DNS 服务可能需要几分钟时间才能解析服务器端点主机名。

使用 WinSCP

按照下文中的说明，使用 WinSCP 从命令行传输文件。

> **Note**
>
> 如果您使用的是 WinSCP 5.19，则可以使用 AWS 您的证书直接连接到 Amazon S3 并上传/下载文件。有关更多详细信息，请参阅连接到 Amazon S3 服务。

AWS Transfer Family 使用 WinSCP 传输文件

1. 打开 WinSCP 客户端。
2. 在登录对话框中，为文件协议选择一个协议：SFTP 或 FTP。

   如果您选择了 加密，请选择下列选项之一：

   - FTP 没有加密
   - 适用于 FTPS 的 TLS/SSL 显式加密
3. 对于 Host name (主机名)，输入您的服务器终端节点。服务器端点位于服务器详细信息页面。有关更多信息，请参阅 查看 SFTP、FTPS 和 FTP 服务器的详细信息。

   > **Note**
   >
   > 如果您的服务器使用 VPC 端点，请参阅查找您的 Amazon VPC 端点。
4. 在端口号中，输入以下内容：

   - **22**适用于 SFTP
   - **21**适用于 FTP/FTPS
5. 在用户名中，输入您为特定身份提供商创建的用户名称。

   > **Note**
   >
   > 用户名应是您为身份提供商创建或配置的用户之一。 AWS Transfer Family 提供以下身份提供商：
   >
   > - 与服务托管用户合作
   > - 使用 Di AWS rectory Service 身份提供商
   > - 使用自定义身份提供程序
6. 选择高级打开高级站点设置对话框。在 SSH 部分中，选择身份验证。

7. 对于私有密钥文件，从文件系统中浏览并选择 SSH 私有密钥文件。

> **Note**
>
> 如果 WinSCP 提供将 SSH 私有密钥转换为 PPK 格式，请选择确定。

8. 选择 OK (确定) 以返回到 Login (登录) 对话框，然后选择 Save (保存)。
9. 在将会话保存为站点对话框中，选择确定以完成您的连接设置。
10. 在登录对话框中，选择工具，然后选择首选项。
11. 在首选项对话框中的传输中，选择耐力。

    对于启用传输恢复/传输到临时文件名选项，选择禁用。

> **Note**
>
> 如果您启用此选项，则会增加上传成本，从而显著降低上传性能。它还可能导致大文件上传失败。

12. 对于传输，选择背景，然后清除使用多个连接进行单次传输复选框。

> **Note**
>
> 如果选择此选项，则上传大文件可能会以突然失败。例如，可以创建会产生 Amazon S3 费用的孤立分段上传。还可能发生静默数据损坏。

13. 执行文件传输。

    您可以使用 drag-and-drop 方法在目标窗口和源窗口之间复制文件。在 WinSCP 中，您可以使用工具栏图标来上传、下载、删除、编辑或修改文件的属性。

> **Note**
>
> 如果您使用 Amazon EFS 进行存储，则本说明不适用。
> 尝试更改远程文件属性（包括时间戳）的命令与 Amazon S3 等对象存储系统不兼容。因此，如果您使用 Amazon S3 进行存储，请务必在执行文件传输之前禁用 WinSCP 时间戳设置（或按`SetStatOption`中所述使用避免`setstat`错误）。为此，请在 WinSCP 传输设置对话框中，禁用设置权限上传选项和保留时间戳常用选项。

使用 Cyberduck

按照下文中的说明，使用 Cyberduck 从命令行传输文件。

AWS Transfer Family 使用 Cyberduck 传输文件

1. 打开 Cyberduck 客户端。
2. 选择打开连接。
3. 在打开连接对话框中，选择协议：SFTP（SSH 文件传输协议）、FTP-SSL（显式身份验证 TLS）或 FTP（文件传输协议）。
4. 对于服务器，输入您的服务器端点。服务器端点位于服务器详细信息页面。有关更多信息，请参阅查看 SFTP、FTPS 和 FTP 服务器的详细信息。

   > **Note**
   >
   > 如果您的服务器使用 VPC 端点，请参阅查找您的 Amazon VPC 端点。

5. 在端口号中，输入以下内容：
   - **22**适用于 SFTP
   - **21**适用于 FTP/FTPS
6. 对于 Username (用户名)，输入您在管理服务器端点的用户中创建的用户的名称。
7. 如果选择了 SFTP，则在 SSH 私有密钥中，选择或输入 SSH 私有密钥。
8. 选择连接。
9. 执行文件传输。

   根据您的文件所在的位置，执行以下操作之一：
   - 在您的本地目录（源）中，选择您要传输的文件，然后将这些文件拖放到 Amazon S3 目录（目标）中。
   - 在 Amazon S3 目录（源）中，选择您要传输的文件，然后将这些文件拖放到您的本地目录（目标）中。

使用 FileZilla

按照以下说明使用传输文件 FileZilla。

要设置 FileZilla 文件传输

1. 打开 FileZilla 客户端。
2. 选择文件，然后选择站点管理器。
3. 在站点管理器对话框中，选择新建站点。
4. 在常规选项卡的协议中选择一个协议：SFTP 或 FTP。

   如果您选择了 加密，请选择下列选项之一：

   - 仅使用纯 FTP（不安全）— 用于 FTP
   - 使用 TLS 上的显式 FTP（如果可用）— 用于 FTPS
5. 在主机名中，输入您正在使用的协议，然后输入您的服务器端点。服务器端点位于服务器详细信息页面。有关更多信息，请参阅 查看 SFTP、FTPS 和 FTP 服务器的详细信息。

   > **Note**
   >
   > 如果您的服务器使用 VPC 端点，请参阅查找您的 Amazon VPC 端点。

   - 如果您使用的是 SFTP，请输入：`sftp://`*`hostname`*
   - 如果您使用的是 FTPS，请输入：`ftps://`*`hostname`*

   请务必将*###*替换为实际的服务器端点。
6. 在端口号中，输入以下内容：

   - **22**适用于 SFTP
   - **21**适用于 FTP/FTPS
7. 如果选择了 SFTP，则选择密钥文件作为登录类型。

   对于密钥文件，选择或输入 SSH 私有密钥。
8. 对于用户名，输入您在管理服务器端点的用户中创建的用户的名称。
9. 选择连接。
10. 执行文件传输。

> **Note**
>
> 如果您中断正在进行的文件传输， AWS Transfer Family 可能会在您的 Amazon S3 存储桶中写入部分对象。如果您中断上传，在继续之前，请检查 Amazon S3 存储桶中文件大小是否与源对象的文件大小相符。

使用 Perl 客户端

如果您使用 NET::SFTP::Foreign perl 客户端，则必须将设置queue_size为。1例如：

```
my $sftp = Net::SFTP::Foreign->new('user@s-12345.server.transfer.us-east-2.amazonaws.com', queue_size => 1);
```

> **Note**
>
> 1.92.02 之前的Net::SFTP::Foreign修订版本需要使用此解决方法。

上传后处理

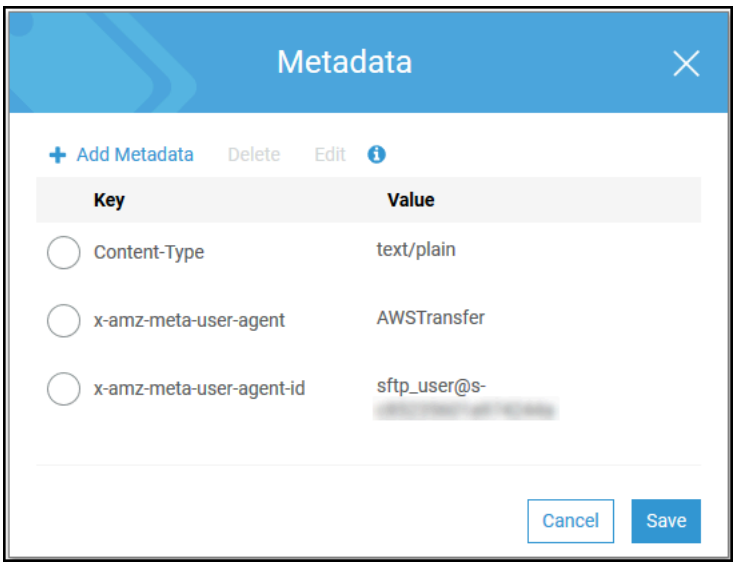
你可以查看上传后的处理信息，包括 Amazon S3 对象元数据和事件通知。

主题

- Amazon S3 对象元数据
- Amazon S3 事件通知

Amazon S3 对象元数据

作为对象元数据的一部分，您会看到一个名为x-amz-meta-user-agent的密钥，其值为AWSTransfer，x-amz-meta-user-agent-id的值为username@server-id。username是上传文件的 Transfer Family 用户，server-id也是用于上传的服务器。可以使用对 Lambda 函数中的 S3 对象进行HeadObject操作来访问这些信息。



Amazon S3 事件通知

当使用 Transfer Family 将对象上传到您的 S3 存储桶时，`RoleSessionName`作为`[AWS:Role Unique Identifier]/username.sessionid@server-id`包含在 S3 事件通知结构的请求者字段中。例如，以下是来自 S3 访问权限日志的、用于复制到 S3 存储桶中的请求者字段示例内容。

```
arn:aws:sts::AWS-Account-ID:assumed-role/IamRoleName/
username.sessionid@server-id
```

在上述中请求者字段中，它显示了名为`IamRoleName`的 IAM 角色。有关配置 S3 事件通知的更多信息，请参阅Amazon Simple Storage Service 开发人员指南中的配置 Amazon S3 事件通知。有关 AWS Identity and Access Management (IAM) 角色唯一标识符的更多信息，请参阅AWS Identity and Access Management 用户指南中的唯一标识符。

管理服务器端点的用户

在下列部分中，您可找到有关如何使用AWS Transfer Family，AWS Directory Service for Microsoft Active Directory或自定义身份提供商添加用户的信息。

如果您使用服务托管身份类型，则将用户添加到您启用文件传输协议的服务器。在执行此操作时，服务器上的各个用户名必须唯一。

作为各个用户属性的一部分，您还可以存储该用户的安全外壳 (SSH) 公有密钥。对于此过程使用的基于密钥的身份验证，必须这样操作。私有密钥存储在您用户的计算机本地。当用户使用客户端发送身份验证请求到服务器时，您的服务器首先确认用户具有关联 SSH 私有密钥的访问权限。然后，服务器成功验证用户身份。

此外，您指定用户的主目录或登录目录，并将 AWS Identity and Access ManagementIAM 角色分配给用户。或者，您可以提供一个会话策略来限制用户仅访问 AmazonS3 存储桶的主目录。

> **Important**
>
> AWS Transfer Family屏蔽长度为 1 或 2 个字符的用户名向 SFTP 服务器进行身份验证。此外，我们还屏蔽了`root`用户名。
> 其背后的原因是密码扫描器进行了大量的恶意登录尝试。

Amazon EFS 与 Amazon S3

每种存储选项的特点：

- 限制访问权限：Amazon S3 支持会话策略；Amazon EFS 支持 POSIX 用户、组和辅助组 ID
- 两者都支持公开密钥/私有密钥
- 两者都支持主目录
- 两者都支持逻辑目录

> **Note**
>
> 对于 Amazon S3，对逻辑目录的大部分支持是通过 API/CLI 实现的。您可以使用控制台中的受限复选框将用户锁定至其主目录，但不能指定虚拟目录结构。

逻辑目录

如果要为用户指定逻辑目录值，则使用的参数取决于用户的类型。

- 对于服务托管的用户，请在`HomeDirectoryMappings`中提供逻辑目录值。
- 对于自定义身份提供商用户，请在中提供逻辑目录值`HomeDirectoryDetails`。

主题

与服务托管用户合作

您可以将 Amazon S3 或 Amazon EFS 服务托管用户添加到您的服务器，具体取决于服务器的域设置。有关更多信息，请参阅 配置 SFTP、FTPS 或 FTP 服务器端点。

要以编程方式添加服务管理用户，请参阅 AP I 示例。CreateUser

> **Note**
>
> 对于服务管理用户，逻辑目录条目的限制为 2,000 个。有关使用逻辑目录的信息，请参见使用逻辑目录简化您的 Transfer Family 目录结构。

主题

- 添加 Amazon S3 服务托管用户
- 添加 Amazon EFS 服务托管用户
- 管理服务托管用户

添加 Amazon S3 服务托管用户

> **Note**
>
> 如果要配置跨账户 Amazon S3 存储桶，请按照知识中心文章中提到的步骤进行操作：如何将 AWS Transfer Family服务器配置为使用其他AWS账户中的 Amazon Simple Storage Service 存储桶？

将 Amazon S3 服务托管用户添加至您的服务器

1. 通过 https://console.aws.amazon.com/transfer/ 打开AWS Transfer Family控制台，然后从导航窗格选择服务器。
2. 在服务器页面上，选中您要将用户添加到的服务器复选框。
3. 选择添加用户。
4. 在用户配置部分的用户名中，输入用户名。此用户名长度最少为 3 个字符，最多为 100 个字符 您可以在用户名中使用以下字符：a–z、A-Z、0–9、下划线“_”、连字符“-”、句点“.”和“@”符号。用户名不能以连字符“-”、句点“.”或“@”符号开头。

5. 对于访问权限，选择您之前创建的提供对 Amazon S3 存储桶访问权限的 IAM 角色。

   您可使用创建 IAM 角色和策略中的过程创建此 IAM 角色。该 IAM 角色包括一个提供对您 Amazon S3 存储桶访问权限的 IAM policy。它还包括与 AWS Transfer Family 服务的信任关系，在另一个 IAM 策略中定义。如果您需要对用户进行精细的访问控制，请参阅使用AWS Transfer Family和 Amazon S3 博客文章增强数据访问控制。

6. （可选）对于策略，选择下列选项之一：

   - 无
   - 现有策略
   - 从 IAM 中选择策略：允许您选择现有的会话策略。选择查看以查看包含策略详细信息的 JSON 对象。
   - 基于主文件夹自动生成策略：为您生成会话策略。选择查看以查看包含策略详细信息的 JSON 对象。

   > **Note**
   >
   > 如果选择基于主文件夹自动生成策略，请不要为此用户选择受限。

   要了解有关会话策略的更多信息，请参阅创建 IAM 角色和策略。要了解有关创建会话策略的更多信息，请参阅为 Amazon S3 存储桶创建会话策略。

7. 对于主目录，选择 Amazon S3 存储桶用于存储使用 AWS Transfer Family 传输的数据。输入用户在使用其客户端登录时转到的 home 目录的路径。

   如果您将此参数留空，则使用 Amazon S3 存储桶的 `root` 目录。在这种情况下，请确保您的 IAM 角色提供对此 `root` 目录的访问权限。

   > **Note**
   >
   > 我们建议您选择包含用户的用户名的目录路径，这使得您可以更高效地使用会话策略。会话策略将用户在 Amazon S3 存储桶中的访问权限限制为该用户的 home 目录。

8. （可选）对于受限，选中该复选框，这样您的用户就无法访问该文件夹之外的任何内容，也看不到 Amazon S3 存储桶或文件夹名称。

> **Note**
>
> 为用户分配主目录并限制用户访问该主目录应该足以锁定用户对指定文件夹的访问权限。如果您需要应用进一步的控制措施，请使用会话策略。
> 如果您为此用户选择受限，则无法选择基于主文件夹自动生成策略，因为主文件夹不是为受限用户定义的值。

9. 对于SSH 公有密钥，输入 SSH 密钥对的 SSH 公有密钥部分。

   您的密钥先由服务进行验证，然后才能添加新用户。

   > **Note**
   >
   > 有关如何生成 SSH 密钥对的说明，请参阅为服务托管用户生成 SSH 密钥。

10. （可选）对于键和值，输入一个或多个标记作为键-值对，然后选择添加标记。
11. 选择 Add (添加) 可将您的新用户添加到所选服务器。

    新用户将出现在服务器详细信息页面的用户部分。

后续步骤 — 对于下一步，请继续前往使用客户端通过服务器端点传输文件。

添加 Amazon EFS 服务托管用户

Amazon EFS 使用便携式操作系统接口 (POSIX) 文件权限模型来表示文件所有权。

- 有关 Amazon EFS 文件所有权的更多详细信息，请参阅 Amazon EFS 文件所有权。
- 有关为 EFS 用户设置目录的更多详细信息，请参阅为 Transfer Family 设置 Amazon EFS 用户。

将 Amazon EFS 服务托管用户添加至您的服务器

1. 通过 https://console.aws.amazon.com/transfer/ 打开AWS Transfer Family控制台，然后从导航窗格选择服务器。
2. 在服务器页面上，选择要向其添加用户的 Amazon EFS 服务器。
3. 选择添加用户以显示添加用户页面。
4. 在用户配置部分中，使用以下设置。

a. 此用户名长度最少为 3 个字符，最多为 100 个字符。您可以在用户名中使用以下字符：a–z、A-Z、0–9、下划线“_”、连字符“-”、句点“.”和“@”符号。用户名不能以连字符“-”、句点“.”或“@”符号开头。

b. 对于用户 ID 和组 ID，请注意以下几点：

- 对于您创建的第一个用户，我们建议您为组 ID 和用户 ID 输入一个值。**0**这将授予用户使用 Amazon EFS 的管理员权限。
- 对于其他用户，请输入用户的 POSIX 用户 ID 和组 ID。这些 ID 用于用户执行的所有 Amazon Elastic File System 操作。
- 对于用户 ID 和组 ID，请勿使用任何前导零。例如，可以接受 **12345**，但不能接受 **012345**。

c. （可选）对于辅助组 ID，请为每个用户输入一个或多个其他 POSIX 组 ID，用逗号分隔。

d. 对于访问权限，请选择符合以下条件的 IAM 角色：

- 仅允许用户访问您希望他们访问的 Amazon EFS 资源（文件系统）。
- 定义用户可以执行哪些文件系统操作和不能执行哪些文件系统操作。

我们建议您使用具有挂载权限和读/写权限的 Amazon EFS 文件系统选择的 IAM 角色。例如，以下两个AWS托管策略的组合虽然相当宽松，但可以为您的用户授予必要的权限：

- AmazonElasticFileSystemClientFullAccess
- AWSTransferConsoleFullAccess

有关更多信息，请参阅 Amazon Elastic File System AWS Transfer Family 支持的博客文章。

e. 对于主目录，请执行以下操作：

- 选择您希望用于存储使用AWS Transfer Family传输的数据的 Amazon EFS 文件系统。
- 决定是否将主目录设置为受限。将主目录设置为受限会产生以下影响：
  - Amazon EFS 用户无法访问该文件夹之外的任何文件或目录。
  - Amazon EFS 用户看不到 Amazon EFS 文件系统名称 (fs-xxxxxxx)。

> **Note**
>
> 当您选择受限选项时，符号链接无法为 Amazon EFS 用户解析。

- （可选）输入您希望用户在使用客户端登录时进入的主目录路径。

  如果您未指定主目录，则使用您的 Amazon EFS 文件系统的根目录。在这种情况下，请确保您的 IAM 角色提供对此根目录的访问权限。

5. 对于SSH 公有密钥，输入 SSH 密钥对的 SSH 公有密钥部分。

   您的密钥先由服务进行验证，然后才能添加新用户。

   > **Note**
   >
   > 有关如何生成 SSH 密钥对的说明，请参阅为服务托管用户生成 SSH 密钥。

6. （可选）为用户输入任何标签。对于键和值，输入一个或多个标记作为键-值对，然后选择添加标记。
7. 选择 Add (添加) 可将您的新用户添加到所选服务器。

   新用户将出现在服务器详细信息页面的用户部分。

首次通过 SFTP 连接到 Transfer Family 服务器时可能会遇到的问题：

- 如果您运行`sftp`命令但提示符未出现，则可能会遇到以下消息：

  ```
  Couldn't canonicalize: Permission denied

  Need cwd
  ```

  在这种情况下，您必须增加用户角色的策略权限。您可以添加AWS托管策略，例如`AmazonElasticFileSystemClientFullAccess`。
- 如果您在`sftp`提示`pwd`时输入查看用户的主目录，则可能会看到以下消息，其中 *`USER-HOME-DIRECTORY`* 是 SFTP 用户的主目录：

  ```
  remote readdir("/USER-HOME-DIRECTORY"): No such file or directory
  ```

  在这种情况下，您应该能够导航到父目录 (`cd ..`)，并创建用户的主目录 (`mkdir` *`username`*)。

后续步骤 — 对于下一步，请继续前往使用客户端通过服务器端点传输文件。

管理服务托管用户

在本部分中，您可以找到有关如何查看用户列表、如何编辑用户详细信息以及如何添加 SSH 公有密钥的信息。

- 查看用户列表
- 查看或编辑用户详细信息
- 删除用户
- 添加 SSH 公钥
- 删除 SSH 公钥

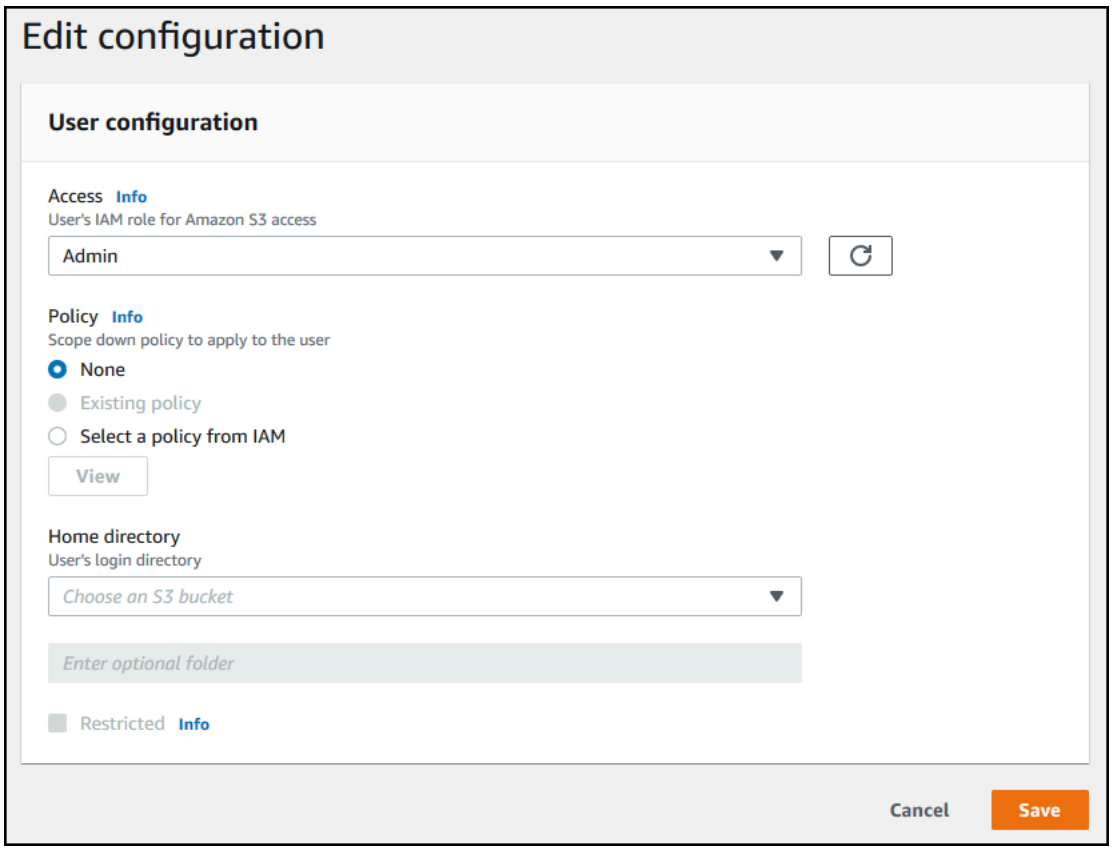
查找您的用户列表

1. 打开AWS Transfer Family控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 从导航窗格中选择服务器以显示服务器页面。
3. 选择服务器 ID 列中的标识符以查看服务器详细信息页面。
4. 在用户下，查看用户列表。

要查看或编辑用户详细信息

1. 打开AWS Transfer Family控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 从导航窗格中选择服务器以显示服务器页面。
3. 选择服务器 ID 列中的标识符以查看服务器详细信息页面。
4. 在用户下，选择一个用户名以查看用户详细信息页面。

   您可以通过选择编辑来更改该页面上的用户属性。
5. 在用户详细信息页面上，选择用户配置旁边的编辑。



6. 在编辑配置页面上的访问权限，选择您之前创建的 IAM 角色，该角色提供对您的 Amazon S3 存储桶的访问权限。

   您可使用创建 IAM 角色和策略中的过程创建此 IAM 角色。该 IAM 角色包括一个提供对您 Amazon S3 存储桶访问权限的 IAM policy。它还包括与 AWS Transfer Family 服务的信任关系，在另一个 IAM 策略中定义。

7. （可选）对于策略，请选择以下选项之一：

   - 无
   - 现有策略
   - 从 IAM 中选择策略以选择现有策略。选择查看以查看包含策略详细信息的 JSON 对象。

   要了解有关会话策略的更多信息，请参阅创建 IAM 角色和策略。要了解有关创建会话策略的更多信息，请参阅为 Amazon S3 存储桶创建会话策略。

8. 对于主目录，选择 Amazon S3 存储桶用于存储使用 AWS Transfer Family 传输的数据。输入用户在使用其客户端登录时转到的 home 目录的路径。

如果您将此参数留空，则使用 Amazon S3 存储桶的 `root` 目录。在这种情况下，请确保您的 IAM 角色提供对此 `root` 目录的访问权限。

> **Note**
>
> 我们建议您选择包含用户的用户名的目录路径，这使得您可以更高效地使用会话策略。会话策略将用户在 Amazon S3 存储桶中的访问权限限制为该用户的 home 目录。

9. （可选）对于受限，选中该复选框，这样您的用户就无法访问该文件夹之外的任何内容，也看不到 Amazon S3 存储桶或文件夹名称。

> **Note**
>
> 当为用户分配主目录并限制用户访问该主目录时，这应该足以锁定用户对指定文件夹的访问权限。当您需要应用进一步的控制措施，请使用会话策略。

10. 选择保存，保存您的更改。

删除用户

1. 打开AWS Transfer Family控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 从导航窗格中选择服务器以显示服务器页面。
3. 选择服务器 ID 列中的标识符以查看服务器详细信息页面。
4. 在用户下，选择一个用户名以查看用户详细信息页面。
5. 在用户详细信息页面上，选择用户名右侧的删除。
6. 在显示的确认对话框中，输入单词 **`delete`**，然后选择删除以确认您要删除该用户。

将从用户列表中删除该用户。

为用户添加 SSH 公钥

1. 打开AWS Transfer Family控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在导航窗格中，选择 Servers (服务器)。
3. 选择服务器 ID 列中的标识符以查看服务器详细信息页面。
4. 在用户下，选择一个用户名以查看用户详细信息页面。

5. 选择 Add SSH public key (添加 SSH 公有密钥) 以向用户添加新的 SSH 公有密钥。

> **Note**
>
> SSH 密钥仅由启用 Secure Shell (SSH) 文件传输协议 (SFTP) 的服务器使用。有关如何生成 SSH 密钥对的信息，请参阅为服务托管用户生成 SSH 密钥。

6. 对于 SSH public key (SSH 公有密钥)，输入 SSH 密钥对的 SSH 公有密钥部分。

   您的密钥先由服务进行验证，然后才能添加新用户。SSH 密钥的格式为 `ssh-rsa string`。要生成 SSH 密钥对，请参阅为服务托管用户生成 SSH 密钥。

7. 选择 Add key (添加密钥)。

删除用户的 SSH 公钥

1. 打开AWS Transfer Family控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在导航窗格中，选择 Servers (服务器)。
3. 选择服务器 ID 列中的标识符以查看服务器详细信息页面。
4. 在用户下，选择一个用户名以查看用户详细信息页面。
5. 要删除公钥，请选中其 SSH 密钥复选框并选择删除。

使用 Di AWS rectory Service 身份提供商

本主题介绍如何使用 AWS Directory Service 身份提供商 AWS Transfer Family。

主题

- 使用 AWS Directory Service for Microsoft Active Directory
- 使用 AWS Azure 活动目录域服务的目录服务

使用 AWS Directory Service for Microsoft Active Directory

您可以使用 AWS Transfer Family 对文件传输的最终用户进行身份验证 AWS Directory Service for Microsoft Active Directory。它可以无缝迁移依赖于活动目录身份验证的文件传输工作流程，而无需更改最终用户的凭证或需要自定义授权者。

使用 AWS Managed Microsoft AD，您可以通过 SFTP、FTPS 和 FTP 安全地为 AWS Directory Service 用户和群组提供对存储在亚马逊简单存储服务 (Amazon S3) 或亚马逊弹性文件系统 (Amazon EFS) 中的数据的访问权限。如果您使用活动目录来存储用户的凭证，则现在可以更轻松地为这些用户启用文件传输功能。

您可以使用 Active Directory 连接器在本地环境 AWS Managed Microsoft AD 中或 AWS 云端提供对 Active Directory 组的访问权限。你可以为已经在你的 Microsoft Windows 环境（无论是在 AWS 云端还是在其本地网络中）中配置的用户提供访问 AWS Managed Microsoft AD 用于身份的 AWS Transfer Family 服务器的权限。

> **Note**
>
> - AWS Transfer Family 不支持 Simple AD。
> - Transfer Family 不支持跨区域活动目录配置：我们仅支持与 Transfer Family 服务器位于同一区域的活动目录集成。
> - Transfer Family 不支持使用 AD Connecto AWS Managed Microsoft AD r 为现有的基于 RADIUS 的 MFA 基础设施启用多因素身份验证 (MFA)。
> - AWS Transfer Family 不支持托管活动目录的复制区域。

要使用 AWS Managed Microsoft AD，必须执行以下步骤：

1. 使用 AWS Directory Service 控制台创建一个或多个 AWS Managed Microsoft AD 目录。
2. 使用 Transfer Family 控制台创建 AWS Managed Microsoft AD 用作其身份提供者的服务器。
3. 添加来自一个或多个 AWS Directory Service 群组的访问权限。
4. 尽管不是必需的，但我们建议您测试和验证用户访问权限。

主题

- 使用 SSH（安全外壳）连接到服务器
- 使用林和 AWS Transfer Family 信任连接到自我管理的 Active Directory

开始使用之前 AWS Directory Service for Microsoft Active Directory

为您的 AD 组提供唯一标识符

在使用之前 AWS Managed Microsoft AD，必须为 Microsoft AD 目录中的每个群组提供唯一标识符。您可以使用每个组的安全标识符 (SID) 来执行此操作。您关联的群组中的用户可以使用 AWS Transfer Family 通过启用的协议访问您的 Amazon S3 或 Amazon EFS 资源。

使用以下 Windows PowerShell 命令检索组的 SID，*YourGroupName*替换为该组的名称。

```
Get-ADGroup -Filter {samAccountName -like "YourGroupName*"} -Properties * | Select
 SamAccountName,ObjectSid
```

> **Note**
>
> 如果您使用 AWS Directory Service 作为身份提供商，并且如果`userPrincipalName`和`SamAccountName`具有不同的值，则 AWS Transfer Family 接受中的值`SamAccountName`。Transfer Family 不接受`userPrincipalName`中指定的值。

为您的角色添加 AWS Directory Service 权限

您还需要 AWS Directory Service API 权限才能 AWS Directory Service 用作身份提供商。需要建议使用以下权限：

- `ds:DescribeDirectories`是 Transfer Family 查找目录所必需的
- `ds:AuthorizeApplication`是为 Transfer Family 添加授权所必需的
- `ds:UnauthorizeApplication`是移除所有临时创建的资源，以防服务器创建过程中出现问题所建议的

将这些权限添加到您用于创建 Transfer Family 服务器的角色中。有关这些权限的更多详细信息，请参阅 AWS Directory Service API 权限：操作、资源和条件参考。

使用活动目录领域

在考虑如何让活动目录用户访问 AWS Transfer Family 服务器时，请记住用户的领域及其组的领域。理想情况下，用户的领域和他们所在群组的领域应该匹配。也就是说，用户和组都在默认领域中，或者两者都位于可信领域。如果不是这样，Transfer Family 将无法对用户进行身份验证。

您可以测试用户以确保配置正确。有关更多信息，请参阅 测试用户。如果用户/组领域出现问题，则会收到错误 No associated access found for user's groups（未找到用户组的关联访问权限）。

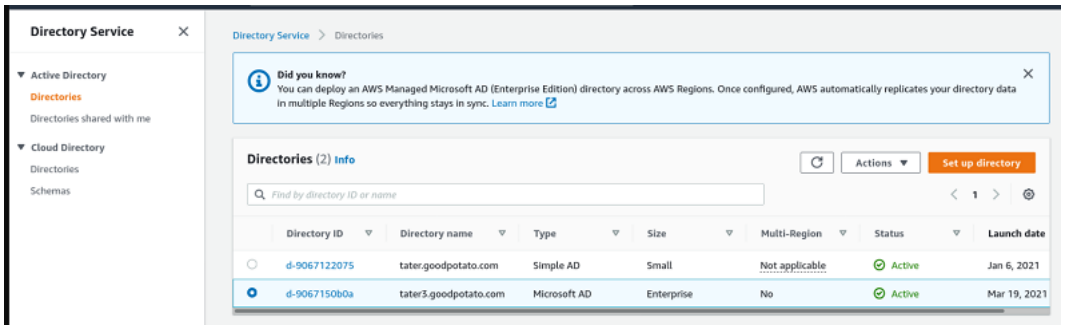
选择 AWS Managed Microsoft AD 作为您的身份提供商

本节介绍如何与服务器 AWS Directory Service for Microsoft Active Directory 一起使用。

要与 Transfer Family AWS Managed Microsoft AD 一起使用

1. 登录 AWS Management Console 并打开 AWS Directory Service 控制台，网址为 https://console.aws.amazon.com/directoryservicev2/。

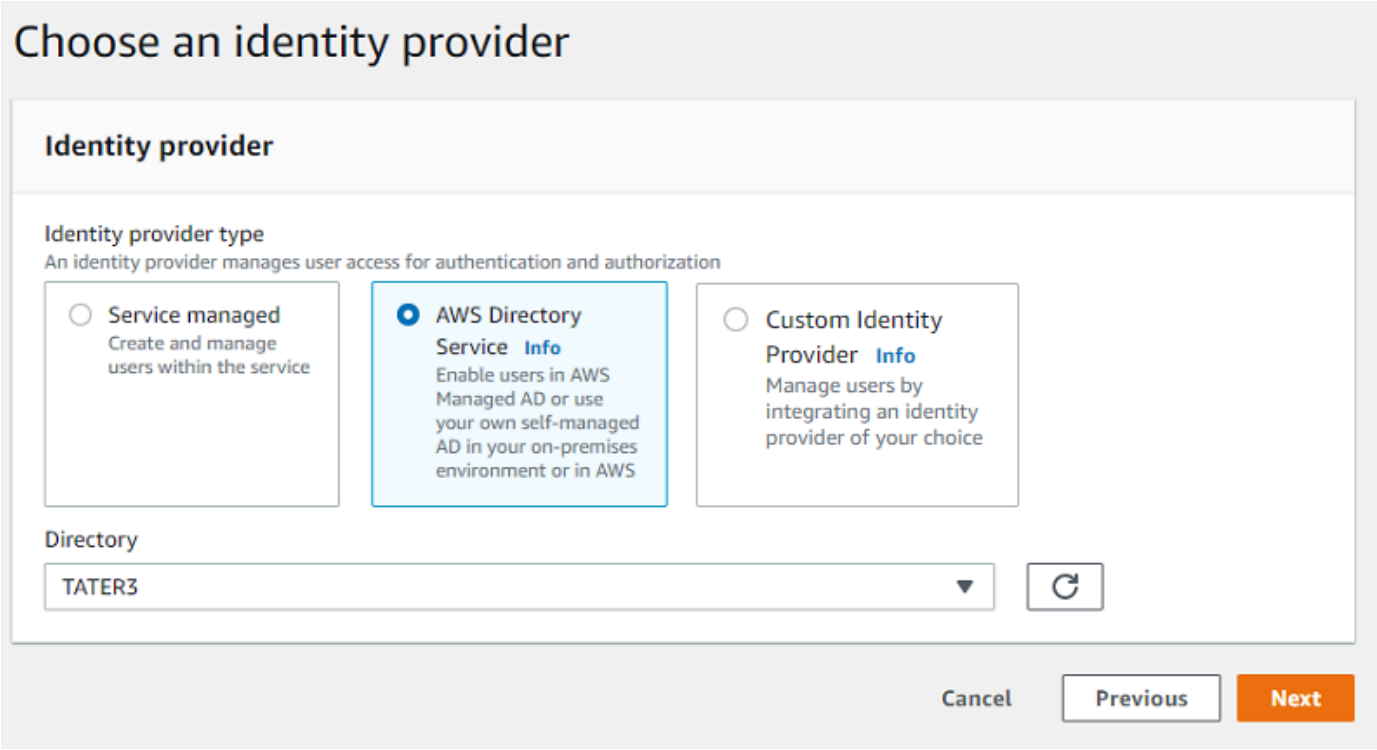
   使用 AWS Directory Service 控制台配置一个或多个托管目录。有关更多信息，请参阅《AWS Directory Service 管理员指南》中的 AWS Managed Microsoft AD。



2. 通过 https://console.aws.amazon.com/transfer/ 打开 AWS Transfer Family 控制台，然后选择 “创建服务器”。
3. 在选择协议页面上，从列表中选择一个或多个协议。

   > **Note**
   >
   > 如果选择 FTPS，则必须提供 AWS Certificate Manager 证书。

4. 在选择身份提供程序中，选择AWS 目录服务。



5. 目录列表包含您配置的所有托管目录。从列表中选择目录，然后选择下一步。

> **Note**
>
> - 不支持跨账户目录和共享目录。 AWS Managed Microsoft AD
> - 要设置以 Directory Service 作为身份提供者的服务器，您需要添加一些 AWS Directory Service 权限。有关更多信息，请参阅 开始使用之前 AWS Directory Service for Microsoft Active Directory。

6. 要完成服务器的创建，请使用下列过程之一：

- 创建启用 SFTP 的服务器
- 创建启用 FTPS 的服务器
- 创建启用 FTP 的服务器

在这些步骤中，继续执行选择身份提供程序之后的步骤。

> **Important**
>
> AWS Directory Service 如果你在 Transfer Family 服务器中使用了 Microsoft AD 目录，则无法将其删除。必须先删除服务器，然后才能删除目录。

授予对组的访问权限

创建服务器后，您必须选择目录中哪些组应有权通过启用的协议使用上传和下载文件 AWS Transfer Family。您可以通过创建访问权限来实现此目的。

> **Note**
>
> 用户必须直接属于您授予访问权限的群组。例如，假设 Bob 是一个用户并且他属于 GroupA，而 GroupA 本身包含在 groupB 中。
>
> - 如果您向 GroupA 授予访问权限，Bob 就会被授予访问权限。
> - 如果您授予对 GroupB（而不是 GroupA）的访问权限，则 Bob 没有访问权限。

向组授予访问权限

1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 导航到您的服务器详细信息页面。
3. 在访问权限部分中，选择添加访问权限。
4. 输入您想要访问此服务器的 AWS Managed Microsoft AD 目录的 SID。

   > **Note**
   >
   > 有关如何查找组的 SID 的信息，请参阅the section called “开始使用之前 AWS Directory Service for Microsoft Active Directory”。

5. 对于访问权限，请为群组选择一个 AWS Identity and Access Management (IAM) 角色。
6. 在策略部分，选择一个策略。默认设置为无。
7. 对于主目录，选择与该组的主目录对应的 S3 存储桶。

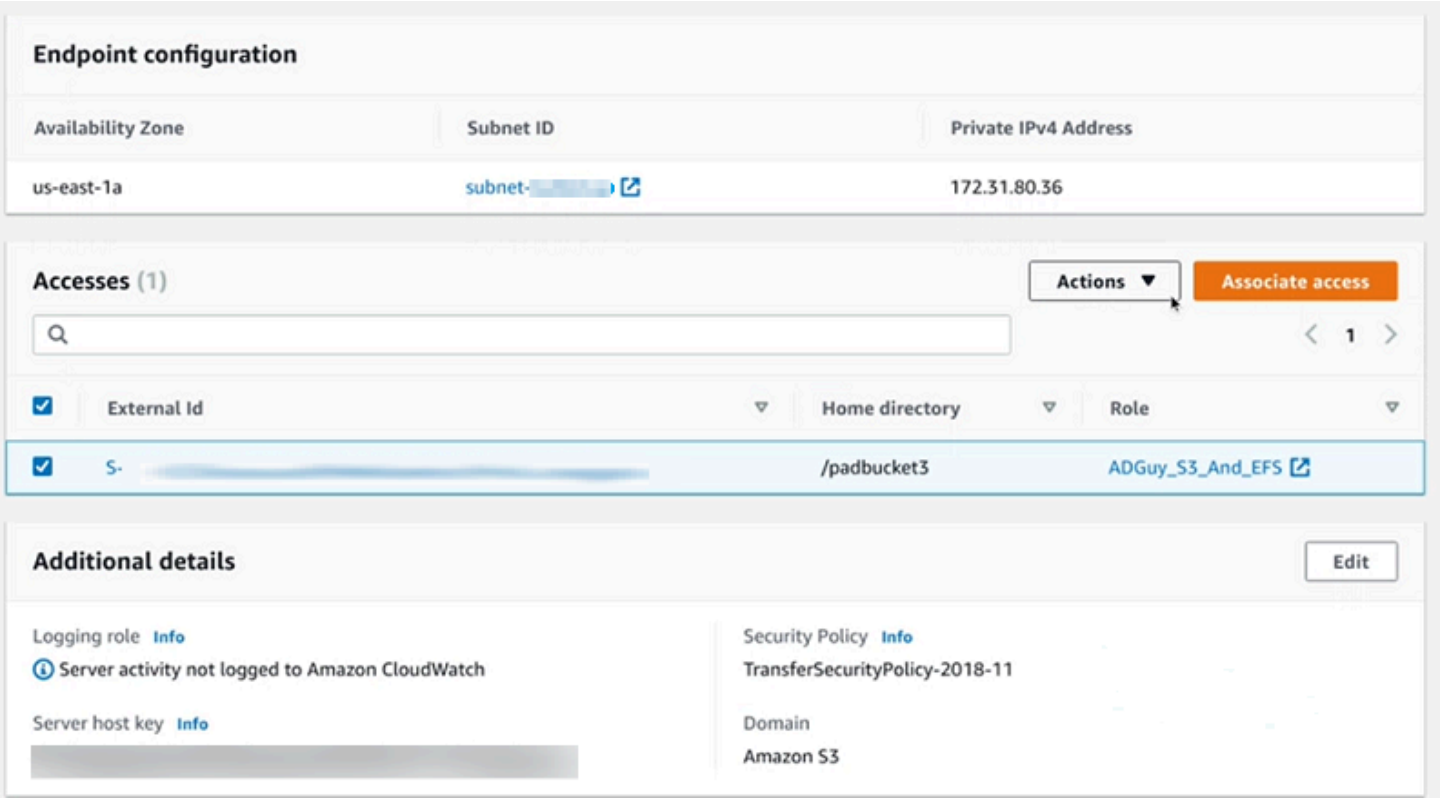
> **Note**
>
> 您可以通过创建会话策略来限制用户在存储桶中看到的部分。例如，要将用户限制在/filetest目录下他们自己的文件夹中，请在框中输入以下文本。
>
> ```
> /filetest/${transfer:UserName}
> ```
>
> 要了解有关创建会话策略的更多信息，请参阅为 Amazon S3 存储桶创建会话策略。

8. 选择添加以创建关联。
9. 请选择您的服务器。
10. 选择添加访问权限。
    - 输入该组的 SID。

    > **Note**
    >
    > 有关如何查找 SID 的信息，请参阅the section called “开始使用之前 AWS Directory Service for Microsoft Active Directory”。

11. 选择添加访问权限。

在访问权限部分中，列出了服务器的访问权限。



测试用户

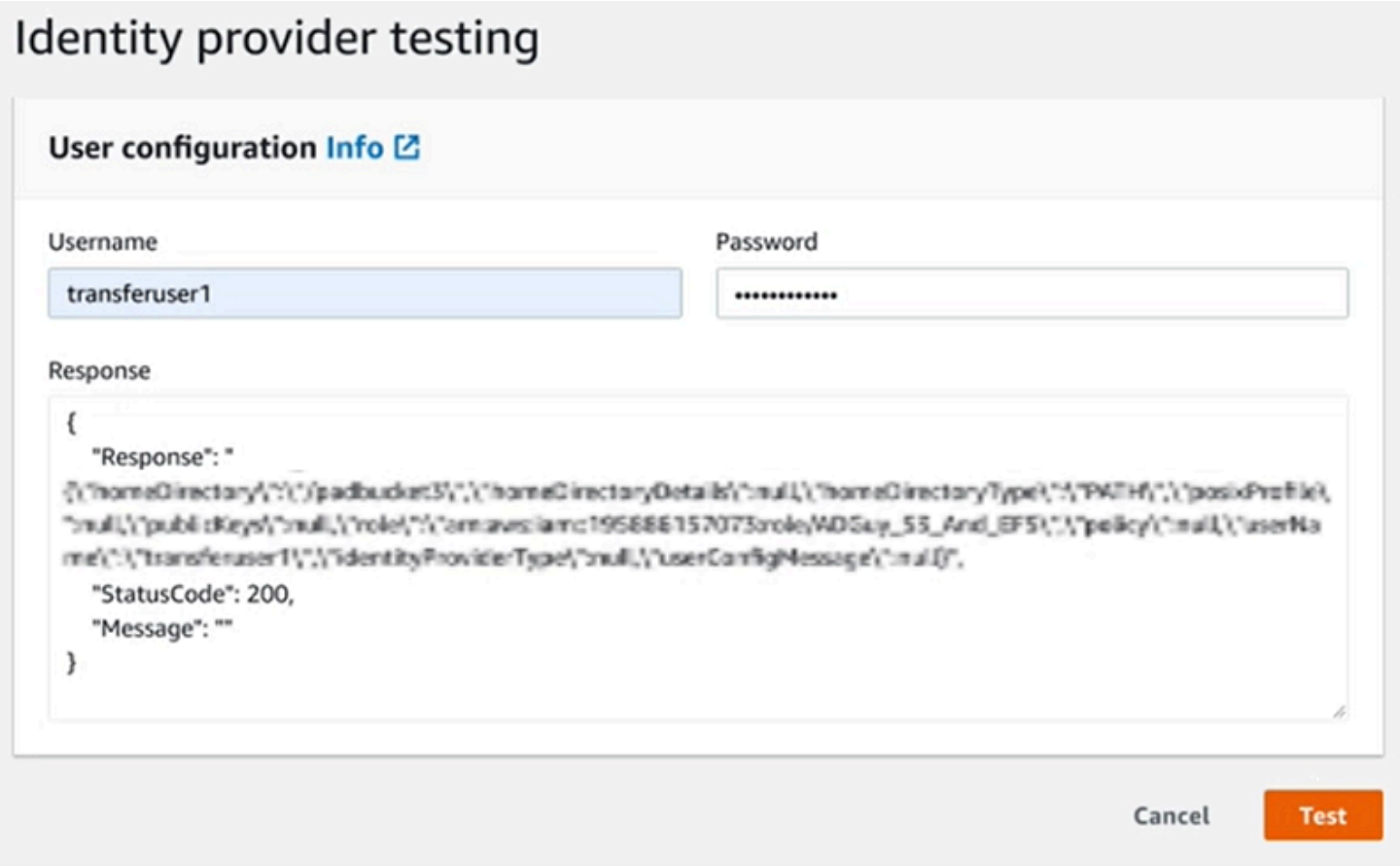
您可以测试用户是否有权访问您的服务器的 AWS Managed Microsoft AD 目录。

> **Note**
>
> 用户必须正好属于端点配置页面的访问权限部分中列出的一个组（外部 ID）。如果用户不属于任何群组，或者属于多个群组，则不会向该用户授予访问权限。

测试特定用户是否具有访问权限

1. 在服务器详细信息页面上，选择操作，然后选择测试。
2. 要进行身份提供程序测试，请输入其中一个具有访问权限的群组中的用户的登录凭证。
3. 选择测试。

您会看到身份提供程序测试成功，显示所选用户已被授予服务器访问权限。



如果用户属于多个具有访问权限的群组，则您会收到以下响应。

```
"Response":"",
"StatusCode":200,
"Message":"More than one associated access found for user's groups."
```

删除群组的服务器访问权限

删除群组的服务器访问权限

1. 在服务器详细信息页面上，选择操作，然后选择删除访问权限。
2. 在对话框中，确认您要移除该组的访问权限。

返回到服务器详细信息页面时，您会看到不再列出该组的访问权限。

使用 SSH（安全外壳）连接到服务器

配置服务器和用户后，您可以使用 SSH 连接到服务器，并使用具有访问权限的用户的完全限定用户名。

```
sftp user@active-directory-domain@vpc-endpoint
```

例如：`transferuserexample@mycompany.com@vpce-0123456abcdef-789xyz.vpc-svc-987654zyxabc.us-east-1.vpce.amazonaws.com`。

此格式以联合身份验证搜索为目标，限制了对可能很大的活动目录的搜索。

> **Note**
>
> 您可以指定简单的用户名。但是，在这种情况下，活动目录代码必须搜索联合身份验证中的所有目录。这可能会限制搜索，即使用户本应具有访问权限，身份验证也可能失败。

身份验证后，用户位于在配置用户时指定的主目录中。

使用林和 AWS Transfer Family 信任连接到自我管理的 Active Directory

您自行管理的 Active Directory (AD) 中的用户也可以使用 AWS IAM Identity Center 单点登录访问 AWS 账户 和 Transfer Family 服务器。为此， AWS Directory Service 有以下选项可用：

- 单向林信任（来自本地 Active Directory 的传出 AWS Managed Microsoft AD 和传入）仅适用于根域。
- 对于子域，您可以使用以下方法之一：
  - 在 AWS Managed Microsoft AD 和本地活动目录之间使用双向信任
  - 对每个子域使用单向外部信任。

例如，当使用可信域连接到服务器时，用户需要指定可信域，例如`transferuserexample@mycompany.com`。

使用 AWS Azure 活动目录域服务的目录服务

- 要利用现有的活动目录林来满足 SFTP 传输需求，可以使用 Active Directory Connector。
- 如果您想在完全托管的服务中获得活动目录的好处和高可用性，则可以使用 AWS Directory Service for Microsoft Active Directory。有关更多信息，请参阅 使用 Di AWS rectory Service 身份提供商。

本主题介绍如何使用 Active Directory Connector 和 Azure 活动目录域服务 (Azure ADDS) 通过 Azure Active Directory 对 SFTP 传输用户进行身份验证。

主题

开始使用适用于 Azure 的 AWS 目录服务之前 Active Directory 域服务

对于 AWS，你需要以下内容：

- 位于您使用 Transfer Family 服务器的 AWS 区域中的虚拟私有云 (VPC)
- 您的 VPC 中至少有两个私有子网
- VPC 必须具备互联网连接
- 用于连接微软 Azure 的 site-to-site VPN 的客户网关和虚拟专用网关

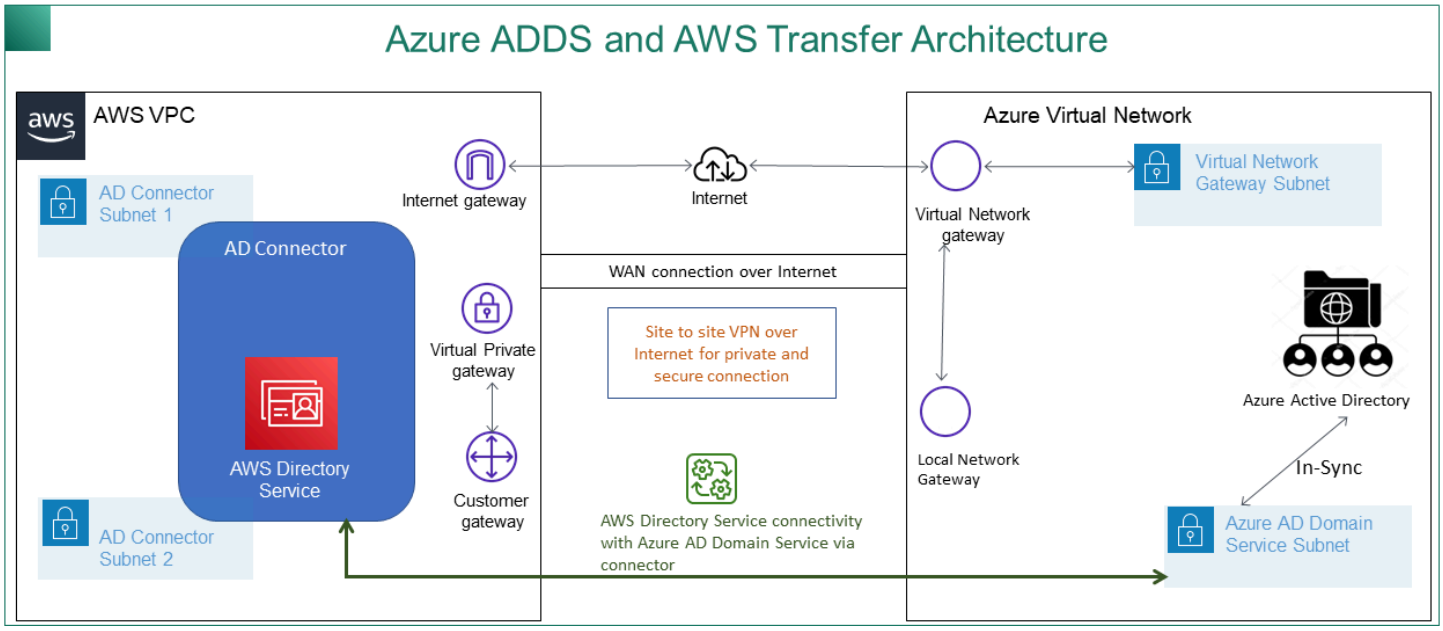
对于微软 Azure，您需要：

- Azure 活动目录和活动目录域服务（Azure ADDS）
- 一个 Azure 资源组
- Azure 虚拟网络
- Amazon VPC 和 Azure 资源组之间的 VPN 连接

> **Note**
>
> 这可以通过本地 IPSEC 隧道或使用 VPN 设备实现。在本主题中，我们使用 Azure 虚拟网络网关和本地网络网关之间的 IPSEC 隧道。必须将隧道配置为允许 Azure ADDS 端点与存放 AWS VPC 的子网之间的流量。

- 用于连接微软 Azure 的 site-to-site VPN 的客户网关和虚拟专用网关

下图显示了在开始之前所需的配置。

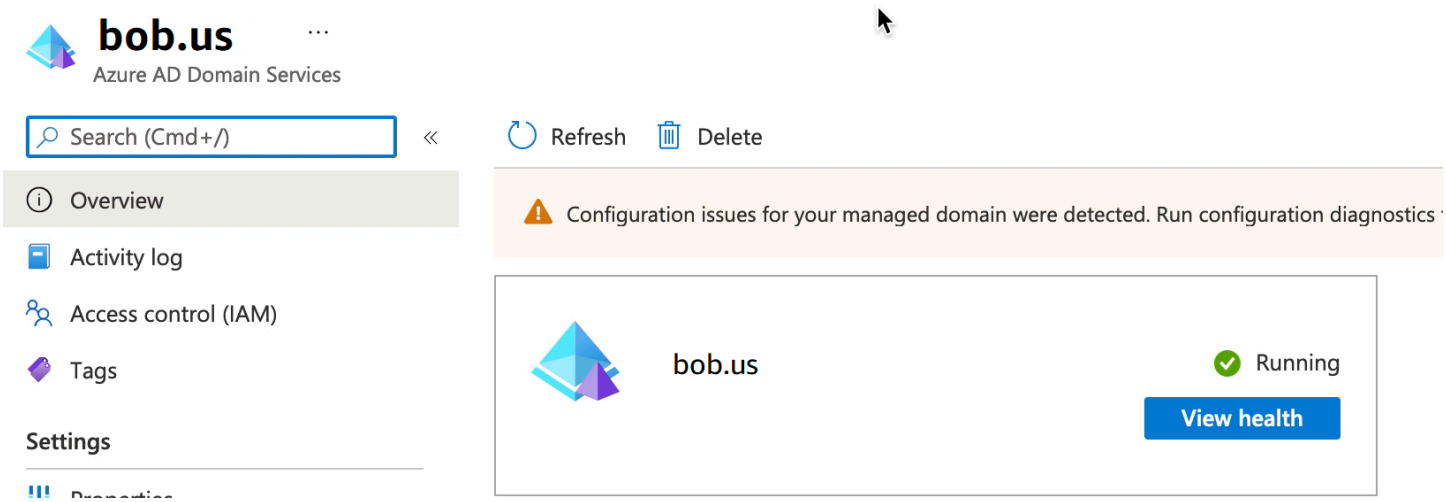


步骤 1：添加 Azure 活动目录域服务

默认情况下，Azure AD 不支持域加入实例。要执行诸如加入域之类的操作以及使用组策略等工具，管理员必须启用 Azure Active Directory 域服务。如果您尚未添加 Azure AD DS，或者现有实现与您希望 SFTP 传输服务器使用的域没有关联，则必须添加一个新实例。

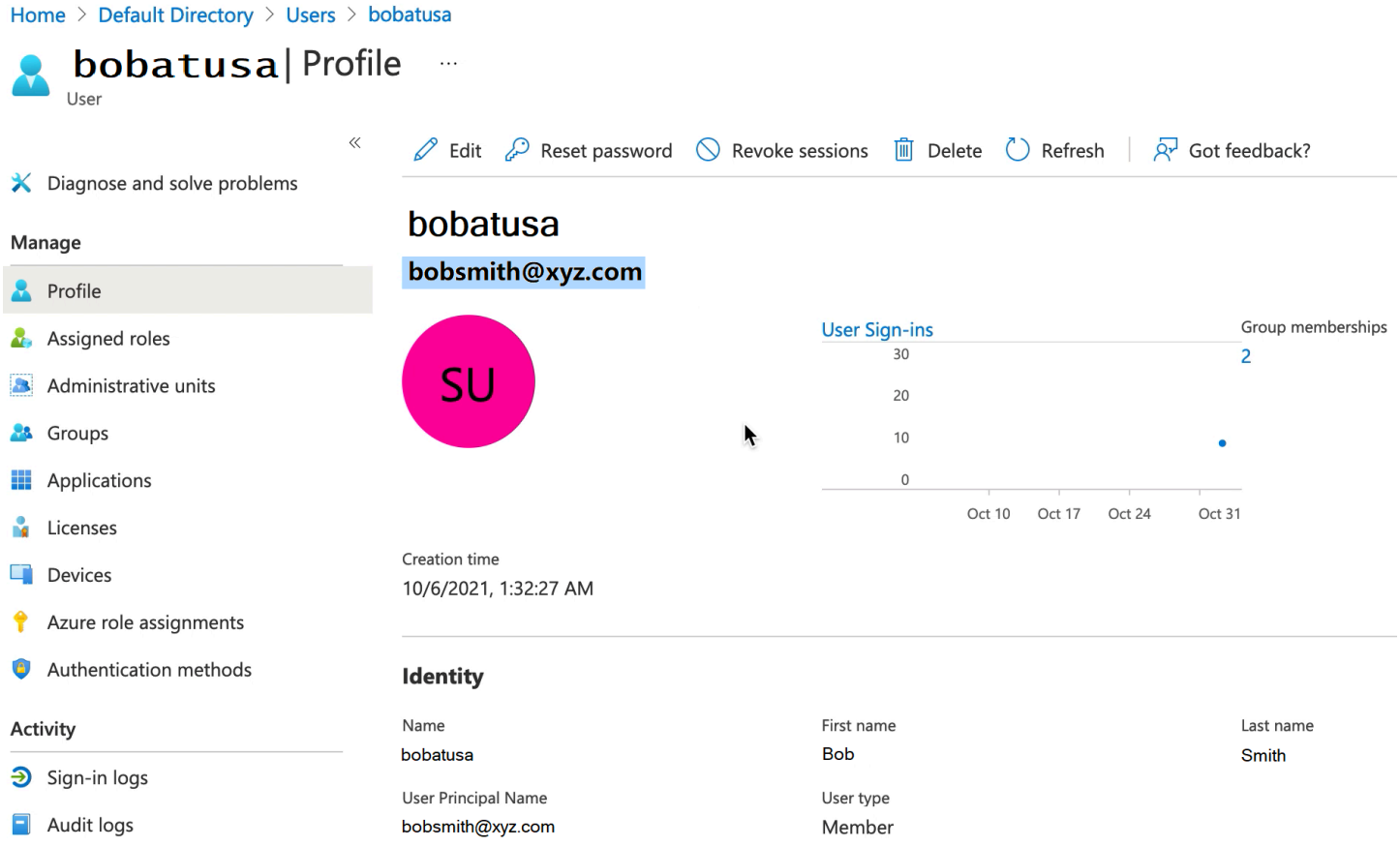
有关启用 Azure 活动目录域服务（Azure ADDS）的信息，请参阅教程：创建和配置 Azure 活动目录域服务托管域。

> **Note**
>
> 启用 Azure ADDS 时，请确保已针对资源组和 SFTP 传输服务器连接的 Azure AD 域进行了配置。



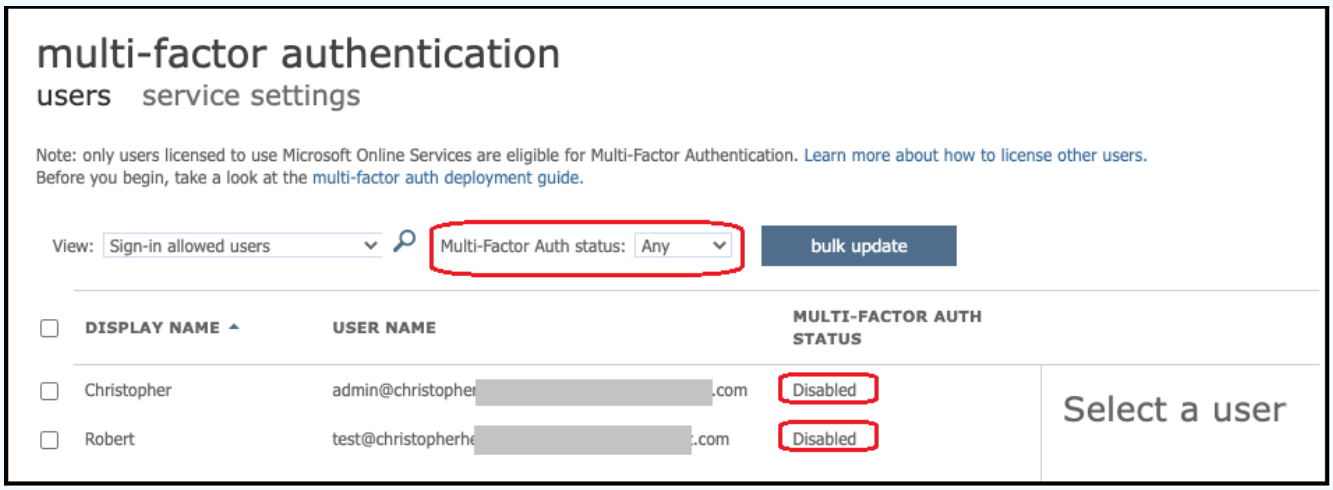
步骤 2：创建服务账号

Azure AD 必须有一个服务账户，该账户必须是 Azure ADDS 中管理员组的一部分。此帐户与 Act AWS ive Directory 连接器一起使用。确保此账户与 Azure ADDS 同步。



Tip

使用 SFTP 协议的 Transfer Family 服务器不支持 Azure Active Directory 的多重身份验证。在用户向 SFTP 进行身份验证后，Transfer Family 服务器无法提供 MFA 令牌。在尝试连接之前，请务必禁用 MFA。



步骤 3：使用 AD Connector 设置 AWS 目录

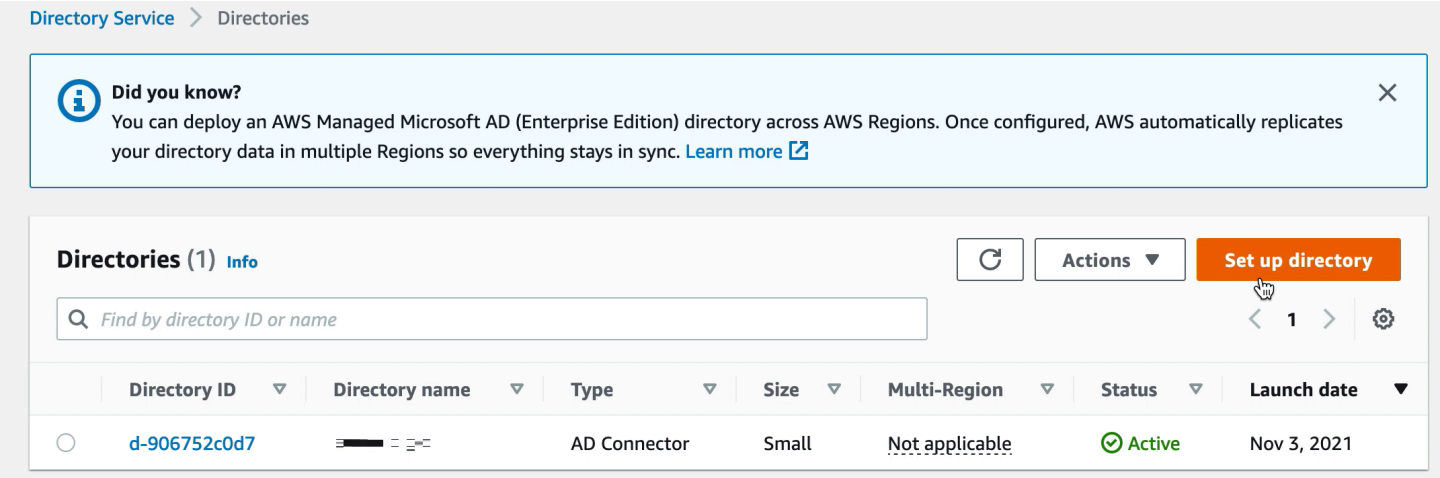
在配置了 Azure ADDS 并创建了在 VPC 和 Azure 虚拟网络之间具有 IPSE AWS C VPN 隧道的服务帐户后，你可以通过从任何 AWS EC2 实例执行 ping Azure ADDS DNS IP 地址来测试连接。

在您确认连接处于活动状态后，您可以继续执行以下操作。

使用 AD Connector 设置您的 AWS 目录

1. 打开 Directory Service 控制台并选择目录。
2. 选择设置目录。
3. 对于目录类型，请选择 AD Connector。
4. 选择目录大小，选择下一步，然后选择您的 VPC 和子网。
5. 选择 下一步，然后如下所示填写各字段：
   - 目录 DNS 名称：输入您用于 Azure ADDS 的域名。
   - DNS IP 地址：输入 Azure ADDS IP 地址。
   - 服务器账户用户名和密码：输入您在步骤 2：创建服务账户中创建的服务账户的详细信息。
6. 完成屏幕内容以创建目录服务。

现在，目录状态应为活动，并且可以与 SFTP 传输服务器一起使用了。



步骤 4：设置 AWS Transfer Family 服务器

使用 SFTP 协议和 AWS Directory Service 身份提供程序类型创建 Transfer Family 服务器。从目录下拉列表中，选择您在步骤 3：使用 AD Connector 设置 AWS 目录中添加的目录。

> **Note**
>
> 如果你在 Transfer Family 服务器中使用了 Microsoft AD AWS 目录，则无法将其删除。必须先删除服务器，然后才能删除目录。

步骤 5：授予对组的访问权限

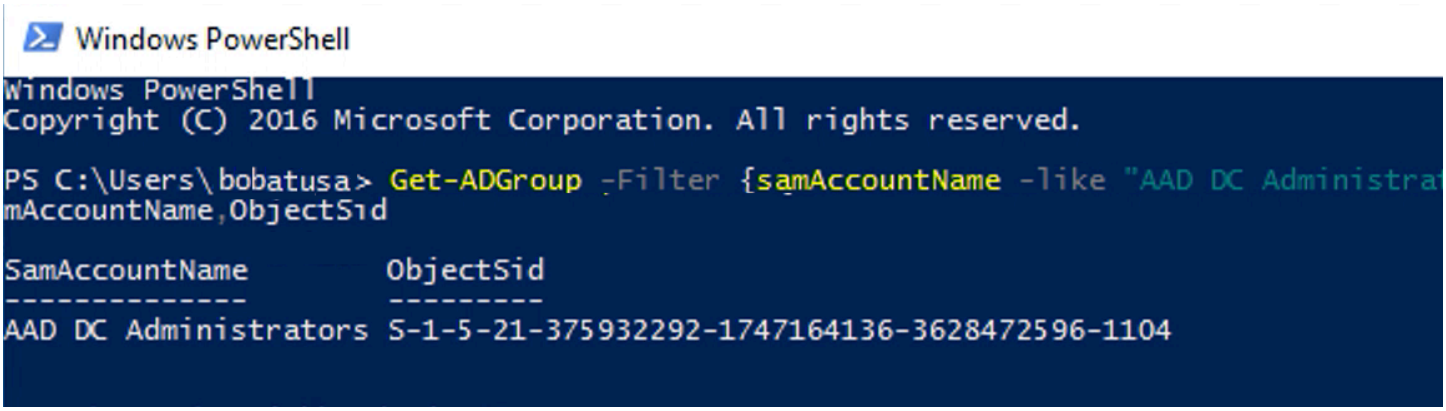
创建服务器后，您必须选择目录中哪些组应有权通过启用的协议使用上传和下载文件 AWS Transfer Family。您可以通过创建访问权限来实现此目的。

> **Note**
>
> 用户必须直接属于您授予访问权限的群组。例如，假设 Bob 是一个用户并且他属于 GroupA，而 GroupA 本身包含在 groupB 中。
>
> - 如果您向 GroupA 授予访问权限，Bob 就会被授予访问权限。
> - 如果您授予对 GroupB（而不是 GroupA）的访问权限，则 Bob 没有访问权限。

要授予访问权限，您需要检索该组的 SID。

使用以下 Windows PowerShell 命令检索组的 SID，*YourGroupName*替换为该组的名称。

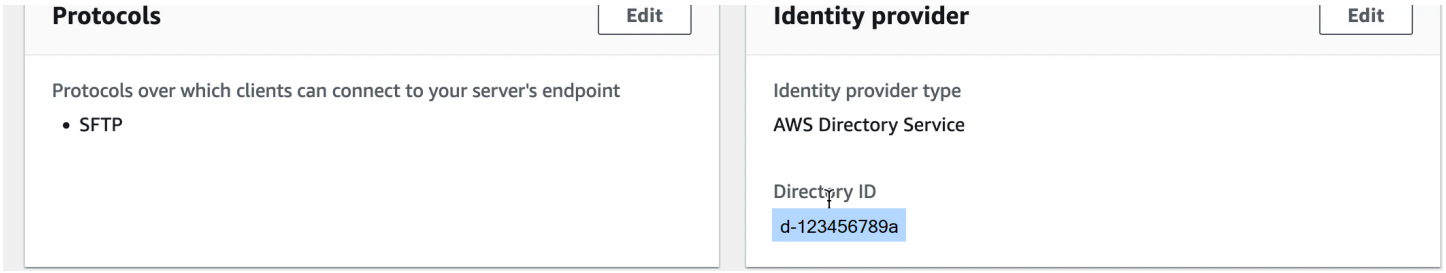
```
Get-ADGroup -Filter {samAccountName -like "YourGroupName*"} -Properties * | Select
 SamAccountName,ObjectSid
```

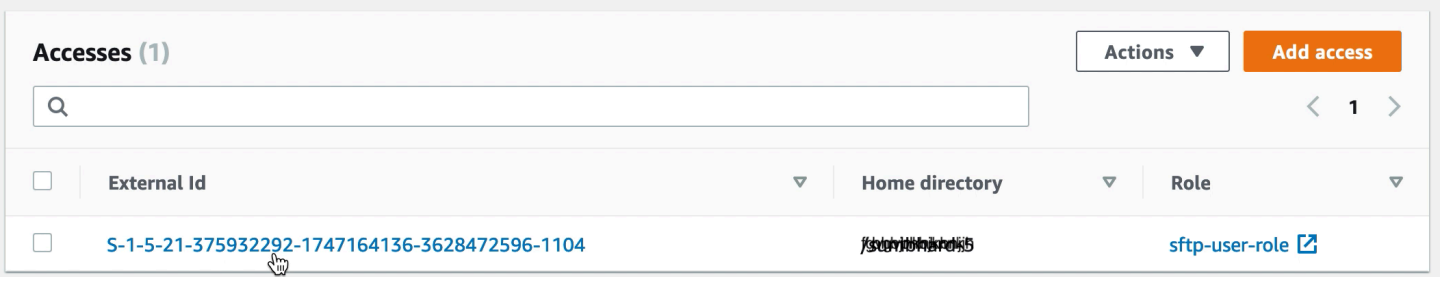


授予对组的访问权限

1. 打开 https://console.aws.amazon.com/transfer/。
2. 导航到您的服务器详细信息页面，然后在访问权限部分中，选择添加访问权限。
3. 输入您从上一个过程的输出中收到的 SID。
4. 在 “访问权限” 中，为群组选择一个 AWS Identity and Access Management 角色。
5. 在策略部分，选择一个策略。默认值为 None（无）。
6. 对于主目录，选择与该组的主目录对应的 S3 存储桶。
7. 选择添加以创建关联。

您的 Transfer 服务器中的详细信息应类似于以下内容：





步骤 6：测试用户

您可以测试 (测试用户) 用户是否有权访问您的服务器的 AWS Managed Microsoft AD 目录。用户必须正好属于端点配置页面的访问权限部分中列出的一个组（外部 ID）。如果用户不属于任何群组，或者属于多个群组，则不会向该用户授予访问权限。

使用自定义身份提供程序

要对用户进行身份验证，您可以使用现有的身份提供商 AWS Transfer Family。您可以使用功能集成您的身份提供商，该 AWS Lambda 功能对您的用户进行身份验证和授权，使其能够访问 Amazon S3 或亚马逊弹性文件系统 (Amazon EFS)。有关更多信息，请参阅 AWS Lambda 用于整合您的身份提供商。您还可以访问 AWS Transfer Family 管理控制台中传输的文件数和字节数等指标的 CloudWatch 图表，从而通过单一控制面板使用集中式仪表板监控文件传输。

或者，您可以提供带有单个 Amazon API Gateway 方法的 RESTful 接口。Transfer Family 调用此方法连接到您的身份提供程序，该提供程序会对您的用户进行身份验证和授权，使其能够访问 Amazon S3 或 Amazon EFS。如果您需要 RESTful API 来集成您的身份提供商，或者您想使用 AWS WAF 其功能来处理地理封锁或速率限制请求，请使用此选项。有关更多信息，请参阅 使用 Amazon API Gateway 整合您的身份提供程序。

无论哪种情况，您都可以使用 AWS Transfer Family 控制台或 CreateServer API 操作创建新服务器。

> **Note**
>
> 我们有一个可供您参加的研讨会，您可以在其中构建文件传输解决方案。该解决方案利用 AWS Transfer Family 托管 SFTP/FTPS 终端节点，利用 Amazon Cognito 和 DynamoDB 进行用户管理。您可以在此处查看本次研讨会的详细信息。

AWS Transfer Family 提供了以下与自定义身份提供商合作的选项。

- AWS Lambda 用于连接您的身份提供商-您可以使用由 Lambda 函数支持的现有身份提供商。您提供 Lambda 函数名称。有关更多信息，请参阅 AWS Lambda 用于整合您的身份提供商。

- 使用 Amazon API Gateway 连接您的身份提供商 — 您可以创建由 Lambda 函数支持的 API 网关方法以用作身份提供商。您提供一个 Amazon API Gateway URL 和一个调用角色。有关更多信息，请参阅 使用 Amazon API Gateway 整合您的身份提供程序。

对于任一选项，您还可以指定如何进行身份验证。

- 密码或密钥-用户可以使用其密码或密钥进行身份验证。这是默认值。
- 仅限密码-用户必须提供密码才能连接。
- 仅限密钥 — 用户必须提供私钥才能连接。
- 密码和密钥 — 用户必须同时提供私钥和密码才能连接。服务器首先检查密钥，如果密钥有效，系统会提示输入密码。如果提供的私有密钥与存储的公有密钥不匹配，则身份验证失败。

使用多种身份验证方法向您的自定义身份提供商进行身份验证

当您使用多种身份验证方法时，Transfer Family 服务器会控制 AND 逻辑。Transfer Family 将其视为向您的自定义身份提供者发出的两个单独请求：但是，它们的效果是结合在一起的。

两个请求都必须成功返回并返回正确的响应，才能完成身份验证。Transfer Family 要求这两个响应必须完整，这意味着它们包含所有必需的元素（角色、主目录、策略和 POSIX 配置文件（如果您使用 Amazon EFS 进行存储）。Transfer Family 还要求密码响应中不得包含公钥。

公钥请求必须有来自身份提供者的单独响应。使用 “密码或密钥” 或 “密码和密钥” 时，这种行为不会改变。

SSH/SFTP 协议首先向软件客户端发送公钥身份验证，然后请求密码身份验证。此操作要求在允许用户完成身份验证之前，两者都必须成功。

主题

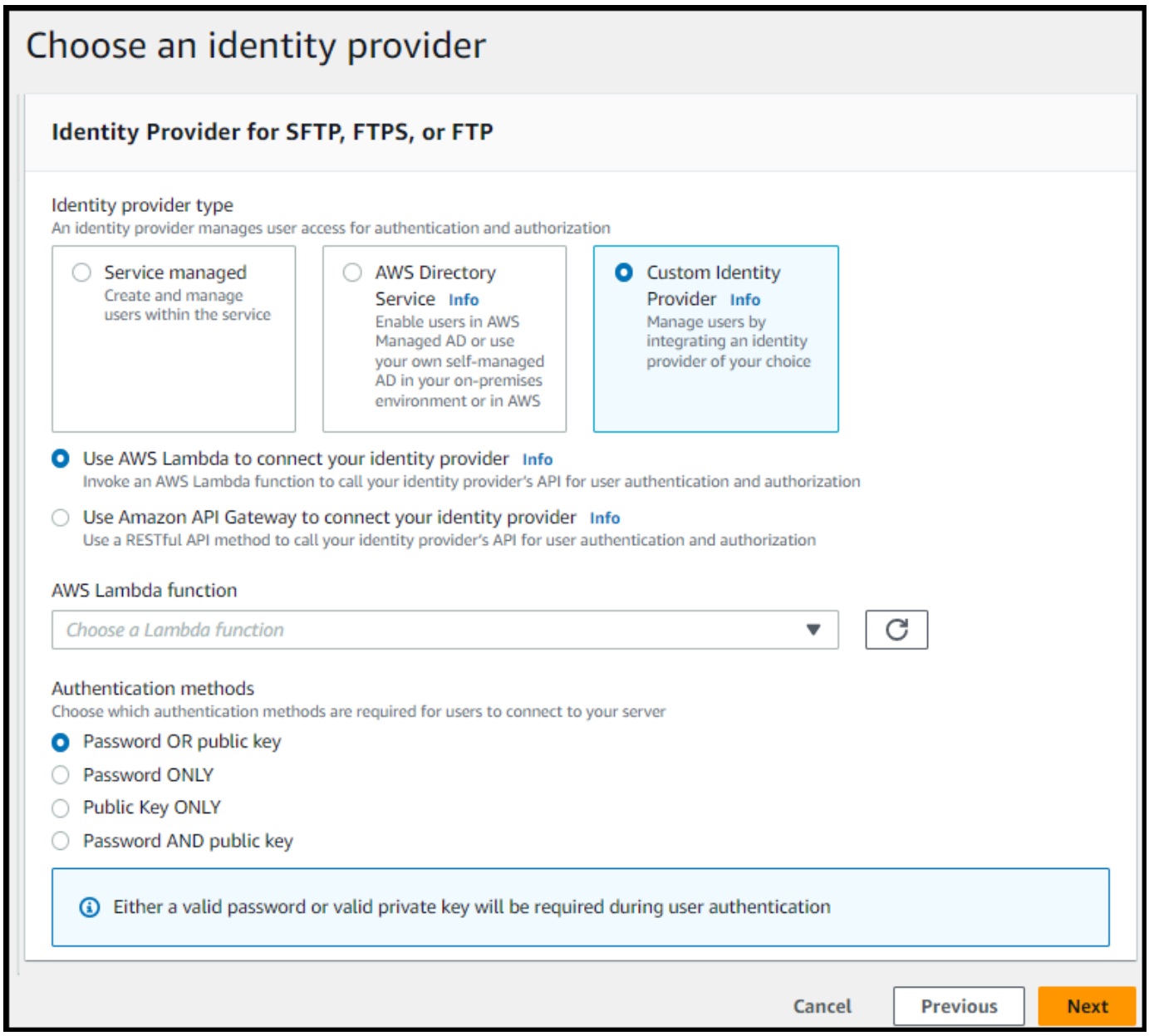
- AWS Lambda 用于整合您的身份提供商
- 使用 Amazon API Gateway 整合您的身份提供程序

AWS Lambda 用于整合您的身份提供商

创建连接到您的自定义身份提供商的 AWS Lambda 函数。您可以使用任何自定义身份提供商，例如 Okta、Secrets Manager 或包含授权和身份验证逻辑的自定义数据存储。 OneLogin

> **Note**
>
> 在创建使用 Lambda 作为身份提供程序的 Transfer Family 服务器之前，必须创建该函数。有关示例 Lambda 函数，请参阅 Lambda 函数示例。或者，您可以部署使用其中一个的 CloudFormation 堆栈Lambda 函数模板。此外，请确保您的 Lambda 函数使用信任 Transfer Family 的基于资源的策略。有关策略示例，请参阅Lambda 资源策略。

1. 打开AWS Transfer Family 控制台。
2. 选择“创建服务器”以打开“创建服务器”页面。在“选择身份提供程序”中，选择“自定义身份提供程序”，如以下屏幕截图所示。



> **Note**
>
> 只有启用 SFTP 作为 Transfer Family 服务器的协议之一时，才能选择身份验证方法。

3. 确保选择了默认值 “ AWS Lambda 用于连接您的身份提供商”。
4. 对于AWS Lambda 函数，选择 Lambda 函数名称。
5. 填写其余的方框，然后选择“创建服务器”。有关创建服务器的其余步骤的详细信息，请参阅 配置 SFTP、FTPS 或 FTP 服务器端点。

Lambda 资源策略

您必须有一个引用 Transfer Family 服务器和 Lambda ARN 的策略。例如，您可以将以下策略与连接到您的身份提供程序的 Lambda 函数一起使用。策略会以 JSON 格式转义为字符串。

```
"Policy":
"{
  "Version": "2012-10-17",
  "Id": "default",
  "Statement": [
    {
      "Sid": "AllowTransferInvocation",
      "Effect": "Allow",
      "Principal": {
        "Service": "transfer.amazonaws.com"
      },
      "Action": "lambda:InvokeFunction",
      "Resource": "arn:aws:transfer:region:account-id:function:my-lambda-auth-
function",
      "Condition": {
        "ArnLike": {
          "AWS:SourceArn": "arn:aws:transfer:region:account-id:server/server-id"
        }
      }
    }
  ]
}"
```

> **Note**
>
> 在该示例中，将每个####### 替换为您自己的信息。

事件消息结构

来自自定义 IDP 的 SFTP 服务器发送给授权程序 Lambda 函数的事件消息结构如下所示。

```
{
    'username': 'value',
    'password': 'value',
    'protocol': 'SFTP',
    'serverId': 's-abcd123456',
```

```
    'sourceIp': '192.168.0.100'
}
```

其中 username 和 password 是发送到服务器的登录凭证的值。

例如，您可输入以下连接命令。

```
sftp bobusa@server_hostname
```

系统会提示您输入密码：

```
Enter password:
    mysecretpassword
```

您可以在 Lambda 函数中进行检查，方法是在 Lambda 函数中打印传递的事件。此部分与以下文本块类似。

```
{
    'username': 'bobusa',
    'password': 'mysecretpassword',
    'protocol': 'SFTP',
    'serverId': 's-abcd123456',
    'sourceIp': '192.168.0.100'
}
```

FTP 和 FTPS 的事件结构类似：唯一的区别是 protocol 参数会使用这些值，而不是 SFTP。

用于身份验证的 Lambda 函数

要实现不同的身份验证策略，请编辑 Lambda 函数。为了帮助您满足应用程序的需求，您可以部署堆 CloudFormation 栈。有关更多信息，AWS Lambda 开发人员指南 或 通过 Node.js 构建 Lambda 函数。

主题

- Lambda 函数模板
- 有效的 Lambda 值
- Lambda 函数示例
- 测试您的配置

Lambda 函数模板

您可以部署使用 Lambda 函数进行身份验证的 AWS CloudFormation 堆栈。我们提供了多个模板，可使用登录凭证对您的用户进行身份验证和授权。您可以修改这些模板或 AWS Lambda 代码以进一步自定义用户访问权限。

> **Note**
>
> 您可以通过在模板中指定启用 FIPS 的安全策略 AWS CloudFormation 来创建启用 FIPS 的 AWS Transfer Family 服务器。有关可用安全策略的描述，请参见 AWS Transfer Family 服务器的安全策略

创建用于身份验证的 AWS CloudFormation 堆栈

1. 打开 AWS CloudFormation 控制台，网址为 https://console.aws.amazon.com/cloudformation。
2. 按照AWS CloudFormation 用户指南中的选择 AWS CloudFormation 堆栈模板中的使用现有模板部署堆栈的说明进行操作。
3. 使用以下模板之一来创建在 Transfer Family 中进行身份验证的 Lambda 函数。

   - 经典 (Amazon Cognito) 堆栈模板

     用于在中创建用作 AWS Lambda 自定义身份提供者的基本模板 AWS Transfer Family。它会针对 Amazon Cognito 进行身份验证以进行基于密码的身份验证，如果使用基于公钥的身份验证，则会从 Amazon S3 存储桶返回公钥。部署后，您可以修改 Lambda 函数代码以执行不同的操作。

   - AWS Secrets Manager 堆栈模板

     与 AWS Transfer Family 服务器 AWS Lambda 一起使用的基本模板，用于将 Secrets Manager 作为身份提供者进行集成。它根据格式`aws/transfer/`*`server-id`*`/`*`username`*的条目 AWS Secrets Manager 进行身份验证。此外，该密钥必须包含返回给 Transfer Family 的所有用户属性的键值对。部署后，您可以修改 Lambda 函数代码以执行不同的操作。

   - Okta 堆栈模板：与 AWS Transfer Family 服务器 AWS Lambda 一起使用，将 Okta 作为自定义身份提供程序集成的基本模板。
   - Okta-MFA 堆栈模板：一种基本模板，用于 AWS Lambda 与 AWS Transfer Family 服务器一起使用，将 Okta 与 MultiFactor 身份验证集成，作为自定义身份提供商。
   - Azure Active Directory 模板：此堆栈的详细信息在博客文章中描述了使用 Azure 活动目录进行身份验证和 AWS Lambda。 AWS Transfer Family

部署堆栈后，您可以在 CloudFormation 控制台的 Outputs 选项卡上查看有关堆栈的详细信息。

部署其中一个堆栈是将自定义身份提供程序集成到 Transfer Family 工作流程的最简单方法。

有效的 Lambda 值

下表详细介绍了 Transfer Family 接受的用于自定义身份提供程序的 Lambda 函数的值。

值	描述	必填
`Role`	指定控制用户对 Amazon S3 存储桶或 Amazon EFS 文件系统访问权限的 IAM 角色的 Amazon Resource Name (ARN) 。附加到此角色的策略确定在将文件传入和传出 Amazon S3 存储桶或 Amazon EFS 文件系统时要为用户提供的访问权限级别。IAM 角色还应包含一个信任关系，从而允许服务器在为用户的传输请求提供服务时访问您的资源。 有关建立信任关系的详细信息，请参阅 建立信任关系。	必需
`PosixProfile`	完整的 POSIX 身份，包括用户 ID (`Uid`)、组 ID (`Gid`) 和任何辅助组 ID (`SecondaryGids`)，用于控制用户对 Amazon EFS 文件系统的访问。POSIX 权限针对文件系统中的文件和目录设置，用于确定用户在将文件传入和传出 Amazon EFS 文件系统时获得的访问权限级别。	Amazon EFS 后备存储为必填项

值	描述	必填
PublicKeys	对此用户有效的 SSH 公钥值列表。空列表表示这不是有效的登录名。密码认证期间不得返回。	可选
Policy	适用于您的用户的会话策略，可让您跨多个用户使用相同的 IAM 角色。此策略将用户的访问范围缩小至 Amazon S3 存储桶的一部分。	可选
HomeDirectoryType	您希望用户在登录服务器时，用户主目录的登录目录（文件夹）的类型。 • 如果您将其设置为 PATH，则用户将在其文件传输协议客户端中原样看到 Amazon S3 存储桶或 Amazon EFS 路径。 • 如果您将其设置为 LOGICAL，则必须在HomeDirectoryDetails 参数中提供映射，以使 Amazon S3 或 Amazon EFS 路径对用户可见。	可选

值	描述	必填
HomeDirectoryDetails	逻辑目录映射指定哪些 Amazon S3 或 Amazon EFS 路径和密钥应对您的用户可见，以及使其对用户可见的方式。您需要指定Entry和Target对，其中 Entry 显示如何使路径可见，Target 是实际的 Amazon S3 或 Amazon EFS 路径。	如果 HomeDirectoryType 值为 LOGICAL，则为必填项
HomeDirectory	用户使用客户端登录服务器时的登录目录。	可选

> **Note**
>
> HomeDirectoryDetails 是 JSON 映射的字符串表示形式。这与 PosixProfile 形成鲜明对比，后者是一个实际的 JSON 映射对象，PublicKeys 是一个字符串的 JSON 数组。有关特定语言的详细信息，请参阅代码示例。

Lambda 函数示例

本节介绍了一些 NodeJS 和 Python 中的 Lambda 函数示例。

> **Note**
>
> 在这些示例中，用户、角色、POSIX 配置文件、密码和主目录详细信息均为示例，必须将其替换为实际值。

Logical home directory, NodeJS

以下 NodeJS 示例函数为拥有逻辑主目录的用户提供了详细信息。

```
// GetUserConfig Lambda
```

```
exports.handler = (event, context, callback) => {
  console.log("Username:", event.username, "ServerId: ", event.serverId);

  var response;
  // Check if the username presented for authentication is correct. This doesn't
 check the value of the server ID, only that it is provided.
  if (event.serverId !== "" && event.username == 'example-user') {
    var homeDirectoryDetails = [
      {
        Entry: "/",
        Target: "/fs-faa1a123"
      }
    ];
    response = {
      Role: 'arn:aws:iam::123456789012:role/transfer-access-role', // The user is
 authenticated if and only if the Role field is not blank
      PosixProfile: {"Gid": 65534, "Uid": 65534}, // Required for EFS access, but
 not needed for S3
      HomeDirectoryDetails: JSON.stringify(homeDirectoryDetails),
      HomeDirectoryType: "LOGICAL",
    };

    // Check if password is provided
    if (!event.password) {
      // If no password provided, return the user's SSH public key
      response['PublicKeys'] = [ "ssh-
rsa abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789" ];
    // Check if password is correct
    } else if (event.password !== 'Password1234') {
      // Return HTTP status 200 but with no role in the response to indicate
 authentication failure
      response = {};
    }
  } else {
    // Return HTTP status 200 but with no role in the response to indicate
 authentication failure
    response = {};
  }
  callback(null, response);
};
```

Path-based home directory, NodeJS

以下 NodeJS 示例函数为拥有基于路径的主目录的用户提供了详细信息。

```
// GetUserConfig Lambda

exports.handler = (event, context, callback) => {
  console.log("Username:", event.username, "ServerId: ", event.serverId);

  var response;
  // Check if the username presented for authentication is correct. This doesn't
 check the value of the server ID, only that it is provided.
  // There is also event.protocol (one of "FTP", "FTPS", "SFTP") and event.sourceIp
 (e.g., "127.0.0.1") to further restrict logins.
  if (event.serverId !== "" && event.username == 'example-user') {
    response = {
      Role: 'arn:aws:iam::123456789012:role/transfer-access-role', // The user is
 authenticated if and only if the Role field is not blank
      Policy: '', // Optional, JSON stringified blob to further restrict this user's
 permissions
      HomeDirectory: '/fs-faa1a123' // Not required, defaults to '/'
    };

    // Check if password is provided
    if (!event.password) {
      // If no password provided, return the user's SSH public key
     response['PublicKeys'] = [ "ssh-
rsa abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789" ];
    // Check if password is correct
    } else if (event.password !== 'Password1234') {
      // Return HTTP status 200 but with no role in the response to indicate
 authentication failure
      response = {};
    }
  } else {
    // Return HTTP status 200 but with no role in the response to indicate
 authentication failure
    response = {};
  }
  callback(null, response);
};
```

Logical home directory, Python

以下 Python 示例函数为拥有逻辑主目录的用户提供了详细信息。

```
# GetUserConfig Python Lambda with LOGICAL HomeDirectoryDetails
```

```
import json

def lambda_handler(event, context):
  print("Username: {}, ServerId: {}".format(event['username'], event['serverId']))

  response = {}

  # Check if the username presented for authentication is correct. This doesn't
 check the value of the server ID, only that it is provided.
  if event['serverId'] != '' and event['username'] == 'example-user':
    homeDirectoryDetails = [
      {
        'Entry': '/',
        'Target': '/fs-faa1a123'
      }
    ]
    response = {
      'Role': 'arn:aws:iam::123456789012:role/transfer-access-role', # The user will
 be authenticated if and only if the Role field is not blank
      'PosixProfile': {"Gid": 65534, "Uid": 65534}, # Required for EFS access, but
 not needed for S3
      'HomeDirectoryDetails': json.dumps(homeDirectoryDetails),
      'HomeDirectoryType': "LOGICAL"
    }

    # Check if password is provided
    if event.get('password', '') == '':
      # If no password provided, return the user's SSH public key
     response['PublicKeys'] = [ "ssh-
rsa abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789" ]
    # Check if password is correct
    elif event['password'] != 'Password1234':
      # Return HTTP status 200 but with no role in the response to indicate
 authentication failure
      response = {}
  else:
    # Return HTTP status 200 but with no role in the response to indicate
 authentication failure
    response = {}

  return response
```

Path-based home directory, Python

以下 Python 示例函数为拥有基于路径的主目录的用户提供了详细信息。

```
# GetUserConfig Python Lambda with PATH HomeDirectory

def lambda_handler(event, context):
  print("Username: {}, ServerId: {}".format(event['username'], event['serverId']))

  response = {}

  # Check if the username presented for authentication is correct. This doesn't
 check the value of the server ID, only that it is provided.
  # There is also event.protocol (one of "FTP", "FTPS", "SFTP") and event.sourceIp
 (e.g., "127.0.0.1") to further restrict logins.
  if event['serverId'] != '' and event['username'] == 'example-user':
    response = {
      'Role': 'arn:aws:iam::123456789012:role/transfer-access-role', # The user will
 be authenticated if and only if the Role field is not blank
      'Policy': '', #  Optional, JSON stringified blob to further restrict this
 user's permissions
      'HomeDirectory': '/fs-fs-faa1a123',
      'HomeDirectoryType': "PATH" # Not strictly required, defaults to PATH
    }

    # Check if password is provided
    if event.get('password', '') == '':
      # If no password provided, return the user's SSH public key
     response['PublicKeys'] = [ "ssh-
rsa abcdef0123456789abcdef0123456789abcdef0123456789abcdef0123456789" ]
    # Check if password is correct
    elif event['password'] != 'Password1234':
      # Return HTTP status 200 but with no role in the response to indicate
 authentication failure
      response = {}
  else:
    # Return HTTP status 200 but with no role in the response to indicate
 authentication failure
    response = {}

  return response
```

测试您的配置

创建自定义身份提供程序后，应测试您的配置。

Console

使用 AWS Transfer Family 控制台测试您的配置

1. 打开AWS Transfer Family 控制台。
2. 在“服务器”页面上，选择您的新服务器，选择“操作”，然后选择“测试”。
3. 输入您在部署 AWS CloudFormation 堆栈时设置的用户名和密码文本。如果您保留默认选项，则用户名为 `myuser`，密码为 `MySuperSecretPassword`。
4. 如果在部署 AWS CloudFormation 堆栈时设置了源 IP 地址，请选择服务器协议并输入源 IP 地址。

CLI

使用 AWS CLI 测试您的配置

1. 运行 test-identity-provider 命令。如后续步骤所述，将 *user input placeholder* 用您自己的信息进行替换。

```
aws transfer test-identity-provider --server-id s-1234abcd5678efgh --user-name myuser --user-password MySuperSecretPassword --server-protocol FTP --source-ip 127.0.0.1
```

2. 输入服务器 ID。
3. 输入您在部署 AWS CloudFormation 堆栈时设置的用户名和密码。如果您保留默认选项，则用户名为 `myuser`，密码为 `MySuperSecretPassword`。
4. 如果在部署 AWS CloudFormation 堆栈时设置了服务器协议和源 IP 地址，请输入它们。

如果用户身份验证成功，则测试将返回 `StatusCode: 200` HTTP 响应、一个空字符串 `Message: ""`（否则将包含失败原因）和一个 `Response` 字段。

> **Note**
>
> 在下面的响应示例中，`Response` 字段是一个已经 “字符串化” 的 JSON 对象（转换为可在程序中使用的扁平 JSON 字符串），其中包含用户角色和权限的详细信息。

```
{
    "Response":"{\"Policy\":\"{\\\"Version\\\":\\\"2012-10-17\\\",\\\"Statement\\\":
[{\\\"Sid\\\":\\\"ReadAndListAllBuckets\\\",\\\"Effect\\\":\\\"Allow\\\",\\\"Action\\
\":[\\\"s3:ListAllMybuckets\\\",\\\"s3:GetBucketLocation\\\",\\\"s3:ListBucket\\\",\\
\"s3:GetObjectVersion\\\",\\\"s3:GetObjectVersion\\\"],\\\"Resource\\\":\\\"*\\\"}]}\",
\"Role\":\"arn:aws:iam::000000000000:role/MyUserS3AccessRole\",\"HomeDirectory\":\"/
\"}",
    "StatusCode": 200,
    "Message": ""
}
```

使用 Amazon API Gateway 整合您的身份提供程序

本主题介绍如何使用 AWS Lambda 函数支持 API Gateway 方法。如果您需要 RESTful API 来集成您的身份提供商，或者您想使用 AWS WAF 其功能来处理地理封锁或速率限制请求，请使用此选项。

使用 API Gateway 集成身份提供程序时的限制

- 此配置不支持自定义域。
- 此配置不支持私有 API Gateway 网址。

如果您需要其中任何一个，则可以使用 Lambda 作为身份提供程序，而无需使用 API Gateway。有关更多信息，请参阅 AWS Lambda 用于整合您的身份提供商。

使用 API Gateway 方法进行身份验证

您可以创建一个 API Gateway 方法，用作 Transfer Family 的身份提供程序。这种方法为您创建和提供 API 提供了一种高度安全的方式。使用 API Gateway，您可以创建 HTTPS 端点，以便以更高的安全性传输所有传入的 API 调用。有关 API Gateway 服务的更多详细信息，请参阅 API Gateway 开发者指南。

API Gateway 提供了一种名为的授权方法`AWS_IAM`，该方法为您提供与内部 AWS 使用的相同基于 AWS Identity and Access Management (IAM) 的身份验证。如果您通过 `AWS_IAM` 启用身份验证，则只有具有调用 API 的明确权限的调用程序才能访问该 API 的 API Gateway 方法。

要将您的 API Gateway 方法用作 Transfer Family 的自定义身份提供程序，请为您的 API Gateway 方法启用 IAM。在此过程中，您需要为一个 IAM 角色提供 Transfer Family 使用您的网关的权限。

> **Note**
>
> 为了提高安全性，可以配置 Web 应用程序防火墙。 AWS WAF 是一种 Web 应用程序防火墙，可让您监视转发到 Amazon API Gateway 的 HTTP 和 HTTPS 请求。有关更多信息，请参阅 是一个 Web 应用程序防火墙。。

使用您的 API Gateway 方法对 Transfer Family 进行自定义身份验证

1. 创建 AWS CloudFormation 堆栈。要实现此目的，应按照以下步骤进行：

   > **Note**
   >
   > 堆栈模板已更新为使用 Base64 编码的密码：有关详细信息，请参阅。对 AWS CloudFormation 模板的改进

   a. 打开 AWS CloudFormation 控制台，网址为 https://console.aws.amazon.com/cloudformation。

   b. 按照AWS CloudFormation 用户指南中的选择 AWS CloudFormation 堆栈模板中的使用现有模板部署堆栈的说明进行操作。

   c. 使用以下基本模板之一创建 AWS Lambda支持的 API Gateway 方法，以便在 Transfer Family 中用作自定义身份提供者。

      - 基本堆栈模板

        默认情况下，您的 API Gateway 方法用作自定义身份提供者，使用硬编码的 SSH（安全外壳）密钥或密码对单个服务器中的单个用户进行身份验证。部署后，您可以修改 Lambda 函数代码以执行不同的操作。

      - AWS Secrets Manager 堆栈模板

        默认情况下，您的 API Gateway 方法会根据格式 `aws/transfer/server-id/username` 的 Secrets Manager 中的条目进行身份验证。此外，该密钥必须包含返回给 Transfer Family 的所有用户属性的键值对。部署后，您可以修改 Lambda 函数代码以执行不同的操作。有关更多信息，请参阅博客文章启用密码身份验证以供 AWS Transfer Family 使用 AWS Secrets Manager。

      - Okta 堆栈模板

您的 API Gateway 方法与 Okta 集成，以作为 Transfer Family 中的自定义身份提供程序。有关更多信息，请参阅博客文章：使用 AWS Transfer Family将 Okta 用作身份提供程序。

部署其中一个堆栈是将自定义身份提供程序集成到 Transfer Family 工作流程的最简单方法。每个堆栈都使用 Lambda 函数来支持基于 API Gateway 的 API 方法。然后，您可以在 Transfer Family 中使用您的 API 方法作为自定义身份提供程序。默认情况下，Lambda 函数对使用 `MySuperSecretPassword` 密码 `myuser` 调用的单个用户进行身份验证。部署后，您可以编辑这些凭证或更新 Lambda 函数代码以执行不同的操作。

> **Important**
>
> 我们建议您编辑默认的用户和密码凭证。

部署堆栈后，您可以在 CloudFormation 控制台的 Outputs 选项卡上查看有关堆栈的详细信息。这些详细信息包括堆栈的 Amazon 资源名称 (ARN)、堆栈创建的 IAM 角色的 ARN 以及您的新网关的 URL。

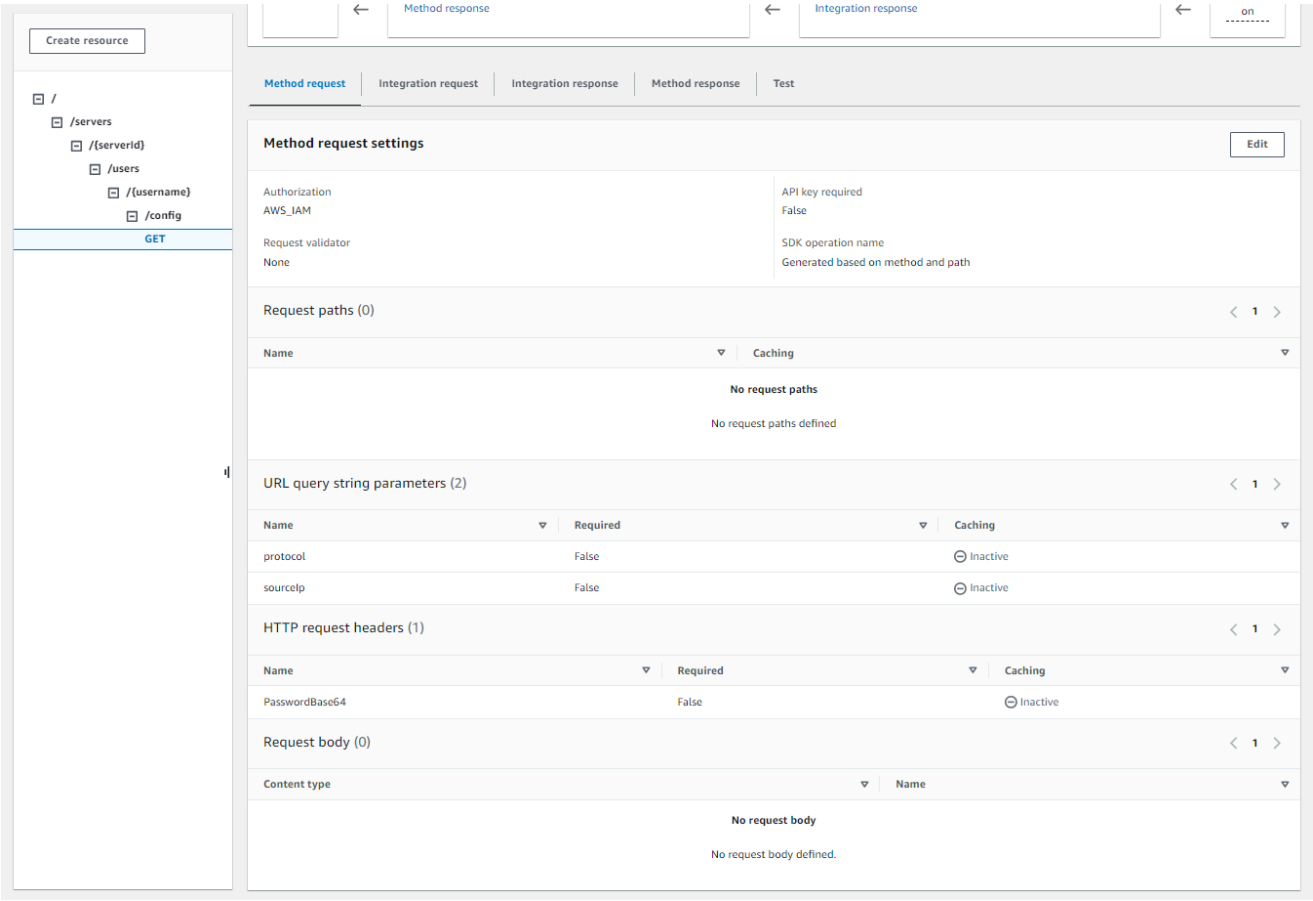
> **Note**
>
> 如果您使用自定义身份提供商选项为用户启用基于密码的身份验证，并且启用了 API Gateway 提供的请求和响应日志，API Gateway 会将用户的密码记录到您的 Amazon 日志中。 CloudWatch 我们建议不要在生产环境中使用此日志。有关更多信息，请参阅《CloudWatch API Gateway 开发者指南》中的 “在 API Gateway 中设置 API 日志”。

2. 检查您的服务器的 API Gateway 方法配置。要实现此目的，应按照以下步骤进行：

   a. 打开 API Gateway 控制台，网址为：https://console.aws.amazon.com/apigateway/。

   b. 选择模板生成的转移自定义身份提供商基本 AWS CloudFormation 模板 API。您可能需要选择您的区域才能看到您的网关。

   c. 在 “资源” 窗格中，选择 GET。以下屏幕截图显示了正确的方法配置。

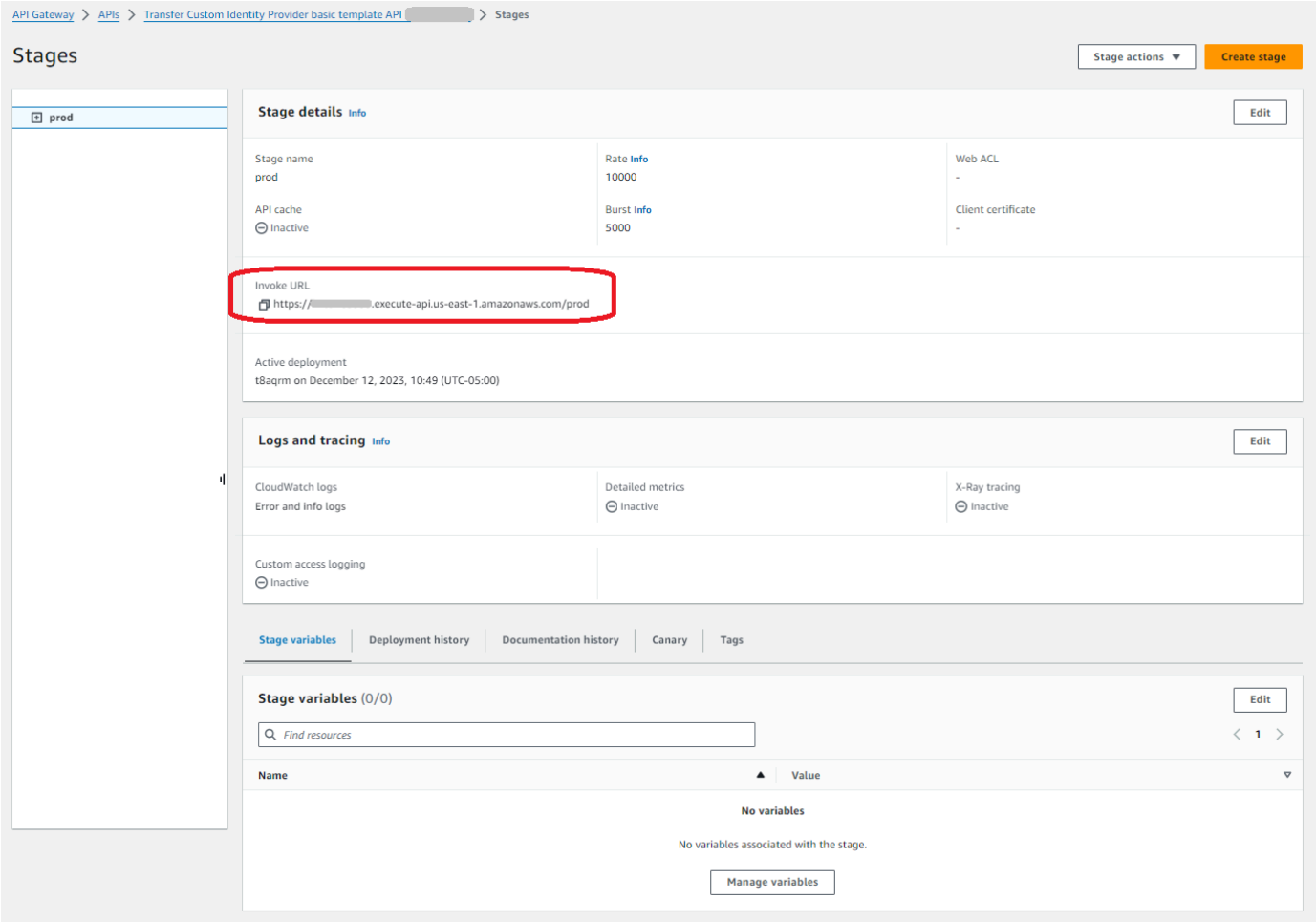


此时，您的 API Gateway 已准备好部署。

3. 在 操作，选择部署 API。对于部署阶段，选择 prod，然后选择部署。

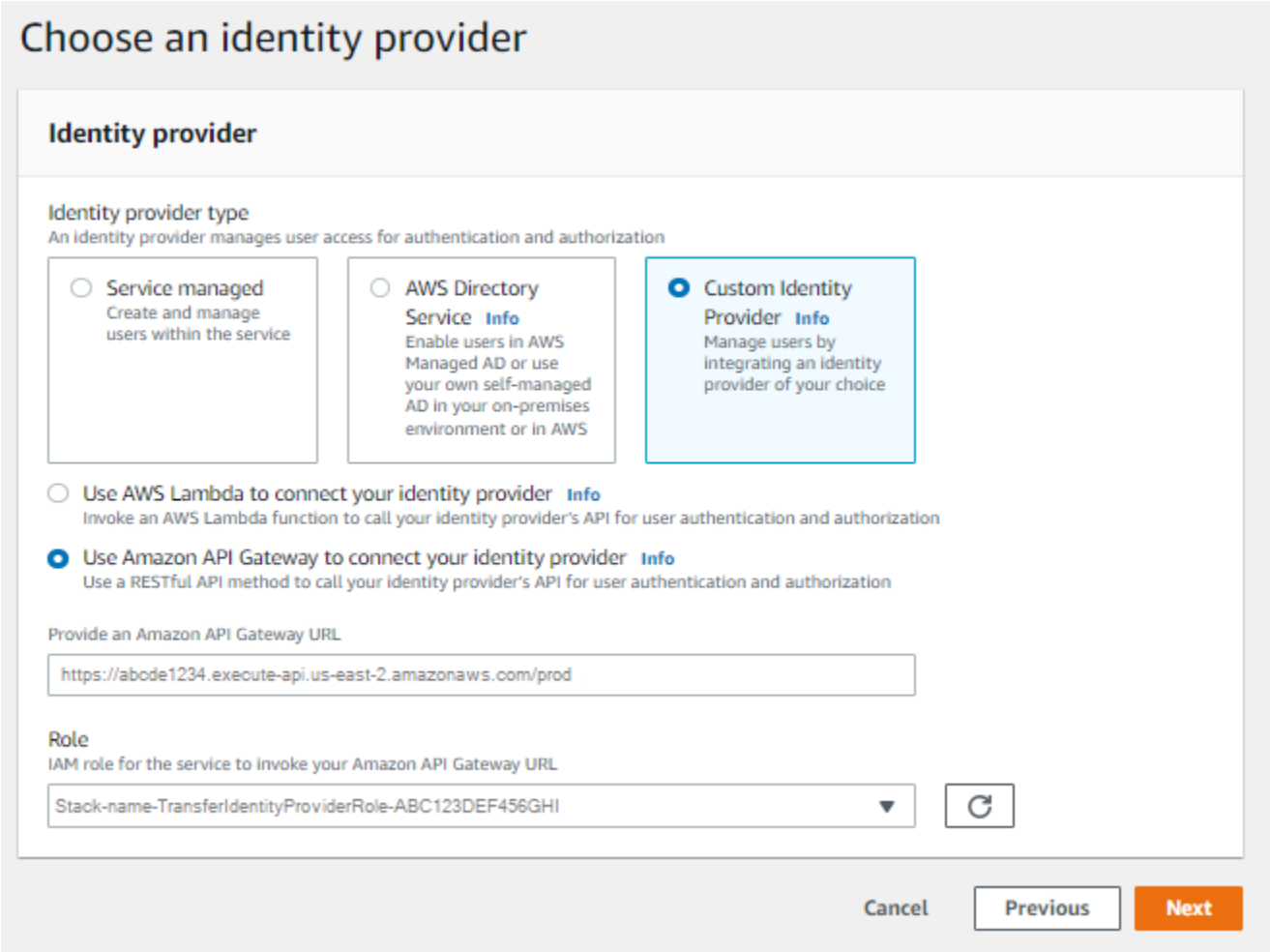
   成功部署 API Gateway 方法后，在 “阶段” > “阶段详情” 中查看其性能，如以下屏幕截图所示。

> **Note**
>
> 复制显示在屏幕顶部的调用 URL 地址。下一步可能需要它。



4. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。
5. 在你创建堆栈时，应该已经为你创建了 Transfer Family。如果不是，请使用以下步骤配置您的服务器。

   a. 选择“创建服务器”以打开“创建服务器”页面。在“选择身份提供程序”中，选择“自定义”，然后选择“使用 Amazon API Gateway 连接到您的身份提供程序”，如以下屏幕截图所示。



b. 在提供 Amazon API Gateway 网址文本框中，粘贴您在本过程的步骤 3 中创建的 API Gateway 端点的调用 URL 地址。

c. 对于角色，选择由 AWS CloudFormation 模板创建的 IAM 角色。此角色允许 Transfer Family 调用您的 API Gateway 方法。

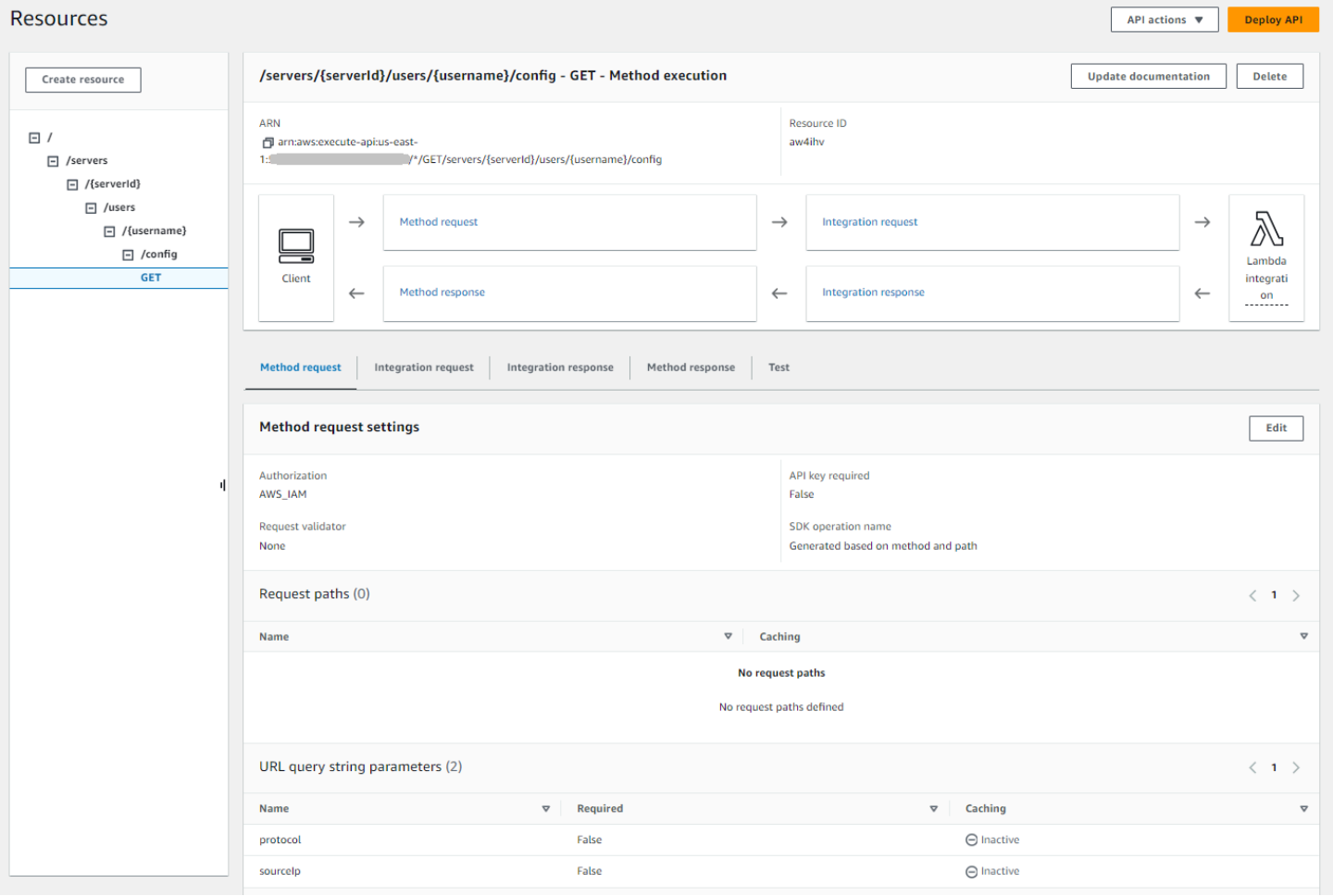
   调用角色包含您在步骤 1 中为创建的堆栈选择的堆栈名称。 AWS CloudFormation 格式如下：*CloudFormation-stack-name*-TransferIdentityProviderRole-*ABC123DEF456GHI*。

d. 填写其余的方框，然后选择“创建服务器”。有关创建服务器的其余步骤的详细信息，请参阅 配置 SFTP、FTPS 或 FTP 服务器端点。

实施您的 API Gateway 方法

要为 Transfer Family 创建自定义身份提供程序，您的 API Gateway 方法必须实现资源路径为 `/servers/`*serverId*`/users/`*username*`/config` 的单个方法。*serverId* 和 *username* 来自

RESTful 资源路径。此外，在方法请求中添加 `sourceIp` 和 `protocol` 作为 URL 查询字符串参数，如下图所示。



> **Note**
>
> 此用户名长度最少为 3 个字符，最多为 100 个字符。您可以在用户名中使用以下字符：a–z、A-Z、0–9、下划线 (_)、连字符 (-)、句点 (.) 和 at 符号 (@)。但是，用户名不能以连字符 (-)、句点 (.) 或 at 符号 (@) 开头。

如果 Transfer Family 代表您的用户尝试进行密码身份验证，则该服务会提供 `Password:` 标头字段。在没有 `Password:` 标头的情况下，Transfer Family 会尝试通过公钥身份验证来验证您的用户。

当您使用身份提供商对最终用户进行身份验证和授权时，除了验证他们的凭据外，您还可以根据最终用户使用的客户端 IP 地址来允许或拒绝访问请求。您可以使用此功能来确保存储在 S3 存储桶或 Amazon EFS 文件系统中的数据只能通过支持的协议从您指定为可信的 IP 地址进行访问。要启用此功能，必须在查询字符串中包含 `sourceIp`。

如果您为服务器启用了多个协议，并且想要通过多个协议使用相同的用户名提供访问权限，则只要在身份提供程序中设置了每个协议的特定凭据，就可以这样做。要启用此功能，必须在 RESTful 资源路径中包含 *protocol* 值。

您的 API Gateway 方法应始终返回 HTTP 状态码 200。任何其他 HTTP 状态代码则表示访问 API 时出错。

Amazon S3 示例响应

示例响应正文是适用于 Amazon S3 的以下格式的 JSON 文档。

```
{
 "Role": "IAM role with configured S3 permissions",
 "PublicKeys": [
     "ssh-rsa public-key1",
     "ssh-rsa public-key2"
  ],
 "Policy": "STS Assume role session policy",
 "HomeDirectory": "/bucketName/path/to/home/directory"
}
```

Note

策略会以 JSON 格式转义为字符串。例如：

```
"Policy":
"{
  \"Version\": \"2012-10-17\",
  \"Statement\":
     [
     {\"Condition\":
        {\"StringLike\":
            {\"s3:prefix\":
               [\"user/*\", \"user/\"]}},
     \"Resource\": \"arn:aws:s3:::bucket\",
     \"Action\": \"s3:ListBucket\",
     \"Effect\": \"Allow\",
     \"Sid\": \"ListHomeDir\"},
     {\"Resource\": \"arn:aws:s3:::*\",
        \"Action\": [\"s3:PutObject\",
        \"s3:GetObject\",
        \"s3:DeleteObjectVersion\",
```

```
        \"s3:DeleteObject\",
        \"s3:GetObjectVersion\",
        \"s3:GetObjectACL\",
        \"s3:PutObjectACL\"],
     \"Effect\": \"Allow\",
     \"Sid\": \"HomeDirObjectAccess\"}]
}"
```

以下示例响应会显示用户具有逻辑主目录类型。

```
{
   "Role": "arn:aws:iam::123456789012:role/transfer-access-role-s3",
   "HomeDirectoryType":"LOGICAL",
   "HomeDirectoryDetails":"[{\"Entry\":\"/\",\"Target\":\"/MY-HOME-BUCKET\"}]",
   "PublicKeys":[""]
}
```

Amazon EFS 示例响应

示例响应正文是 Amazon EFS 的以下格式的 JSON 文档。

```
{
 "Role": "IAM role with configured EFS permissions",
 "PublicKeys": [
     "ssh-rsa public-key1",
     "ssh-rsa public-key2"
  ],
 "PosixProfile": {
   "Uid": "POSIX user ID",
   "Gid": "POSIX group ID",
   "SecondaryGids": [Optional list of secondary Group IDs],
 },
 "HomeDirectory": "/fs-id/path/to/home/directory"
}
```

Role 字段表示身份验证成功。在进行密码身份验证时（当您提供 Password: 标头时），您无需提供 SSH 公钥。如果无法对用户进行身份验证，例如，如果密码不正确，则您的方法应返回未设置 Role 的响应。此类响应的一个例子是空的 JSON 对象。

以下示例响应显示了具有逻辑主目录类型的用户。

```
{
    "Role": "arn:aws:iam::123456789012:role/transfer-access-role-efs",
    "HomeDirectoryType": "LOGICAL",
    "HomeDirectoryDetails":"[{\"Entry\":\"/\",\"Target\":\"/faa1a123\"}]",
    "PublicKeys":[""],
    "PosixProfile":{"Uid":65534,"Gid":65534}
}
```

您可以在 JSON 格式的 Lambda 函数中包含用户策略。有关在 Transfer Family 中配置用户策略的更多信息，请参阅 管理访问控制。

默认 Lambda 函数

要实施不同的身份验证策略，请编辑您的网关使用的 Lambda 函数。为了帮助您满足应用程序的需求，您可以在 Node.js 中使用以下示例 Lambda 函数。有关更多信息， AWS Lambda 开发人员指南 或 通过 Node.js 构建 Lambda 函数。

以下示例 Lambda 函数使用您的用户名、密码（如果您正在执行密码身份验证）、服务器 ID、协议和客户端 IP 地址。您可以使用这些输入的组合来查找您的身份提供程序并确定是否应接受登录。

> **Note**
>
> 如果您为服务器启用了多个协议，并且想要通过多个协议使用相同的用户名提供访问权限，则只要在身份提供程序中设置了相关协议的特定凭据，就可以这样做。
> 对于文件传输协议 (FTP)，我们建议为 Secure Shell (SSH) 文件传输协议 (SFTP) 和 SSL (FTPS) 文件传输协议设置不同的凭证。我们建议为 FTP 保留单独的凭据，因为与 SFTP 和 FTPS 不同，FTP 以明文形式传输凭据。通过将 FTP 凭证与 SFTP 或 FTPS 隔离开来，如果共享或公开 FTP 凭证，则使用 SFTP 或 FTPS 的工作负载会保持安全。

此示例函数会返回角色和逻辑主目录详细信息以及公钥（如果它执行公钥身份验证）。

创建服务托管用户时，可以设置他们的主目录，无论是逻辑目录还是物理目录均是如此。同样，我们需要 Lambda 函数的结果来传达所需的用户物理或逻辑目录结构。您设置的参数取决于该 HomeDirectoryType 字段的值。

- HomeDirectoryType 设置为 PATH — 然后，HomeDirectory 字段必须是您的用户可见的 Amazon S3 存储桶绝对前缀或 Amazon EFS 绝对路径。

- HomeDirectoryType 设置为 LOGICAL — 请不要设置 HomeDirectory 字段。相反，我们设置了一个提供所需入口/目标映射的 HomeDirectoryDetails 字段，类似于服务托管用户的 HomeDirectoryDetails 参数中描述的值。

Lambda 函数示例 中列出了示例函数。

用于的 Lambda 函数 AWS Secrets Manager

要 AWS Secrets Manager 用作您的身份提供商，您可以使用示例 AWS CloudFormation 模板中的 Lambda 函数。Lambda 函数使用您的凭证查询 Secrets Manager 服务，如果成功，则会返回指定的密钥。有关 Secrets Manager 的更多信息，请参阅《AWS Secrets Manager 用户指南》https://docs.aws.amazon.com/secretsmanager/latest/userguide/intro.html。

要下载使用此 Lambda 函数的示例 AWS CloudFormation 模板，请访问提供的 Amazon S3 存储桶。AWS Transfer Family

对 AWS CloudFormation 模板的改进

已发布的 CloudFormation模板已对 API Gateway 界面进行了改进。现在，这些模板在 API Gateway 中使用 Base64 编码的密码。如果没有此增强功能，您的现有部署可以继续运行，但不允许使用基本 US-ASCII 字符集之外的字符的密码。

启用此功能的模板更改如下：

- GetUserConfigRequest AWS::ApiGateway::Method资源必须有这个RequestTemplates代码（斜体行是更新的行）

```
RequestTemplates:
   application/json: |
   {
      "username": "$util.urlDecode($input.params('username'))",
      "password":
 "$util.escapeJavaScript($util.base64Decode($input.params('PasswordBase64'))).replaceAll("\
\'","'")",
      "protocol": "$input.params('protocol')",
      "serverId": "$input.params('serverId')",
      "sourceIp": "$input.params('sourceIp')"
}
```

- GetUserConfig资源必须更改RequestParameters为使用PasswordBase64标题（斜体行是更新的行）：

```
RequestParameters:
   method.request.header.PasswordBase64: false
   method.request.querystring.protocol: false
   method.request.querystring.sourceIp: false
```

检查堆栈的模板是否是最新的

1. 打开 AWS CloudFormation 控制台，网址为 https://console.aws.amazon.com/cloudformation。
2. 从堆栈列表中选择您的堆栈。
3. 在详细信息面板中，选择模板选项卡。
4. 寻找以下内容：

   - 搜索RequestTemplates并确保你有以下行：

     ```
     "password":
      "$util.escapeJavaScript($util.base64Decode($input.params('PasswordBase64'))).replaceAll(
     \'","'")",
     ```

   - 搜索RequestParameters并确保你有以下行：

     ```
     method.request.header.PasswordBase64: false
     ```

如果您没有看到更新的行，请编辑您的堆栈。有关如何更新 AWS CloudFormation 堆栈的详细信息，请参阅《用户指南》中的AWS CloudFormation修改堆栈模板。

使用逻辑目录简化您的 Transfer Family 目录结构

要简化 AWS Transfer Family 服务器目录结构，可以使用逻辑目录。使用逻辑目录，可以构造一个使用用户友好名称的虚拟目录结构，当用户连接到 Amazon S3 存储桶或 Amazon EFS 文件系统时，可以导航这些名称。使用逻辑目录时，可以避免向最终用户泄露绝对目录路径、Amazon S3 存储桶名称和 EFS 文件系统名称。

> **Note**
>
> 您应该使用会话策略，以便您的最终用户只能执行您允许他们执行的操作。

您应该使用逻辑目录为最终用户创建用户友好的虚拟目录，并抽象存储桶名称。逻辑目录映射仅允许用户访问其指定的逻辑路径和子目录，禁止使用遍历逻辑根目录的相对路径。
Transfer Family 会验证可能包含相对元素的每条路径，并在我们将这些路径传递给 Amazon S3 之前主动阻止解析这些路径；这样可以防止您的用户超越其逻辑映射。
尽管 Transfer Family 会阻止您的最终用户访问其逻辑目录之外的目录，但我们也建议您使用唯一的角色或会话策略在存储级别强制执行最低权限。

通过执行所谓的chroot操作，您可以使用逻辑目录将用户的根目录设置为存储层次结构中的所需位置。在此模式下，用户无法导航到您为其配置的主目录或根目录之外的目录。

例如，尽管 Amazon S3 用户的范围已缩小为仅限访问/*mybucket*/*home*/${transfer:*UserName*}，但有些客户端允许用户向上遍历文件夹至/*mybucket*/*home*。在这种情况下，用户只有在注销并再次登录 Transfer Family 服务器后才会重新进入其预期的主目录。执行chroot操作可以防止这种情况的发生。

您可以跨存储桶和前缀创建自己的目录结构。如果您的工作流程需要一个无法通过存储桶前缀复制的特定目录结构，则此功能适用。您还可以链接到 Amazon S3 中的多个非连续位置，类似于在 Linux 文件系统中创建符号链接，其中您的目录路径引用文件系统中的不同位置。

逻辑目录文件映射

HomeDirectoryMapEntry数据类型现在包括一个Type参数。在此参数存在之前，您可能已经创建了以文件为目标的逻辑目录映射。如果您之前创建过任何此类逻辑目录映射，则必须将显式设置Type为FILE，否则这些映射将无法正常运行。
一种方法是调用 UpdateUser API，并将现有映射TypeFILE的设置为。

使用逻辑目录的规则

在构建逻辑目录映射之前，应了解以下规则：

- 如果Entry是"/"，则只能有一个映射，因为不允许重叠路径。
- 逻辑目录支持最大 2.1 MB 的映射（对于服务管理用户，此限制为 2,000 个条目）。也就是说，包含映射的数据结构的最大大小为 2.1 MB。如果您有很多映射，则可以按如下方式计算映射的大小：
  1. 用格式写出一个典型的映射{"Entry":"/*entry-path*","Target":"/*target-path*"}，其中*entry-path*和*target-path*是你将要使用的实际值。

2. 计算该字符串中的字符，然后添加一 (1)。
3. 将该数字乘以服务器的近似映射数。

如果您在步骤 3 中估计的数字小于 2.1 MB，则您的映射在可接受的限制范围内。

- 如果存储桶或文件系统路径已根据用户名参数化，则目标可以使用该`${transfer:UserName}`变量。
- 目标可以是不同存储桶或文件系统中的路径，但您必须确保映射的 AWS Identity and Access Management (IAM) 角色（响应中的`Role`参数）提供对这些存储桶或文件系统的访问权限。
- 不要指定`HomeDirectory`参数，因为当你使用`HomeDirectoryType`参数的值时，`EntryTarget`成对会暗示这个`LOGICAL`值。
- 目标必须以正斜杠 (/) 字符开头，但在指定时不要使用尾部的正斜杠 (/)。`Target`例如，`/DOC-EXAMPLE-BUCKET/images`可以接受`DOC-EXAMPLE-BUCKET/images`，但不`/DOC-EXAMPLE-BUCKET/images/`是。
- Amazon S3 是一个对象存储，这意味着文件夹是一个虚拟概念，没有实际的目录层次结构。如果您的应用程序从客户端发出`stat`操作，则当您使用 Amazon S3 进行存储时，所有内容都将归类为文件。亚马逊简单存储服务用户指南中的使用文件夹在 Amazon S3 控制台中组织对象中描述了此行为。如果您的应用程序要求`stat`准确显示某物是文件还是文件夹，则可以使用亚马逊弹性文件系统（Amazon EFS）作为 Transfer Family 服务器的存储选项。
- 如果您为用户指定逻辑目录值，则使用的参数取决于用户的类型：
  - 对于服务托管的用户，请在`HomeDirectoryMappings`中提供逻辑目录值。
  - 对于自定义身份提供商用户，请在中提供逻辑目录值`HomeDirectoryDetails`。

> **Important**
>
> 除非您选择优化 Amazon S3 目录的性能（创建或更新服务器时），否则根目录必须在启动时存在。对于 Amazon S3，这意味着您必须已经创建了一个以正斜杠 (/) 结尾的零字节对象才能创建根文件夹。避免这个问题是考虑优化 Amazon S3 性能的理由。

实现逻辑目录和 chroot

要使用逻辑目录和chroot功能，您必须执行以下操作：

为每个用户开启逻辑目录。为此，请在创建或更新用户时将`HomeDirectoryType`参数设置为`LOGICAL`。

```
"HomeDirectoryType": "LOGICAL"
```

chroot

对于chroot，创建一个由每个用户的单个Entry和Target配对组成的目录结构。根文件夹是Entry重点，Target是存储桶或文件系统中要映射到的位置。

Example for Amazon S3

```
[{"Entry": "/", "Target": "/mybucket/jane"}]
```

Example for Amazon EFS

```
[{"Entry": "/", "Target": "/fs-faa1a123/jane"}]
```

您可以像前面的示例一样使用绝对路径，也可以使用${transfer:UserName}动态替换用户名，如下例所示。

```
[{"Entry": "/", "Target":
"/mybucket/${transfer:UserName}"}]
```

在前面的示例中，用户被锁定到其根目录，且无法在层次结构中向上移动。

虚拟目录结构

对于虚拟目录结构，只要用户的 IAM 角色映射有权访问它们，您就可以创建多个EntryTarget配对，目标位于您的 S3 存储桶或 EFS 文件系统的任意位置，包括跨多个存储桶或文件系统。

在以下虚拟结构示例中，当用户登录 AWS SFTP 时，他们位于根目录中，子目录为/pics、/doc/reporting、和。/anotherpath/subpath/financials

> **Note**
>
> 除非您选择优化 Amazon S3 目录的性能（当您创建或更新服务器时），否则如果目录尚不存在，则用户或管理员需要创建这些目录。避免这个问题是考虑优化 Amazon S3 性能的理由。对于 Amazon EFS，您仍然需要管理员来创建逻辑映射或/目录。

```
[
```

```
{"Entry": "/pics", "Target": "/bucket1/pics"},
{"Entry": "/doc", "Target": "/bucket1/anotherpath/docs"},
{"Entry": "/reporting", "Target": "/reportingbucket/Q1"},
{"Entry": "/anotherpath/subpath/financials", "Target": "/reportingbucket/financials"}]
```

Note

您只可以将文件上传到映射的特定文件夹。这意味着，在前面的示例中，您不能上传到/anotherpath或anotherpath/subpath目录；只有anotherpath/subpath/financials。您也无法直接映射到这些路径，因为不允许重叠路径。
例如，假设您创建以下映射：

```
{
   "Entry": "/pics",
   "Target": "/mybucket/pics"
},
{
   "Entry": "/doc",
   "Target": "/mybucket/mydocs"
},
{
   "Entry": "/temp",
   "Target": "/mybucket"
}
```

您只可以将文件上传到这些存储桶中。当您首次通过sftp连接时，您会被放到根目录中/。如果您尝试将文件上传到该目录，则上传将失败。以下命令显示了一个示例序列：

```
sftp> pwd
Remote working directory: /
sftp> put file
Uploading file to /file
remote open("/file"): No such file or directory
```

要上传到任何directory/sub-directory，必须将路径明确映射到sub-directory。

有关为用户配置逻辑目录的更多信息（包括可供下载和chroot使用的 AWS CloudFormation 模板），请参阅 AWS 存储博客中的使用 chroot 和逻辑目录简化 AWS SFTP 结构。

配置逻辑目录示例

在此示例中，我们创建一个用户并分配两个逻辑目录。以下命令使用逻辑目录pics和doc创建新用户（适用于现有的 Transfer Family 服务器）。

```
aws transfer create-user --user-name marymajor-logical --server-id s-11112222333344445
 --role arn:aws:iam::1234abcd5678:role/marymajor-role --home-directory-type LOGICAL \
 --home-directory-mappings "[{\"Entry\":\"/pics\", \"Target\":\"/DOC-EXAMPLE-BUCKET1/
pics\"}, {\"Entry\":\"/doc\", \"Target\":\"/DOC-EXAMPLE-BUCKET2/test/mydocs\"}]" \
 --ssh-public-key-body file://~/.ssh/id_rsa.pub
```

如果**marymajor**是现有用户并且她的主目录类型是PATH，则您可以使用与前一个命令类似的命令将其更改为LOGICAL。

```
aws transfer update-user --user-name marymajor-logical \
 --server-id s-11112222333344445 --role arn:aws:iam::1234abcd5678:role/marymajor-role \
 --home-directory-type LOGICAL --home-directory-mappings "[{\"Entry\":\"/pics\",
 \"Target\":\"/DOC-EXAMPLE-BUCKET1/pics\"}, \
 {\"Entry\":\"/doc\", \"Target\":\"/DOC-EXAMPLE-BUCKET2/test/mydocs\"}]"
```

请注意以下几点：

- 如果目录/*DOC-EXAMPLE-BUCKET1*/pics和/*DOC-EXAMPLE-BUCKET2*/test/mydocs尚未存在，则用户（或管理员）需要创建这些目录。
- 当**marymajor**连接到服务器并运行`ls -l`命令时，她会看到以下内容：

  ```
  drwxr--r--   1        -        -        0 Mar 17 15:42 doc
  drwxr--r--   1        -        -        0 Mar 17 16:04 pics
  ```

- **marymajor**无法在此级别创建任何文件或目录。但是，在pics和doc中，她可以添加子目录。
- 她添加到pics和doc的文件分别添加到 AmazonS3 路径/*DOC-EXAMPLE-BUCKET1*/pics和/*DOC-EXAMPLE-BUCKET2*/test/mydocs。
- 在此示例中，我们指定两个不同的存储桶来说明这种可能性。但是，您可以将同一个存储桶用于为用户指定的多个或所有逻辑目录。

为 Amazon EFS 配置逻辑目录

如果 Transfer Family 服务器使用 Amazon EFS，则必须先创建具有读写访问权限的用户主目录，然后用户才能在其逻辑主目录中工作。用户无法自己创建此目录，因为他们将缺乏mkdir对逻辑主目录的权限。

如果用户的主目录不存在，并且他们运行了ls命令，则系统会按如下方式做出响应：

```
sftp> ls
remote readdir ("/"): No such file or directory
```

对父目录具有管理访问权限的用户需要创建该用户的逻辑主目录。

自定义 AWS Lambda 响应

您可以将逻辑目录与连接到自定义身份提供商的 Lambda 函数一起使用。为此，在 Lambda 函数中，将HomeDirectoryType指定为**LOGICAL**，并为 HomeDirectoryDetails 参数添加Entry和Target值。例如：

```
HomeDirectoryType: "LOGICAL"
HomeDirectoryDetails: "[{\"Entry\": \"/\", \"Target\": \"/DOC-EXAMPLE-BUCKET/
theRealFolder"}]"
```

以下代码是来自自定义 Lambda 身份验证调用的成功响应示例。

```
aws transfer test-identity-provider --server-id s-1234567890abcdef0 --user-name myuser
{
    "Url": "https://a1b2c3d4e5.execute-api.us-east-2.amazonaws.com/prod/servers/
s-1234567890abcdef0/users/myuser/config",
    "Message": "",
    "Response": "{\"Role\": \"arn:aws:iam::123456789012:role/bob-usa-role\",
\"HomeDirectoryType\": \"LOGICAL\",\"HomeDirectoryDetails\": \"[{\\\"Entry\\\":\\\"/
myhome\\\",\\\"Target\\\":\\\"/DOC-EXAMPLE-BUCKET/theRealFolder\\\"}]\",\"PublicKeys\":
 \"[ssh-rsa myrsapubkey]\"}",
    "StatusCode": 200
}
```

> **Note**
>
> 仅当您使用 API Gateway 方法作为自定义身份提供商时，才会返回该"Url":行。

AWS Transfer Family SFTP 连接器

AWS Transfer Family SFTP 连接器使用 SFTP 协议建立一种关系，用于在 Amazon 存储和外部合作伙伴之间发送文件和消息。您可以将文件从 Amazon S3 发送到合作伙伴拥有的外部目的地。您也可以使用 SFTP 连接器从合作伙伴的 SFTP 服务器检索文件。

> **Note**
>
> 当前，SFTP 连接器只能用于连接到提供互联网访问端点的远程 SFTP 服务器。

请查看 AWS Transfer Family SFTP 连接器，了解 Transfer Family SFTP 连接器的简要介绍。

主题

- 配置 SFTP 连接器
- 使用 SFTP 连接器发送和检索文件
- 管理 SFTP 连接器

配置 SFTP 连接器

本主题介绍如何创建 SFTP 连接器、与之关联的安全算法、如何存储用于保存凭据的密钥、有关格式化私钥的详细信息以及测试连接器的说明。

主题

- 创建 SFTP 连接器
- 存储密钥以与 SFTP 连接器配合使用
- 生成并格式化 SFTP 连接器私钥
- 测试 SFTP 连接器

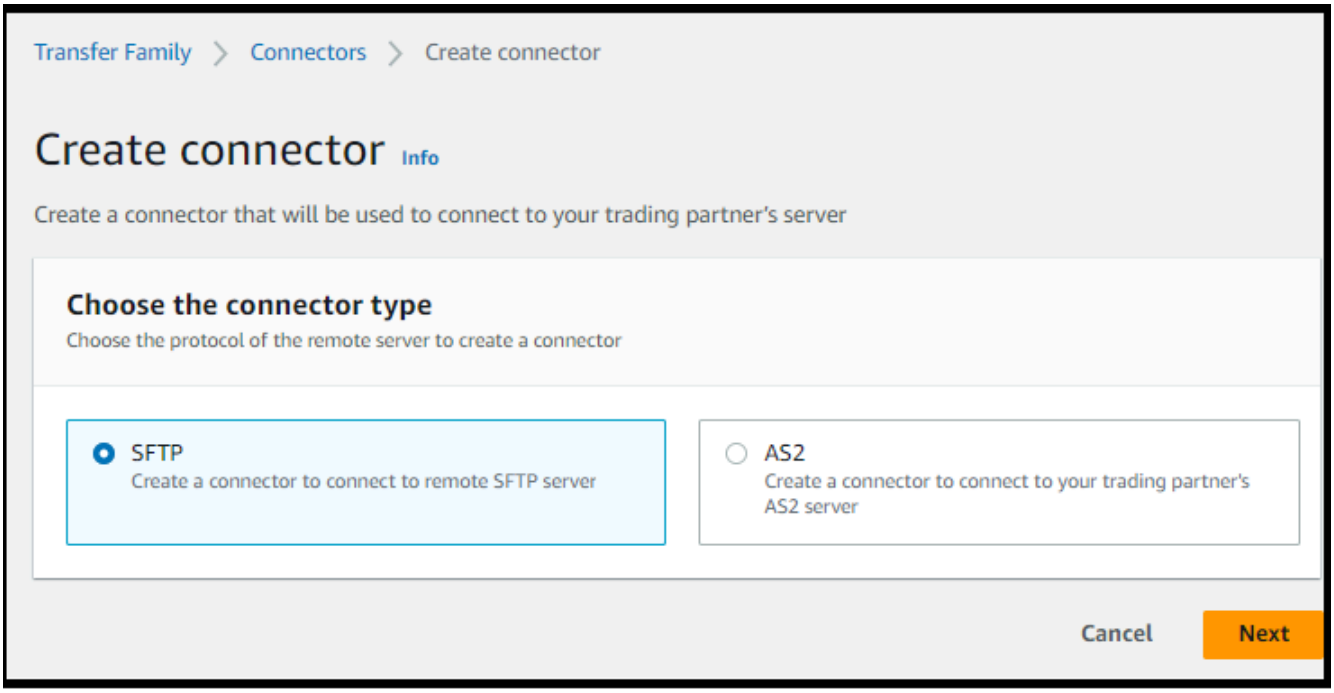
创建 SFTP 连接器

此过程说明如何使用 AWS Transfer Family 控制台或 AWS CLI创建 SFTP 连接器。

Console

若要创建 SFTP 连接器

1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在导航窗格中，选择连接，然后选择创建连接器。
3. 选择 SFTP 作为连接器类型，以创建 SFTP 连接器，然后选择“下一步”。



4. 在连接器配置部分中，提供以下信息：

   - 在 URL 中，输入远程 SFTP 服务器的 URL。例如 `sftp://AnyCompany.com`，此 URL 的格式必须为 `sftp://`*`partner-SFTP-server-url`*。

     > **Note**
     >
     > （可选）您可以在 URL 中提供端口号。格式为 `sftp://`*`partner-SFTP-server-url`*`:`*`port-number`*。默认端口号（未指定端口时）为端口 22。

   - 对于访问角色，请选择要使用的 (IAM) 角色的 Amazon 资源名称 AWS Identity and Access Management (ARN)。
     - 确保此角色提供对 `StartFileTransfer` 请求中所使用文件位置父目录提供读取和写入权限。
     - 请确保此角色为 **`secretsmanager:GetSecretValue`** 提供访问密钥的权限。

> **Note**
>
> 在策略中，您必须为密钥指定 ARN。ARN 包含机密名称，但在名称后面附加了六个随机的字母数字字符。密钥的 ARN 格式如下。
>
> ```
> arn:aws:secretsmanager:region:account-id:secret:aws/
> transfer/SecretName-6RandomCharacters
> ```

- 此角色还应包含一个信任关系，从而允许服务器在为用户的传输请求提供服务时访问您的资源。有关建立信任关系的详细信息，请参阅 建立信任关系。

以下示例授予访问*### S3 ## DOC-EXAMPLE-BUCK* ET 以及存储在 Secrets Manager 中的指定密钥所需的权限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
        "Sid": "AllowListingOfUserFolder",
        "Action": [
            "s3:ListBucket",
            "s3:GetBucketLocation"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
        ]
    },
    {
        "Sid": "HomeDirObjectAccess",
        "Effect": "Allow",
        "Action": [
            "s3:PutObject",
            "s3:GetObject",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion",
            "s3:GetObjectVersion",
            "s3:GetObjectACL",
            "s3:PutObjectACL"
        ],
        "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
```

```
    },
    {
        "Sid": "GetConnectorSecretValue",
        "Effect": "Allow",
        "Action": [
            "secretsmanager:GetSecretValue"
        ],
        "Resource": "arn:aws:secretsmanager:region:account-id:secret:aws/
transfer/SecretName-6RandomCharacters"
    }
  ]
}
```

> **Note**
>
> 对于访问角色，该示例授予对单个密钥的访问权限。但是，您可以使用通配符，如果您想为多个用户和密钥重复使用相同的 IAM 角色，这样可以节省工作量。例如，以下资源语句为名称以 aws/transfer 开头的所有密钥授予权限。
>
> ```
> "Resource": "arn:aws:secretsmanager:region:account-id:secret:aws/
> transfer/*"
> ```
>
> 您也可以将包含您的 SFTP 凭据的密钥存储在另一个 AWS 账户中。有关启用跨账户秘密访问的详细信息，请参阅其他账户中用户的 AWS Secrets Manager 密钥权限。

- （可选）对于日志记录角色，选择连接器用于将事件推送到 CloudWatch 日志的 IAM 角色。以下示例策略列出了记录 SFTP 连接器事件的必要权限。

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "SFTPConnectorPermissions",
        "Effect": "Allow",
        "Action": [
            "logs:CreateLogStream",
            "logs:DescribeLogStreams",
            "logs:CreateLogGroup",
            "logs:PutLogEvents"
        ],
        "Resource": [
            "arn:aws:logs:*:*:log-group:/aws/transfer/*"
```

```
        ]
    }]
}
```

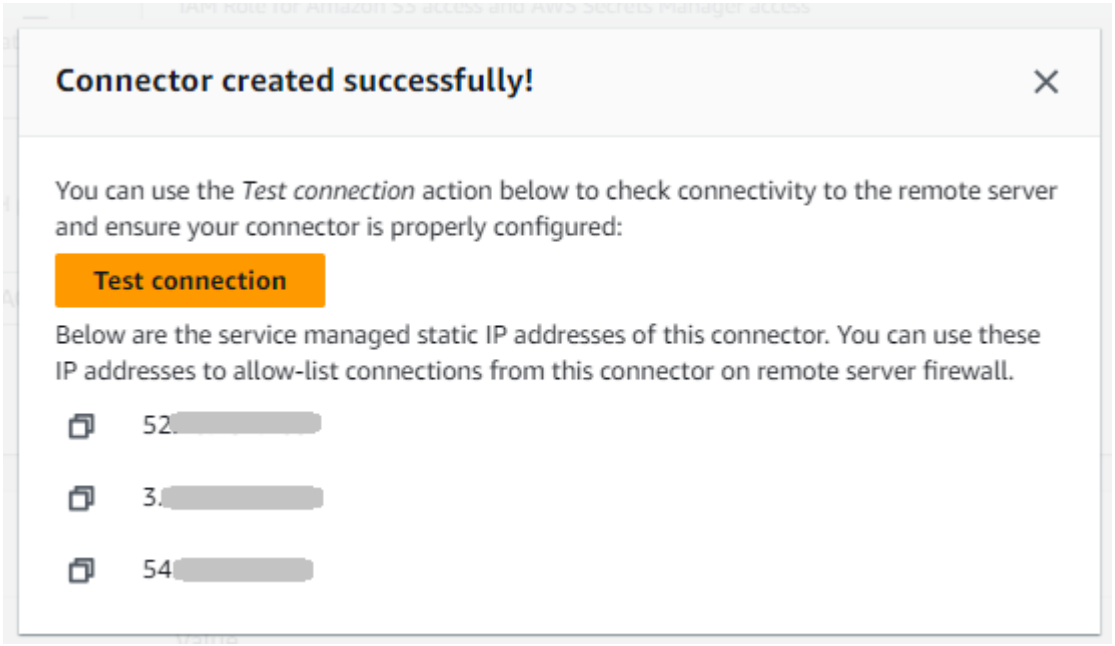
5. 在 SFTP Configuration 面板中提供以下信息：

   - 对于 Connector 凭据，从下拉列表中选择包含 SFTP 用户私钥或密码的密钥的名称。 AWS Secrets Manager 您必须创建密钥并以特定方式存储它。有关更多信息，请参阅 存储密钥以与 SFTP 连接器配合使用。
   - 对于受信任的主机密钥，请粘贴用于标识外部服务器的主机密钥的公共部分。您可以添加多个密钥，方法是选择“添加可信主机密钥”来添加其他密钥。您可以对 SFTP 服务器使用 `ssh-keyscan` 命令以检索必要的密钥。有关 Transfer Family 支持的受信任主机密钥的格式和类型的详细信息，请参阅 SFTPConnectorConfig。

6. 在 “加密算法选项” 部分，从 “安全策略” 字段的下拉列表中选择一个安全策略。安全策略允许您选择连接器支持的加密算法。有关可用安全策略和算法的详细信息，请参阅AWS Transfer Family SFTP 连接器的安全策略。

7. （可选）对于标签 部分的 键 和 值，以键/值对格式输入一个或多个标签。

8. 确认所有设置后，选择创建连接器以创建 SFTP 连接器。如果成功创建了连接器，则会出现一个屏幕，其中包含分配的静态 IP 地址列表和测试连接按钮。使用按钮测试新连接器的配置。



“连接器”页面会出现，其中新 SFTP 连接器的 ID 已添加到列表中。要查看连接器的详细信息，请参阅 查看 SFTP 连接器详细信息。

CLI

可使用 create-connector 命令创建连接器。要使用此命令创建 SFTP 连接器，必须提供以下信息。

- 远程 SFTP 服务器的 URL。例如 `sftp://AnyCompany.com`，此 URL 的格式必须为 `sftp://partner-SFTP-server-url`。
- 访问角色。选择 AWS Identity and Access Management IAM 角色的 Amazon 资源名称（ARN）。
  - 确保此角色提供对 `StartFileTransfer` 请求中所使用文件位置父目录提供读取和写入权限。
  - 请确保此角色为 **`secretsmanager:GetSecretValue`** 提供访问密钥的权限。

    > **Note**
    >
    > 在策略中，您必须为密钥指定 ARN。ARN 包含机密名称，但在名称后面附加了六个随机的字母数字字符。密钥的 ARN 格式如下。
    >
    > ```
    > arn:aws:secretsmanager:region:account-id:secret:aws/
    > transfer/SecretName-6RandomCharacters
    > ```

  - 此角色还应包含一个信任关系，从而允许服务器在为用户的传输请求提供服务时访问您的资源。有关建立信任关系的详细信息，请参阅 建立信任关系。

以下示例授予访问*### S3 ## DOC-EXAMPLE-BUCK* ET 以及存储在 Secrets Manager 中的指定密钥所需的权限。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
        "Sid": "AllowListingOfUserFolder",
        "Action": [
            "s3:ListBucket",
            "s3:GetBucketLocation"
        ],
        "Effect": "Allow",
        "Resource": [
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
        ]
    },
```

```
    {
        "Sid": "HomeDirObjectAccess",
        "Effect": "Allow",
        "Action": [
            "s3:PutObject",
            "s3:GetObject",
            "s3:DeleteObject",
            "s3:DeleteObjectVersion",
            "s3:GetObjectVersion",
            "s3:GetObjectACL",
            "s3:PutObjectACL"
        ],
        "Resource": "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
    },
    {
        "Sid": "GetConnectorSecretValue",
        "Effect": "Allow",
        "Action": [
            "secretsmanager:GetSecretValue"
        ],
        "Resource": "arn:aws:secretsmanager:region:account-id:secret:aws/
transfer/SecretName-6RandomCharacters"
    }
  ]
}
```

> **Note**
>
> 对于访问角色，该示例授予对单个密钥的访问权限。但是，您可以使用通配符，如果您想为多个用户和密钥重复使用相同的 IAM 角色，这样可以节省工作量。例如，以下资源语句为名称以 aws/transfer 开头的所有密钥授予权限。
>
> ```
> "Resource": "arn:aws:secretsmanager:region:account-id:secret:aws/transfer/
> *"
> ```
>
> 您也可以将包含您的 SFTP 凭据的密钥存储在另一个 AWS 账户中。有关启用跨账户秘密访问的详细信息，请参阅其他账户中用户的 AWS Secrets Manager 密钥权限。

- （可选）为连接器选择用于将事件推送到 CloudWatch 日志的 IAM 角色。以下示例策略列出了记录 SFTP 连接器事件的必要权限。

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "SFTPConnectorPermissions",
        "Effect": "Allow",
        "Action": [
            "logs:CreateLogStream",
            "logs:DescribeLogStreams",
            "logs:CreateLogGroup",
            "logs:PutLogEvents"
        ],
        "Resource": [
            "arn:aws:logs:*:*:log-group:/aws/transfer/*"
        ]
    }]
}
```

- 提供以下 SFTP 配置信息。
  - 中包含 SFTP 用户的私钥或密码 AWS Secrets Manager 的密钥的 ARN。
  - 用于识别外部服务器的主机密钥的公共部分。如果您愿意，可以提供多个可信的主机密钥。

  提供 SFTP 信息的最简单方法是将其保存到文件中。例如，将以下示例文本复制到名为 `testSFTPConfig.json` 的文件中。

```
// Listing for testSFTPConfig.json
{
   "UserSecretId": "arn:aws::secretsmanager:us-east-2:123456789012:secret:aws/
transfer/example-username-key",
   "TrustedHostKeys": [
      "sftp.example.com ssh-rsa AAAAbbbb...EEEE="
   ]
}
```

- 为连接器指定安全策略，输入安全策略名称。

> **Note**
>
> `SecretId`可以是整个 ARN，也可以是密钥的名称（*`example-username-key`*在前面的列表中）。

然后运行以下命令以创建连接器。

```
aws transfer create-connector --url "sftp://partner-SFTP-server-url" \
--access-role your-IAM-role-for-bucket-access \
--logging-role arn:aws:iam::your-account-id:role/service-role/
AWSTransferLoggingAccess \
--sftp-config file:///path/to/testSFTPConfig.json
--security-policy-name security-policy-name
```

存储密钥以与 SFTP 连接器配合使用

你可以使用 Secrets Manager 来存储 SFTP 连接器的用户凭证。创建密钥时，必须提供用户名。此外，您可以提供密码、私钥或两者兼而有之。有关更多信息，请参阅 SFTP 连接器配额。

> **Note**
>
> 当你在 Secret AWS 账户 s Manager 中存储密钥时，会产生费用。有关定价的信息，请参阅 AWS Secrets Manager 定价。

若要在 Secrets Manager 中存储 SFTP 连接器的用户凭证

1. 登录 AWS Management Console 并打开 AWS Secrets Manager 控制台，网址为 https://console.aws.amazon.com/secretsmanager/。
2. 在左侧导航窗格中，选择密钥。
3. 在密钥页面，选择存储新密钥。
4. 在选择密钥类型页面上，对于密钥类型，选择其他类型密钥。
5. 在键/值对部分，选择键/值选项卡。
   - 键 — 输入**`Username`**。
   - 值 — 输入有权连接到合作伙伴服务器的用户名。
6. 如果要提供密码，请选择添加行，然后在键/值对部分中，选择键/值选项卡。

   选择添加行，然后在键/值对部分选择键/值选项卡。
   - 键 — 输入**`Password`**。
   - 值 – 输入用户的密码。

7. 如果要提供私钥，请参阅 生成并格式化 SFTP 连接器私钥，其中介绍了如何输入私钥数据。

> **Note**
>
> 您输入的私钥数据必须与在远程 SFTP 服务器中为该用户存储的公钥相对应。

8. 选择下一步。
9. 在配置密钥页面，输入密钥的名称和描述。建议对名称使用前缀 **aws/transfer/**。例如，您可以将密钥命名为 **aws/transfer/connector-1**。
10. 选择下一步，接受配置轮换页面的默认设置。然后选择下一步。
11. 在审核页面，选择存储以创建和存储密钥。

生成并格式化 SFTP 连接器私钥

有关生成公钥/私钥对的完整详细信息，请参见。在 macOS、Linux 或 Unix 系统创建 SSH 密钥

例如，要生成用于 SFTP 连接器的私钥，以下示例命令会生成正确的密钥类型（将 *key_name* ####对的实际文件名）：

```
ssh-keygen -t rsa -b 4096 -m PEM -f key_name -N ""
```

> **Note**
>
> 创建用于 SFTP 连接器的密钥对时，请不要使用密码。要使 SFTP 配置正常运行，必须使用空密码。

此命令创建一个 RSA 密钥对，密钥大小为 4096 位。密钥以传统 PEM 格式生成，Transfer Family 需要使用该格式才能与 SFTP 连接器密钥一起使用。密钥保存在当前目录 *key_namekey_name*.pub（私钥）和（公钥）中：即运行ssh-keygen命令的目录。

> **Note**
>
> Transfer Family 不支持用于 SFTP 连接器的密钥采用 OpenSSH 格式 (-----BEGIN OPENSSH PRIVATE KEY-----)。密钥必须采用旧版 PEM 格式（ -----BEGIN RSA PRIVATE KEY----- 或 -----BEGIN EC PRIVATE KEY-----）。通过在运行命令时提供 -m PEM 选项，您可以使用 ssh-keygen 工具转换密钥。

生成密钥后，必须确保私钥采用 JSON 格式的嵌入式换行符 (”\ n “) 进行格式化。

使用命令将您现有的私钥转换为正确的格式，即带有嵌入式换行符的 JSON 格式。在这里，我们提供了jq和 Powershell 的示例。你可以使用任何你想要的工具或命令将私钥转换为带有嵌入式换行符的 JSON 格式。

jq command

此示例使用jq命令，该命令可从 Download jq 中下载。

```
jq -sR . path-to-private-key-file
```

例如，如果您的私钥文件位于中~/.ssh/my_private_key，则命令如下所示。

```
jq -sR . ~/.ssh/my_private_key
```

这会将正确格式的密钥（带有嵌入的换行符）输出到标准输出。

PowerShell

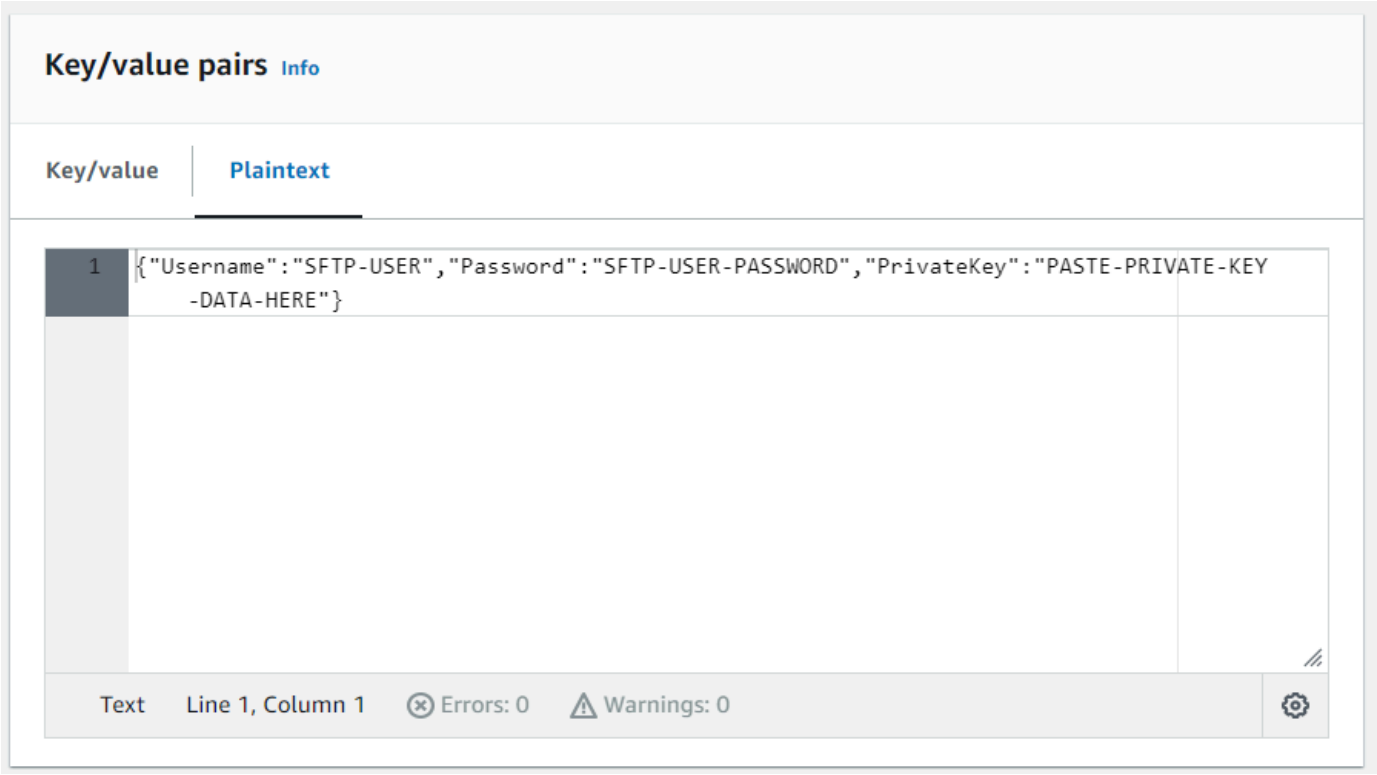
如果您使用的是 Windows，则可以使用将密钥 PowerShell 转换为正确的格式。以下 Powershell 命令将私钥转换为正确的格式。

```
Get-Content -Raw path-to-private-key-file | ConvertTo-Json
```

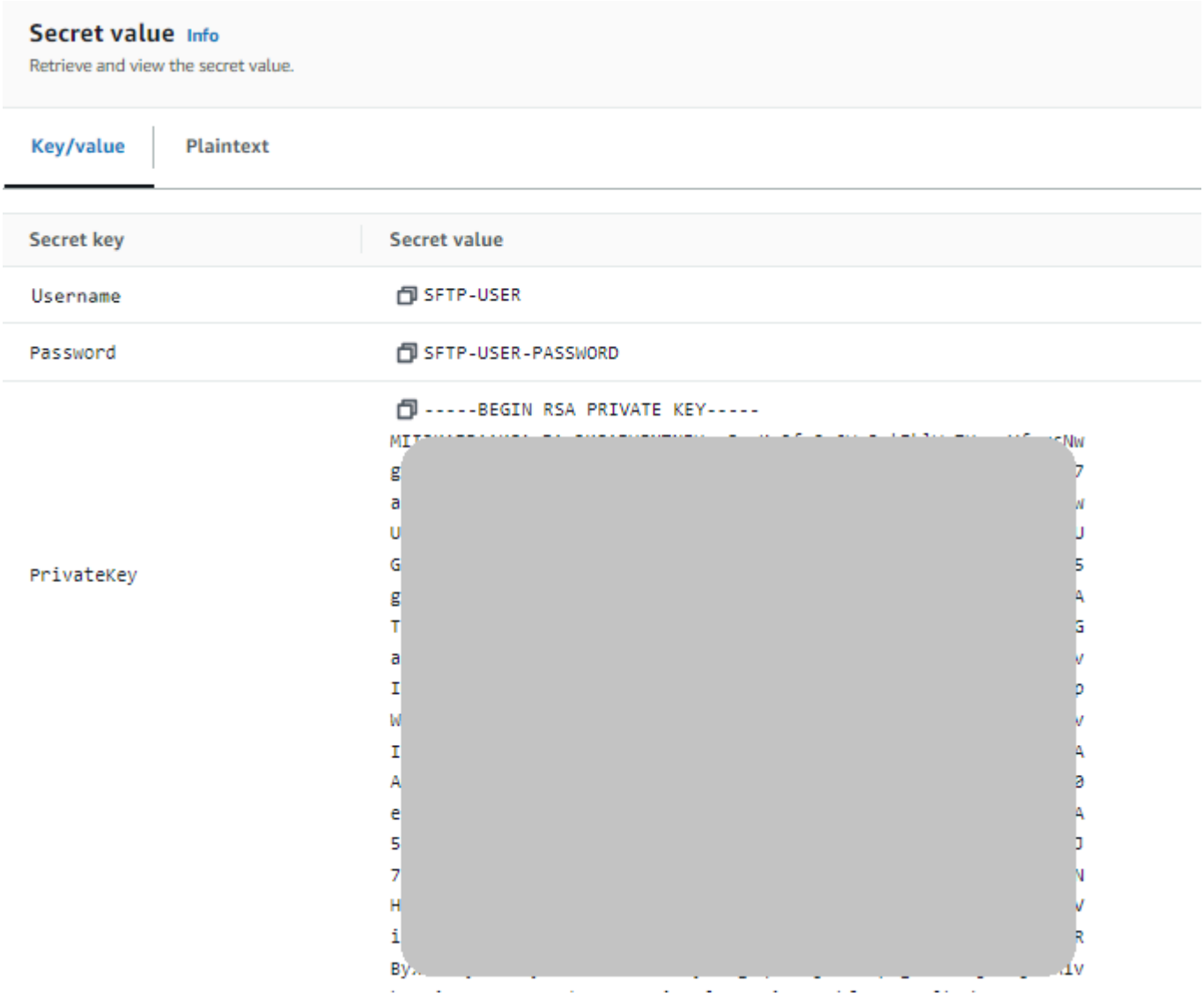
将私钥数据添加到密钥中以用于 SFTP 连接器

1. 在 Secrets Manager 控制台中，存储其他类型的密钥时，选择纯文本选项卡。文本应为空，只带有左右大括号 {}。
2. 使用以下格式粘贴您的用户名、私钥数据和/或密码。要获取私钥数据，请粘贴您在步骤 1 中运行的命令的输出。

```
{"Username":"SFTP-USER","Password":"SFTP-USER-PASSWORD","PrivateKey":"PASTE-PRIVATE-KEY-DATA-HERE"}
```



如果正确粘贴了私钥数据，则在选择“键/值”选项卡时应该会看到以下内容。请注意，私钥数据显示line-by-line的是连续的文本字符串。



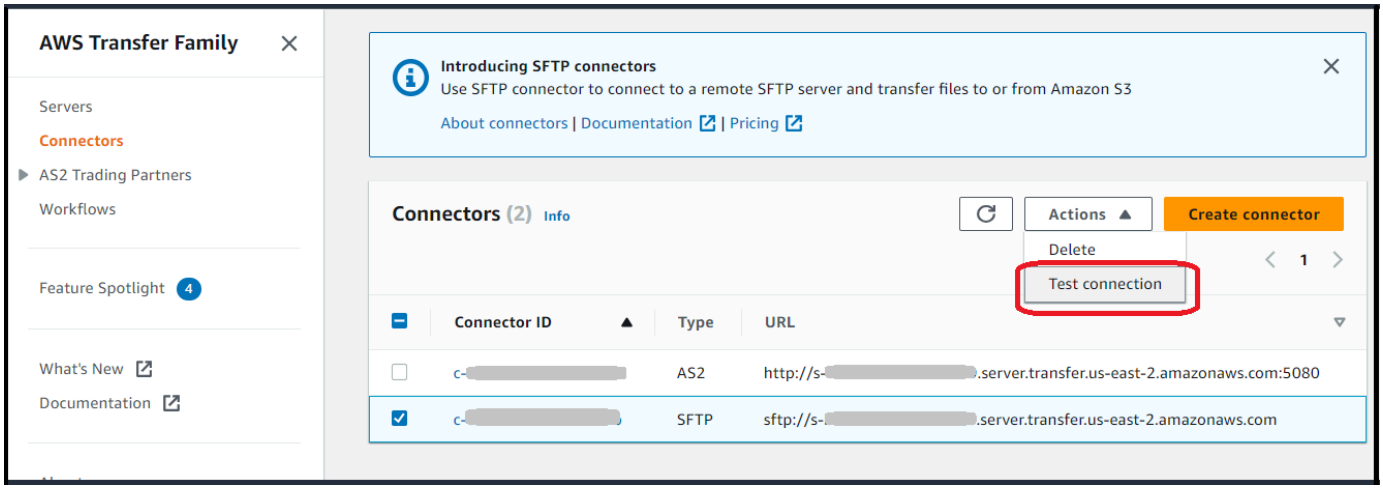
3. 继续执行步骤 8 中的 存储密钥以与 SFTP 连接器配合使用 过程，并遵循该过程直至结束。

测试 SFTP 连接器

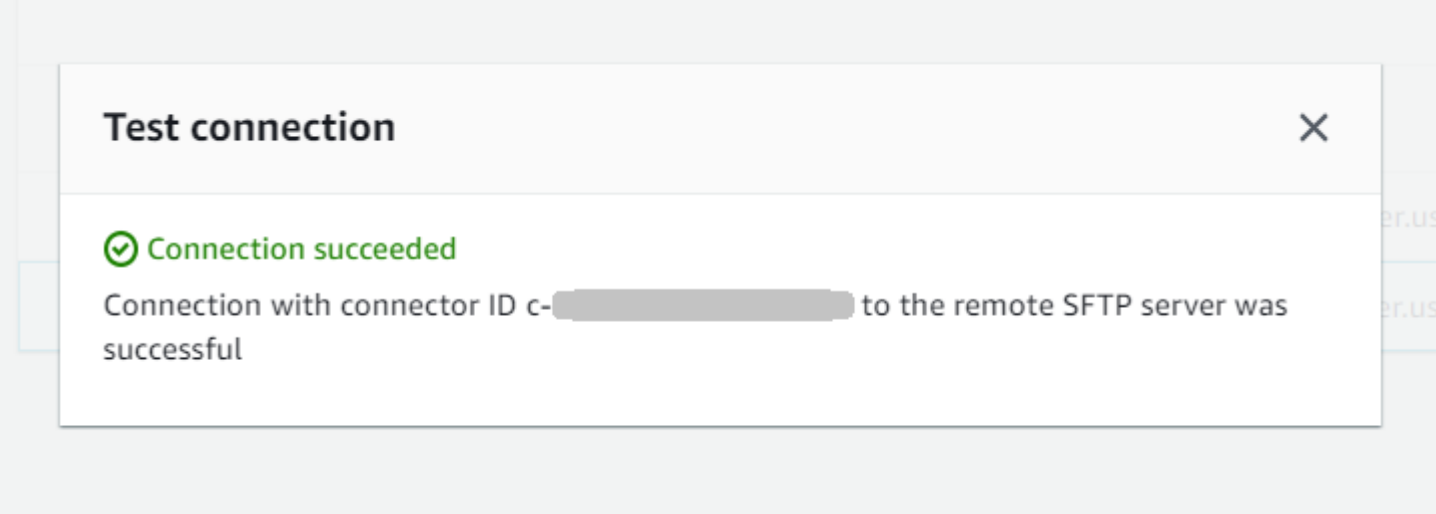
创建 SFTP 连接器后，我们建议您在尝试使用新连接器传输任何文件之前对其进行测试。

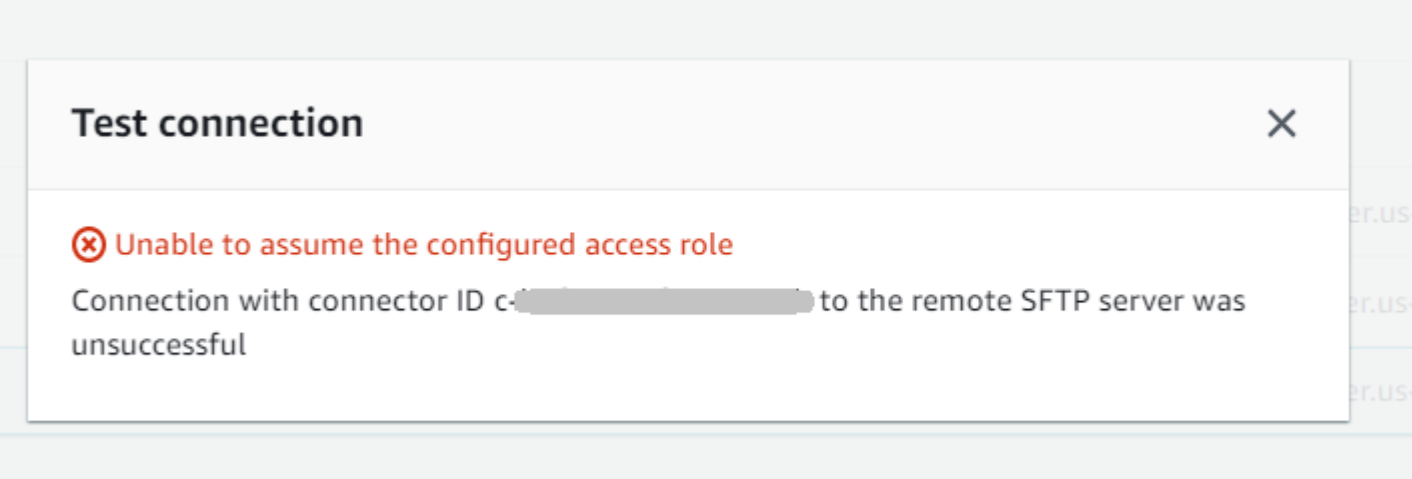
若要测试 SFTP 连接器

1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在左侧导航窗格中，选择连接器，然后选择一个连接器。
3. 从 操作菜单中选择 测试连接。



系统会返回一条消息，指示测试是通过还是失败。如果测试失败，系统会根据测试失败的原因提供错误消息。





> **Note**
>
> 要使用 API 测试您的连接器，请参阅 TestConnectionAPI 文档。

使用 SFTP 连接器发送和检索文件

SFTP 连接器扩展了 AWS Transfer Family 与云端和本地远程服务器通信的功能。您可以将远程源中生成和存储的数据与 AWS 托管的数据仓库集成，用于分析、业务应用程序、报告和审计。要启动向远程 SFTP 服务器的文件传输，您可以使用 StartFileTransferAPI 操作，该操作使用 SFTP 连接器来执行传输。每个 `StartFileTransfer` 请求可以包含 10 个不同的路径。

您可以通过查看服务器日志来监控文件传输。连接器活动会记录到格式为 `aws/transfer/`*`connector-id`*（例如 `aws/transfer/c-1234567890abcdef0`）的日志流中。如果您没有看到连接器的任何日志，请确保已为连接器指定了具有正确权限的日志记录角色。

有关创建连接器的详细信息，请参阅 配置 SFTP 连接器。

要使用 SFTP 连接器发送和检索文件，请使用 `start-file-transfer` AWS Command Line Interface (AWS CLI) 命令。根据是要发送文件（出站传输）还是接收文件（入站传输），您可以指定以下参数。

- 出站传输
  - `send-file-paths` 包含一到十个源文件路径，用于将文件传输到合作伙伴的 SFTP 服务器。
  - `remote-directory-path` 是客户的 SFTP 服务器上向其发送文件的远程路径。
- 入站传输
  - `retrieve-file-paths` 包含一到十条远程路径。每个路径都指定了将文件从合作伙伴的 SFTP 服务器传输到您的 Transfer Family 服务器的位置。
  - `local-directory-path` 是存储文件的 Amazon S3 位置（存储桶和可选前缀）。

要发送文件，请指定 `send-file-paths` 和 `remote-directory-path` 参数。您最多可以为 `send-file-paths` 参数指定 10 个文件。以下示例命令将位于 Amazon S3 存储空间中的名为 */DOC-EXAMPLE-SOURCE-BUCKET*`/file1.txt` 和 */DOC-EXAMPLE-SOURCE-BUCKET*`/file2.txt` 的文件发送到合作伙伴的 SFTP 服务器上的 `/tmp` 目录。要使用此示例命令，请将 *DOC-EXAMPLE-SOURCE-BUCKET* 替换为您自己的存储桶。

```
aws transfer start-file-transfer --send-file-paths /DOC-EXAMPLE-SOURCE-BUCKET/
file1.txt /DOC-EXAMPLE-SOURCE-BUCKET/file2.txt \
    --remote-directory-path /tmp --connector-id c-1111AAAA2222BBBB3 --region us-east-2
```

要接收文件，请指定 retrieve-file-paths 和 local-directory-path 参数。以下示例检索合作伙伴的 SFTP 服务器上的文件 /my/remote/file1.txt 和 /my/remote/file2.txt，并将其放在 Amazon S3 位置 /*DOC-EXAMPLE-BUCKET*/*prefix*。要使用此示例命令，请将 *user input placeholders* 替换为您自己的信息。

```
aws transfer start-file-transfer --retrieve-file-paths /my/remote/file1.txt  /my/
remote/file2.txt \
   --local-directory-path /DOC-EXAMPLE-BUCKET/prefix --connector-id c-2222BBBB3333CCCC4
 --region us-east-2
```

前面的示例指定了 SFTP 服务器上的绝对路径。您也可以使用相对路径：即相对于 SFTP 用户主目录的路径。例如，如果 SFTP 用户是 marymajor，而他们在 SFTP 服务器上的主目录是 /users/marymajor/，则以下命令会将 /*DOC-EXAMPLE-SOURCE-BUCKET*/file1.txt 发送到 /users/marymajor/test-connectors/file1.txt

```
aws transfer start-file-transfer --send-file-paths /DOC-EXAMPLE-SOURCE-BUCKET/file1.txt
 \
   --remote-directory-path test-connectors --connector-id c-2222BBBB3333CCCC4 --
region us-east-2
```

管理 SFTP 连接器

本主题介绍如何查看和更新 SFTP 连接器，并列出与 SFTP 连接器相关的配额。

> **Note**
>
> 系统会自动为每个连接器分配静态 IP 地址，这些地址在连接器的生命周期内保持不变。这允许您连接仅接受来自已知 IP 地址的入站连接的远程 SFTP 服务器。您的连接器会分配一组静态 IP 地址，这些地址由您中使用相同协议（SFTP 或 AS2）的所有连接器共享。 AWS 账户

主题

- 更新 SFTP 连接器

- 查看 SFTP 连接器详细信息
- SFTP 连接器配额

更新 SFTP 连接器

要更改连接器的现有参数值，可以运行 update-connector 命令。以下命令将区域 *region-id* 中连接器 *connector-id* 的密钥更新为 *secret-ARN*。要使用此示例命令，请将 *user input placeholders* 替换为您自己的信息。

```
aws transfer update-connector --sftp-config '{"UserSecretId":"secret-ARN"}' \
   --connector-id connector-id --region region-id
```

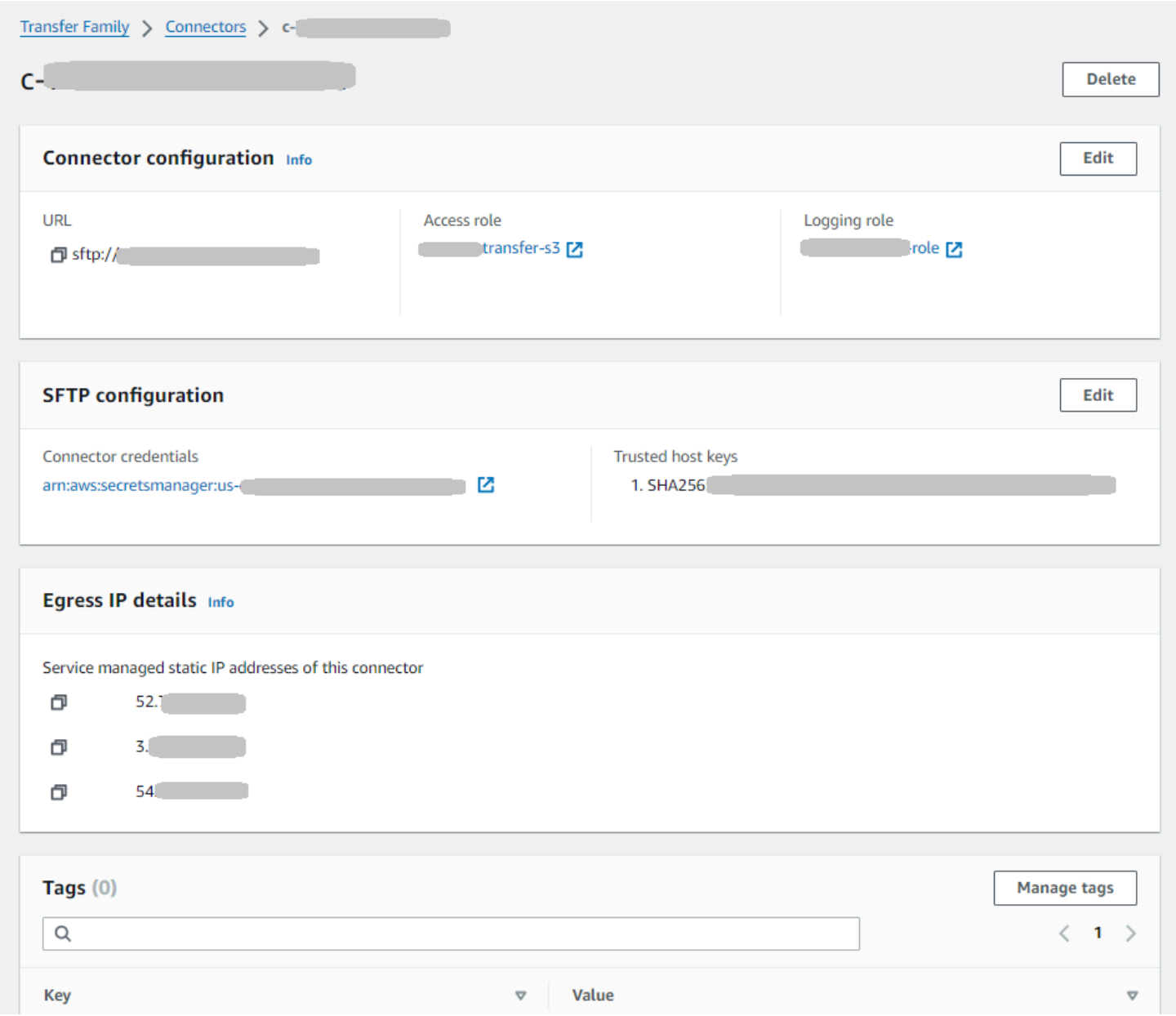
查看 SFTP 连接器详细信息

您可以在 AWS Transfer Family 控制台中找到 SFTP 连接器的详细信息和属性列表。

要查看连接器详细信息

1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在左侧导航窗格中，选择 Connectors (连接器)。
3. 在“连接器 ID”列中选择标识符以查看所选连接器的详细信息页面。

您可以通过在连接器详细信息页面上选择编辑来更改 SFTP 连接器的属性。



> **Note**
>
> 你可以通过运行以下 AWS Command Line Interface (AWS CLI) 命令来获取其中的大部分信息，尽管格式不同。要使用此示例命令，请将 *user input placeholders* 替换为您自己的信息。
>
> ```
> aws transfer describe-connector --connector-id your-connector-id
> ```
>
> 有关更多信息，请参阅《 API 参考》中的DescribeConnector。

SFTP 连接器配额

SFTP 连接器有以下配额。AS2 连接器的配额如 AS2 配额和限制 中所述。要申请增加可调整的配额，请参阅 AWS 一般参考 中的 AWS 服务 配额。

SFTP 连接器配额

名称	默认值	可调整
每秒最大测试连接事务数 (TPS)	每账户每秒 1 个请求	不支持
最大 StartFileTransfer TPS	每账户每秒 5 个请求	支持
待处理文件传输的最大队列大小	1000	不支持
最大文件大小	50 吉比特 (GiB)	不支持
每个文件的最大传输时间	6 小时	不支持
每个文件的最大请求等待时间	6 小时	不支持
最短 AccessRole 或 LoggingRole 会话持续时间	60 分钟	不支持
最大并发文件传输	每个连接器 1 个并发文件传输	不支持
每个账户每秒的最大文件传输请求数量	3	支持
每个账户的最大连接器数量（SFTP 和 AS2 连接器均计入此计数）	100	是
每个账户的连接器的最大带宽（SFTP 和 AS2 连接器均构成此值）	50MBps	不支持

为了存储 SFTP 连接器的凭证，每个 Secrets Manager 密钥都有与之关联的配额。如果您出于多种目的使用同一个密钥来存储多种类型的密钥，则可能会遇到这些配额。

- 单个密钥的总长度：12,000 个字符
- **`Password`**字符串的最大长度：1024 个字符
- **`PrivateKey`**字符串的最大长度：8192 个字符
- **`Username`**字符串的最大长度：100 个字符

AWS Transfer Family 适用于 AS2

适用性声明 2 (AS2) 是 RFC 定义的文件传输规范，包括强大的消息保护和验证机制。AS2 协议对于具有合规性要求的工作流程至关重要，这些要求依赖于协议中内置的数据保护和安全功能。

> **Note**
>
> Transfer Family 的 AS2 已通过 Drummond 认证。

零售、生命科学、制造、金融服务和公用事业等行业中依赖 AS2 处理供应链、物流和支付工作流程的客户可以使用 AWS Transfer Family AS2 端点安全地与其业务合作伙伴进行交易。已处理的数据可在本地访问， AWS 用于处理、分析和机器学习。这些数据也可用于与运行的企业资源规划 (ERP) 和客户关系管理 (CRM) 系统集成。 AWS借助AS2，客户可以大规模进行其 business-to-business (B2B) 交易， AWS 同时保持现有的业务合作伙伴集成和合规性。

如果您是 Transfer Family 客户，希望与配置了启用 AS2 的服务器的合作伙伴交换文件，则设置包括生成一个用于加密的公有-私有密钥对，另一个用于与合作伙伴签署和交换公有密钥。

我们有一个可以参加的研讨会，您可以在其中配置启用 AS2 的 Transfer Family 端点和 Transfer Family AS2 连接器。您可以在此处查看本次研讨会的详细信息。

保护传输中的 AS2 有效负载通常涉及使用加密消息语法 (CMS)，且通常使用加密和数字签名来提供数据保护和对等身份验证。已签名的消息处置通知（MDN）响应有效负载提供消息已接收并成功解密的验证（不可否认性）。

这些 CMS 有效负载和 MDN 响应通过 HTTP 进行传输。

> **Note**
>
> 目前不支持 HTTPS AS2 服务器端点。目前，客户负责 TLS 终止。

有关设置适用性声明 2 (AS2) 配置的详细 step-by-step 演练，请参阅教程。设置 AS2 配置

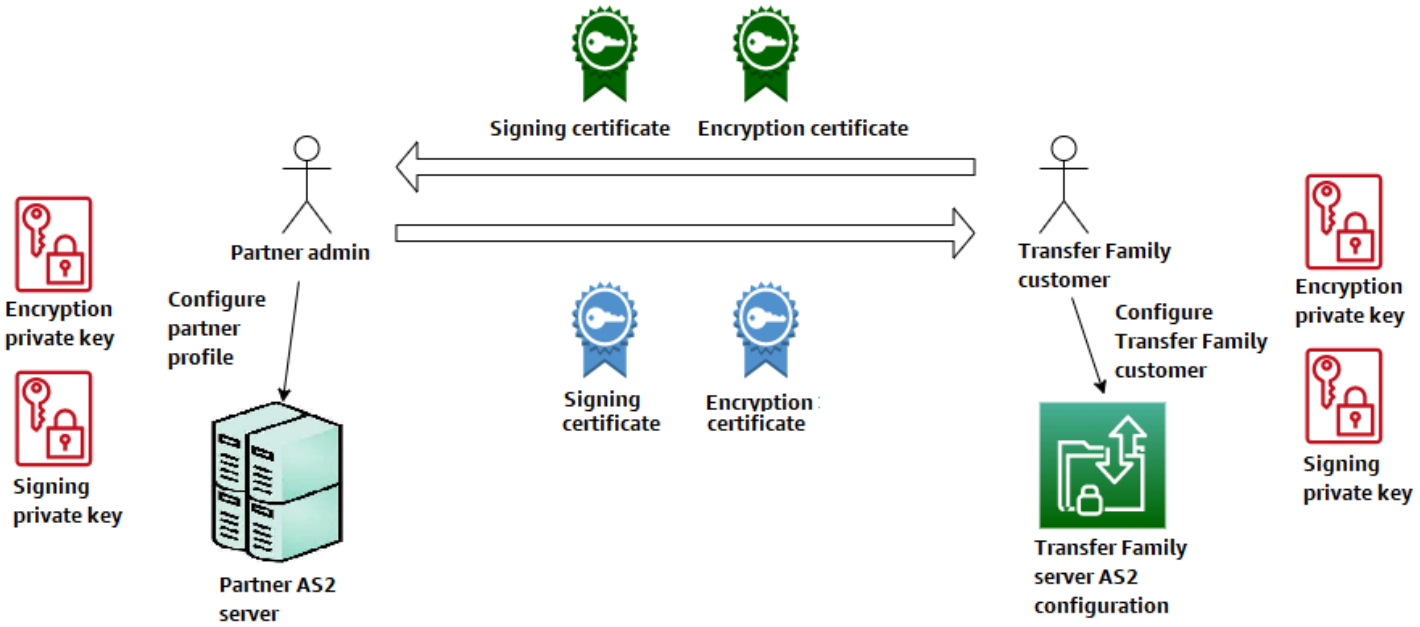
主题

- AS2 使用案例
- 配置 AS2
- 配置 AS2 连接器

- 管理 AS2 合作伙伴
- 发送和接收 AS2 消息
- 监控 AS2 的使用情况

AS2 使用案例

如果您是想要与已配置 AS2 服务器的合作伙伴交换文件的 AWS Transfer Family 客户，则设置中最复杂的部分涉及生成一个用于加密的公私密钥对，另一个用于与合作伙伴签署和交换公钥。



在 AWS Transfer Family 与 AS2 一起使用时，请考虑以下变体。

> **Note**
>
> 贸易伙伴是与该合作伙伴资料关联的合作伙伴。
> 下表中所有提及 MDN 的内容都假设已签名的 MDN。

AS2 使用案例

仅限入站的使用案例
• 将加密的 AS2 消息从交易伙伴传输到 Transfer Family 服务器。 在此情况下，您可以执行以下操作：

1. 为您的贸易伙伴和您自己创建档案。
2. 创建使用 AS2 协议的 Transfer Family 服务器。
3. 创建协议并将其添加到您的服务器。
4. 导入带有私钥的证书并将其添加到您的个人资料中，然后将公钥导入您的合作伙伴资料进行加密。
5. 拿到这些物品后，将证书的公有密钥发送给您的交易伙伴。

现在，您的合作伙伴可以向您发送加密消息，您可以将其解密并存储在您的 Amazon S3 存储桶中。

- 将加密的 AS2 消息从交易伙伴传输到 Transfer Family 服务器并添加签名。

  在这种情况下，您仍然只进行入站传输，但现在您希望让您的合作伙伴签署他们发送的消息。在这种情况下，请导入贸易伙伴的签名公钥（作为添加到合作伙伴资料中的签名证书）。
- 将加密的 AS2 消息从交易伙伴传输到 Transfer Family 服务器，然后添加签名和发送 MDN 响应。

  在这种情况下，您仍然只进行入站传输，但是现在，除了接收已签名的有效负载外，您的交易伙伴还希望接收签名的 MDN 响应。

  1. 导入您的公有和私有签名密钥（作为签名证书导入您的配置文件中）。
  2. 将公开签名密钥发送给您的贸易伙伴。

仅限出站的使用案例

- 将加密的 AS2 消息从 Transfer Family 服务器传输给贸易伙伴。

  这种情况与仅限入站传输的使用案例类似，不同之处在于您无需向 AS2 服务器添加协议，而是创建连接器。在这种情况下，您可以将贸易伙伴的公钥导入他们的个人资料中。
- 将加密的 AS2 消息从 Transfer Family 服务器传输给贸易伙伴并添加签名。

  您仍然只进行出站转账，但现在您的贸易伙伴希望您在发送给他们的消息上签名。

  1. 导入您的签名私有密钥（作为签名证书添加至您的配置文件中）。
  2. 将您的公钥发送给您的贸易伙伴。
- 将加密的 AS2 消息从 Transfer Family 服务器传输到贸易伙伴，然后添加签名并发送 MDN 响应。

  您仍然只能进行出站转账，但是现在，除了发送已签名的有效载荷外，您还希望收到贸易伙伴签名的 MDN 响应。

  1. 您的贸易伙伴向您发送他们的公开签名密钥。
  2. 导入您的贸易伙伴的公钥（作为添加到您的合作伙伴资料中的签名证书）。

入站和出站使用案例

- 在 Transfer Family 服务器和交易伙伴之间双向传输加密的 AS2 消息。

  在此情况下，您可以执行以下操作：

  1. 为您的贸易伙伴和您自己创建档案。
  2. 创建使用 AS2 协议的 Transfer Family 服务器。
  3. 创建协议并将其添加到您的服务器。
  4. 创建连接器。
  5. 导入带有私钥的证书并将其添加到您的个人资料中，然后将公钥导入您的合作伙伴资料进行加密。
  6. 从您的贸易伙伴那里接收公钥并将其添加到他们的个人资料中进行加密。
  7. 拿到这些物品后，将证书的公有密钥发送给您的交易伙伴。

  现在，您和您的交易伙伴可以交换加密消息，并且双方都可以对其进行解密。您可以将接收的消息存储在 Amazon S3 存储桶中，且您的合作伙伴可以解密和存储您发送给他们的消息。
- 在 Transfer Family 服务器和贸易伙伴之间双向传输加密的 AS2 消息并添加签名。

  现在您和您的合作伙伴想要签名消息。

  1. 导入您的签名私有密钥（作为签名证书添加至您的配置文件中）。
  2. 将您的公钥发送给您的贸易伙伴。
  3. 导入您的贸易伙伴的签名公钥并将其添加到他们的个人资料中。
- 在 Transfer Family 服务器和交易伙伴之间双向传输加密的 AS2 消息，然后添加签名并发送 MDN 响应。

  现在，您想交换签名的有效负载，并且您和您的交易伙伴都想要 MDN 响应。

  1. 您的贸易伙伴向您发送他们的公开签名密钥。
  2. 导入贸易伙伴的公钥（作为合作伙伴资料的签名证书）。
  3. 将您的公钥发送给您的贸易伙伴。

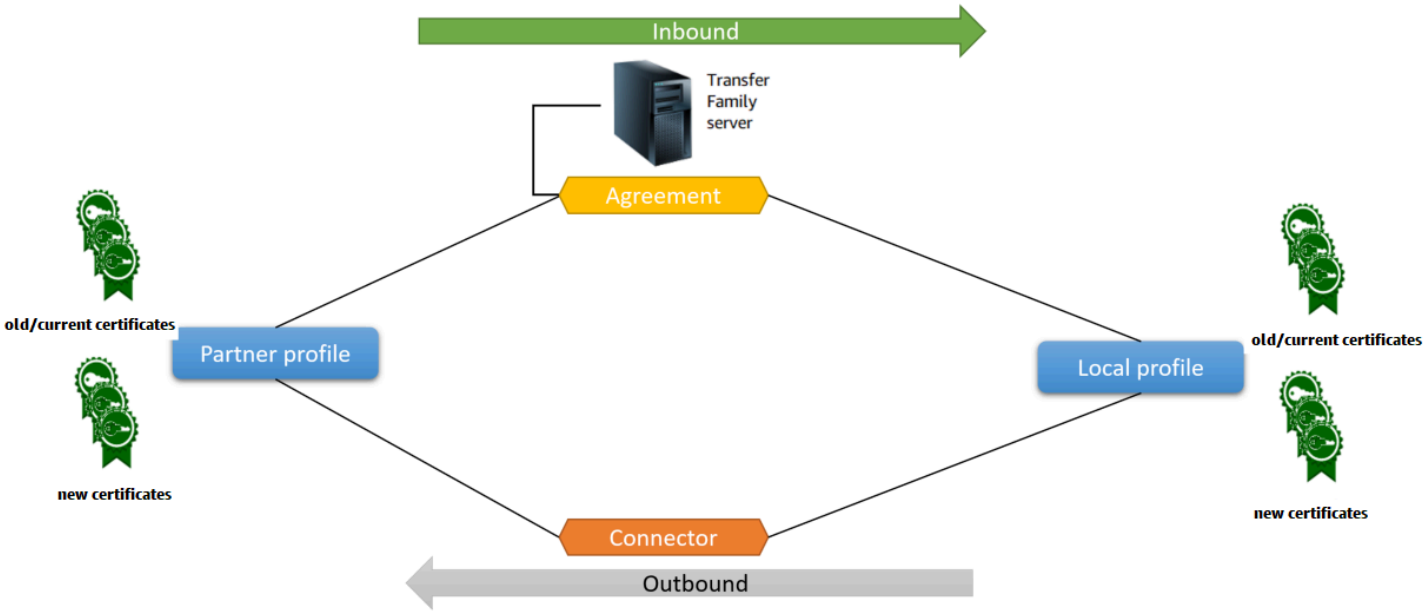
配置 AS2

要创建启用 AS2 的服务器，还必须指定以下组件：

- 协议 — 双边贸易伙伴协议或伙伴关系，定义交换消息（文件）的双方之间的关系。为了定义协议，Transfer Family 结合了服务器、本地配置文件、合作伙伴配置文件和证书信息。Transfer Family AS2-入站流程使用协议。
- 证书 — 在 AS2 通信中使用公有密钥 (X.509) 证书进行消息加密和验证。证书也用于连接器端点。
- 本地配置文件和合作伙伴配置文件 — 本地配置文件定义本地（启用 AS2 的 Transfer Family 服务器）组织或“一方”。同样，合作伙伴配置文件定义了 Transfer Family 外部的远程合作伙伴组织。

虽然并非所有启用 AS2 的服务器都需要连接器，但对于出站传输，则需要连接器。连接器捕获出站连接的参数。连接器是将文件发送到客户的外部非AWS服务器所必需的。

下图显示了入站和出站流程中涉及的 AS2 对象之间的关系。



有关 AS2 配置的 end-to-end 示例，请参见设置 AS2 配置。

主题

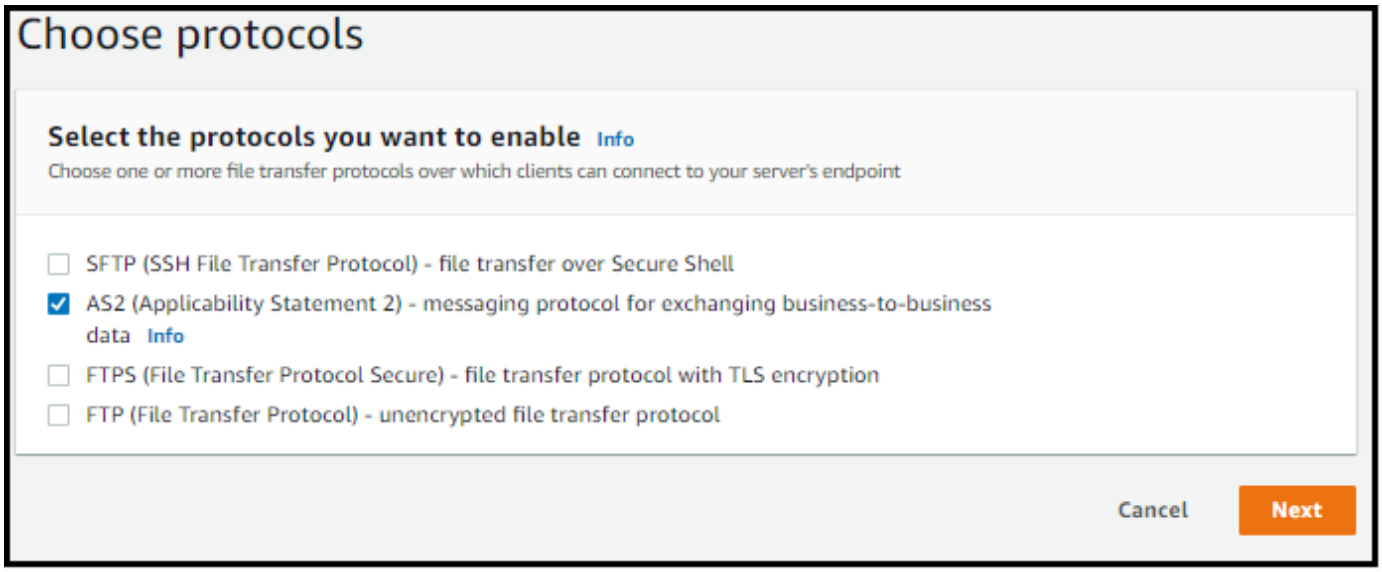
- 使用 Transfer Family 控制台创建 AS2 服务器
- 使用模板创建演示 Transfer Family AS2 堆栈
- AS2 的配置和限制
- AS2 特征和功能

使用 Transfer Family 控制台创建 AS2 服务器

此步骤说明了如何使用 Transfer Family 控制台创建启用 AS2 的服务器。如果要改用 AWS CLI，请参阅 the section called “第 2 步：创建使用 AS2 协议的 Transfer Family 服务器”。

要创建启用 AS2 的服务器

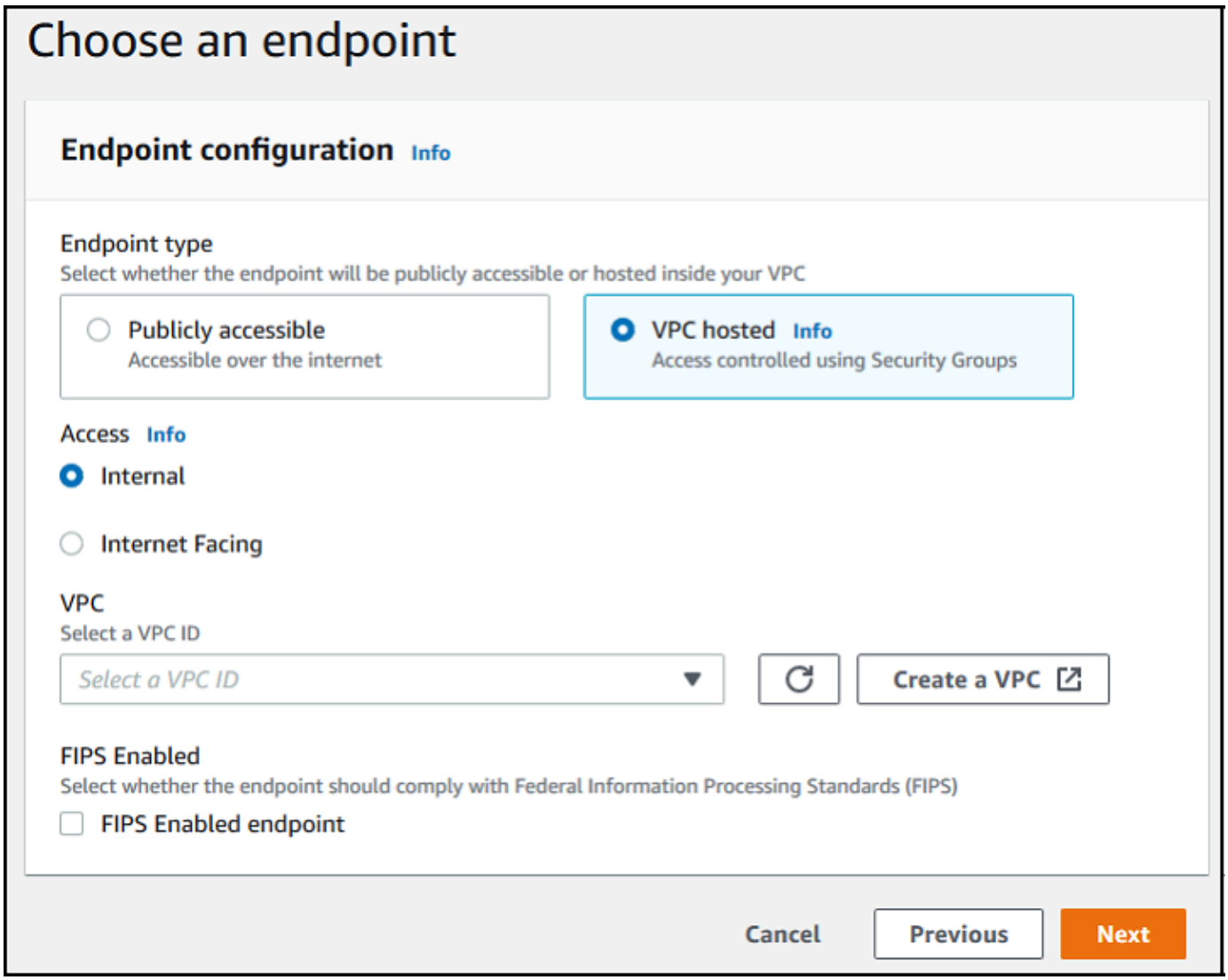
1. 打开AWS Transfer Family控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在左侧的导航窗格中，选择服务器，然后选择创建服务器。
3. 在选择协议页面上，选择 AS2（适用性声明 2），然后选择下一步。



4. 在选择身份提供商页面上，选择下一步。

> **Note**
>
> 对于 AS2，您无法选择身份提供商，因为 AS2 协议不支持基本身份验证。相反，您可以通过虚拟私有云 (VPC) 安全组控制访问权限。

5. 在选择端点页面上，执行以下操作：



a. 对于端点类型，选择托管服务器端点的 VPC 托管。有关设置 VPC 主机端点的信息，请参阅在虚拟私有云中创建服务器。

> **Note**
>
> AS2 协议不支持可公开访问的端点。要使您的 VPC 端点可通过互联网访问，请在访问权限下选择面向互联网，然后提供您的弹性 IP 地址。

b. 对于访问权限，请选择下列选项之一：

- 内部 — 选择此选项可在您的 VPC 和 VPC 连接的环境中提供访问权限，例如通过AWS Direct Connect或 VPN 的本地数据中心。

- 面向互联网 — 选择此选项可通过互联网以及 VPC 和 VPC 连接环境（例如通过AWS Direct Connect或 VPN 的本地数据中心）提供访问权限。

  如果您选择面向互联网，请在出现提示时提供您的弹性 IP 地址。

c. 对于 VPC，选择现有 VPC 或选择创建 VPC 以创建新的 VPC。

d. 对于启用 FIPS，请清除启用 FIPS 端点复选框。

> **Note**
>
> AS2 协议不支持启用 FIPS 的端点。

e. 请选择 Next（下一步）。

6. 在选择域页面上，选择 Amazon S3 以使用所选协议将文件作为对象存储和访问。

   请选择 Next（下一步）。

7. 在配置其他详细信息页面上，选择所需的设置。

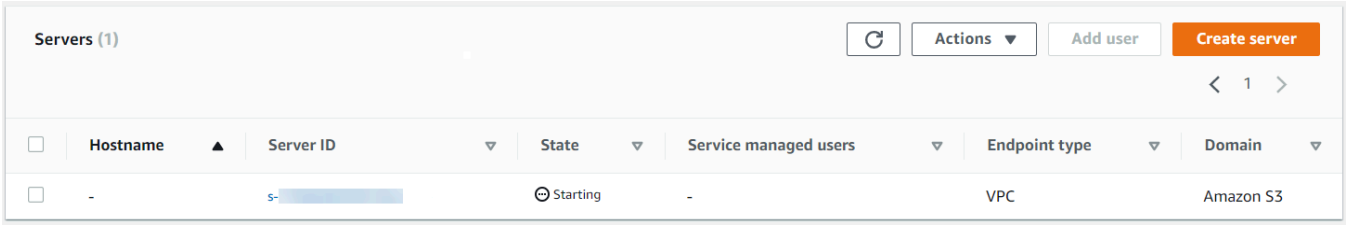
> **Note**
>
> 如果您正在与 AS2 一起配置任何其他协议，则所有其他详细设置都适用。但是，对于 AS2 协议，唯一适用的设置是CloudWatch 日志和标签部分中的设置。
> 尽管设置 CloudWatch 日志记录角色是可选的，但我们强烈建议您对其进行设置，以便您可以查看消息状态并解决配置问题。

8. 在查看并创建页面上，查看您的选择以确保它们正确无误。

   - 如果要编辑任何设置，请选择要更改步骤旁边的编辑。

> **Note**
>
> 如果您编辑某个步骤，我们建议您在选择编辑的步骤之后查看每个步骤。

   - 如果没有任何更改，请选择创建服务器来创建您的服务器。您将转至如下所示的 Servers (服务器) 页面，其中列出了您的新服务器。



您的新服务器状态更改为在线可能需要几分钟时间。到时候，您的服务器可以执行用户的文件操作。

使用模板创建演示 Transfer Family AS2 堆栈

我们提供了一个独立的AWS CloudFormation模板来快速创建启用 AS2 的 Transfer Family 服务器。该模板为服务器配置公有 Amazon VPC 端点、证书、本地和合作伙伴配置文件、协议和连接器。

在使用此模板之前，请注意以下事项：

- 如果您根据此模板创建堆栈，则需为使用的 AWS 资源计费。
- 该模板会创建多个证书并将其放入AWS Secrets Manager中以安全地存储它们。如果您愿意，您可以从 Secrets Manager 中删除这些证书，因为使用此服务需要付费。在 Secrets Manager 中删除这些证书不会将其从 Transfer Family 服务器中删除。因此，演示堆栈的功能不受影响。但是，对于要在生产 AS2 服务器上使用的证书，您可能需要使用 Secrets Manager 来管理和定期轮换存储的证书。
- 我们建议您仅将模板用作基础，主要用于演示目的。如果您想在生产环境中使用此演示堆栈，我们建议您修改模板的 YAML 代码以创建更强大的堆栈。例如，创建生产级证书，并创建可在生产中使用的AWS Lambda函数。

使用模板创建支持 AS2 的 Transfer Family 服务器 CloudFormation

1. 打开 AWS CloudFormation 控制台，地址：https://console.aws.amazon.com/cloudformation。
2. 在左侧导航窗格中，选择 Stacks (堆栈)。
3. 选择 Create stack (创建堆栈)，然后选择 With new resources (standard) (使用新资源(标准))。
4. 在先决条件 — 准备模板部分，请选择模板已就绪。
5. 复制此链接，即AS2 演示模板，然后将其粘贴到Amazon S3 URL字段中。
6. 请选择 Next（下一步）。
7. 在指定堆栈详细信息页面上，命名您的堆栈，然后指定以下参数：

- 在 AS2 下，输入本地 AS2 ID 和合作伙伴 AS2 ID 的值，或者分别接受默认值`local`和`partner`。
- 在网络下，输入安全组入口 CIDR IP 的值，或接受默认值`0.0.0.0/0`。

> **Note**
>
> 此值采用 CIDR 格式，指定允许哪些 IP 地址可传入流量到 AS2 服务器。默认值`0.0.0.0/0`允许所有 IP 地址。

- 在常规下，输入前缀的值，或接受默认值`transfer-as2`。此前缀位于堆栈创建的任何资源名称之前。例如，如果您使用默认前缀，则会将您的 Amazon S3 存储桶命名为`transfer-as2-`*`TransferS3BucketName`*。

8. 请选择 Next（下一步）。在配置堆栈选项页面上，再次选择下一步。
9. 查看您正在创建的堆栈的详细信息，然后选择创建堆栈。

> **Note**
>
> 在页面底部的Capabilities（能力）下，您必须确认AWS CloudFormation可能会创建 AWS Identity and Access Management (IAM) 资源。

创建堆栈后，您可以使用 AWS Command Line Interface (AWS CLI) 将测试 AS2 消息从伙伴服务器发送到本地 Transfer Family 服务器。将创建用于发送测试消息的示例AWS CLI命令以及堆栈中的所有其他资源。

要使用此示例命令，请转到堆栈的 “输出” 选项卡，然后复制 TransferExampleAs2Com mand。然后，您可以使用AWS CLI运行该命令。如果尚未安装AWS CLI，请参阅AWS Command Line Interface用户指南中的安装或更新最新版本的AWS CLI。

此示例命令采用以下格式：

```
aws s3api put-object --bucket TransferS3BucketName --key test.txt && aws transfer
 start-file-transfer --region aws-region --connector-id TransferConnectorId --send-
file-paths /TransferS3BucketName/test.txt
```

> **Note**
>
> 此命令的版本包含堆栈中*TransferS3BucketName*和*TransferConnectorId*资源的实际值。

此示例命令由两个单独的命令组成，这两个命令使用&&字符串链接在一起。

第一个命令在您的存储桶中创建一个新的空文本文件：

```
aws s3api put-object --bucket TransferS3BucketName --key test.txt
```

然后，第二个命令使用连接器将文件从合作伙伴配置文件发送到本地配置文件。Transfer Family 服务器已设置协议，允许本地配置文件接受来自合作伙伴配置文件的消息。

```
aws transfer start-file-transfer --region aws-region --connector-id TransferConnectorId
 --send-file-paths /TransferS3BucketName/test.txt
```

运行命令后，您可以转到您的 Amazon S3 存储桶 (*TransferS3BucketName*) 并查看其内容。如果命令成功，您应看到存储桶中有以下对象：

- `processed/` – 此文件夹包含一个 JSON 文件，该文件描述传输的文件和 MDN 响应。
- `processing/` – 此文件夹暂时包含正在处理的文件，但在传输完成后，此文件夹应为空。
- *server-id*`/` – 此文件夹根据您的 Transfer Family 服务器 ID 命名。它包含`from-`*partner*（此文件夹根据合作伙伴的 AS2 ID 动态命名），其本身包含`failed/`、`processed/`和`processing/`文件夹。/*server-id*`/from-`*partner*`/processed/`文件夹包含传输的文本文件的副本以及相应的 JSON 和 MDN 文件。
- `test.txt` – 此对象是传输的（空）文件。

AS2 的配置和限制

本主题说明了使用适用性声明 2（AS2）协议的传输支持的配置、特征和功能，包括接受的密码和摘要。本节还介绍了 AS2 传输的限制和已知问题。

主题

- AS2 支持的配置
- AS2 配额和限制

AS2 支持的配置

签名、加密、压缩、MDN

对于入站和出站传输，以下项目为必需或可选项目：

- 加密 - 必需（对于 HTTP 传输，这是目前唯一支持的传输方法）。只有通过终止 TLS 的代理（例如应用程序负载均衡器(ALB)）转发且 `X-Forwarded-Proto: https` 标头存在的情况下，才会接受未加密的消息。
- 签名 - 可选
- 压缩 - 可选（目前唯一支持的压缩算法是 ZLIB）
- 邮件处置通知 (MDN) - 可选

密码

入站和出站传输均支持以下密码：

- AES128_CBC
- AES192_CBC
- AES256_CBC
- 3DES（仅用于向后兼容）

摘要

支持以下摘要：

- 入站签名和 MDN – SHA1、SHA256、SHA384、SHA512
- 出站签名和 MDN – SHA1、SHA256、SHA384、SHA512

MDN

对于 MDN 响应，支持某些类型，如下所示：

- 入站传输 - 同步和异步
- 出站传输 - 仅限同步
- 简单邮件传输协议（SMTP）（电子邮件 MDN） – 不支持

Transports

- 入站传输 – HTTP 是目前唯一支持的传输，您必须明确指定。

> **Note**
>
> 如果您需要使用 HTTPS 进行入站传输，则可以在应用程序负载均衡器或网络负载均衡器上终止 TLS。通过 HTTPS 接收 AS2 消息中对此进行了描述。

- 出站传输 - 如果您提供 HTTP URL，则还必须指定加密算法。如果您提供 HTTPS URL，则可以选择为加密算法指定 NONE。

AS2 配额和限制

本节讨论 AS2 的限额和限制

主题

- AS2 限额
- 处理密钥的限额
- 已知限制条件

AS2 限额

AS2 文件传输有以下限额。要申请增加可调整的限额，请参阅AWS 一般参考中的AWS 服务 限额。

AS2 限额

名称	默认值	可调整
每台服务器的入站 AS2 请求	每秒 25 个	不支持
每台服务器正在处理的入站 AS2 请求	100	不支持
每个文件传输功能请求的最大文件数	10	不支持
每个连接器正在处理的出站 AS2 请求	100	不支持

名称	默认值	可调整
最大文件大小（压缩或未压缩）	50 MiB	支持
不活动超时	350 秒	不支持
每个账户的最大合作伙伴配置文件数	1000（每个合作伙伴配置文件最多 10 个证书：不可调整）	支持
每个账户的最大证书数	1000	支持
每个账户每秒的最大文件传输请求数量	3	支持
每个账户的最大连接器数量（SFTP 和 AS2 连接器均计入此计数）	100	是
每个账户的连接器的最大带宽（SFTP 和 AS2 连接器均构成此值）	50MBps	不支持
每台服务器的最大协议数	100	是

处理密钥的限额

AWS Transfer Family AWS Secrets Manager 代表使用基本身份验证的 AS2 客户拨打电话。此外，Secrets Manager 还会拨打电话 AWS KMS。

> **Note**
>
> 这些限额并不特定于您对 Transfer Family 的密钥的使用：它们是在您 AWS 账户中所有服务之间共享的。

对于 Secrets Manager`GetSecretValue`，适用的配额是组合速率 DescribeSecret 和 GetSecretValue API 请求，如AWS Secrets Manager 配额中所述。

Secrets Manager **`GetSecretValue`**

名称	值	描述
和 GetSecretValue API 请求 DescribeSecret 的合并速率	每个受支持的区域：每秒 1 万个	DescribeSecret 和 GetSecret Value API 请求的每秒最大交易总和。

对于 AWS KMS，以下配额适用`Decrypt`。有关详细信息，请参阅[每个 AWS KMS API 操作的请求配额]

AWS KMS **`Decrypt`**

限额名称	默认值（每秒请求数）

加密操作（对称）请求速率	这些共享配额因请求中使用的 AWS KMS 密钥类型 AWS 区域 和密钥类型而异。每个配额都单独计算。
	• 5500（共享）
	• 在以下区域中为 10000（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
	  • 欧洲（伦敦），eu-west-2
	• 在以下区域中为 50000（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

自定义密钥存储请求限额	自定义密钥存储请求限额是针对每个自定义密钥存储单独计算的。

限额名称	默认值（每秒请求数）
Note 此限额仅适用于使用外部密钥存储的情况。	• 每个 AWS CloudHSM 密钥库有 1,800 个（共享） • 每个外部密钥存储 1800 次（共享）

已知限制条件

- 不支持服务器端 TCP 保持活动状态。除非客户端发送保持活动状态的数据包，否则连接将在处于非活动状态 350 秒后超时。
- 要使有效协议被服务接受并显示在 Amazon CloudWatch 日志中，消息必须包含有效的 AS2 标头。
- 从 AWS Transfer Family AS2 接收消息的服务器必须支持 RFC 6211 中定义的用于验证消息签名的加密消息语法 (CMS) 算法保护属性。某些较早的 IBM Sterling 产品不支持此属性。
- 重复的消息 ID 会导致 已处理/警告：duplicate-document 消息。
- AS2 证书的密钥长度必须至少为 2048 位，最多为 4096 位。
- 向交易伙伴的 HTTPS 端点发送 AS2 消息或异步 MDN 时，消息或 MDN 必须使用由公开信任的证书颁发机构 (CA) 签署的有效 SSL 证书。目前仅支持出站传输自签名证书。
- 端点必须支持 TLS 版本 1.2 协议和安全策略允许的加密算法（如AWS Transfer Family 服务器的安全策略中所述）。
- 目前不支持双向 TLS (mTLS)。
- 目前不支持 AS2 版本 1.2 中的多个附件和证书交换消息 (CEM)。
- 基本身份验证目前仅支持出站消息。

AS2 特征和功能

下表列出了使用 AS2 的 Transfer Family 资源可用的特征和功能。

AS2 特征

Transfer Family 为 AS2 提供以下特征。

功能	由... 支持 AWS Transfer Family
德拉蒙德认证	支持
AWS CloudFormation 支持	支持
亚马逊 CloudWatch指标	支持
SHA-2 加密算法	支持
对亚马逊 S3 的支持	支持
Amazon EFS 支持	不支持
定时消息	是 [1]
AWS Transfer Family 托管工作流程	不支持
证书交换消息 (CEM)	不支持
双向 TLS (mTLS)	不支持
Support 对自签名证书的支持	支持

1. 通过使用 Amazon 的计划 AWS Lambda 功能提供出站预设消息 EventBridge

AS2 发送和接收功能

下表提供了 AWS Transfer Family AS2 发送和接收功能的列表。

能力	入站：通过服务器接收	出站：使用连接器发送
TLS 加密传输 (HTTPS)	是 [1]	支持
非 TLS 传输 (HTTP)	支持	是 [2]
同步 MDN	支持	支持
消息压缩	支持	支持
异步 MDN	支持	不支持

能力	入站：通过服务器接收	出站：使用连接器发送
静态 IP 地址	支持	支持
带上你自己的 IP 地址	支持	不支持
多个文件附件	不支持	不支持
基本身份验证	不支持	支持
AS2 重启	不适用	不支持
每封邮件的自定义主题	不适用	不支持

1. 网络负载均衡器（NLB）提供入站 TLS 加密传输

2. 只有启用加密后，出站非 TLS 传输才可用

配置 AS2 连接器

连接器的用途是在交易伙伴之间建立出站传输的关系 — 将 AS2 文件从 Transfer Family 服务器发送至合作伙伴拥有的外部目的地。对于连接器，您可以指定本地方、远程合作伙伴及其证书（通过创建本地和合作伙伴配置文件）。

有了连接器后，您可以将信息传输给您的交易伙伴。每台 AS2 服务器都分配了三个静态 IP 地址。AS2 连接器使用这些 IP 地址通过 AS2 向您的贸易伙伴发送异步 mDN。

> **Note**
>
> 交易伙伴收到的消息大小将与 Amazon S3 中的对象大小不匹配。之所以出现这种差异，是因为 AS2 消息在发送文件之前将其封装在信封中。因此，即使文件是以压缩方式发送的，文件大小也可能会增加。因此，请确保交易伙伴的最大文件大小大于您发送的文件的大小。

创建 AS2 连接器

此过程说明如何使用 AWS Transfer Family 控制台创建 AS2 连接器。如果要 AWS CLI 改用，请参阅the section called “第 6 步：创建您与合作伙伴之间的连接器”。

创建 AS2 连接器

1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在左侧的导航窗格中，选择连接器，然后选择创建连接器。
3. 在连接器配置部分中，指定以下信息：

   - URL — 输入出站连接的 URL。
   - 访问角色-选择要使用的 (IAM) 角色的亚马逊资源名称 AWS Identity and Access Management (ARN)。确保角色提供对 `StartFileTransfer` 请求中所使用文件位置父目录的读取和写入访问权限。此外，确保角色对拟定发送`StartFileTransfer`的父目录提供读取和写入访问权限。

     > **Note**
     >
     > 如果您对连接器执行基本身份验证，则访问角色需要密钥的`secretsmanager:GetSecretValue`权限。如果使用客户管理的密钥而不是 in 对密钥进行加密 AWS Secrets Manager，则该角色还需要`kms:Decrypt`获得该密钥的权限。 AWS 托管式密钥 如果您使用前缀 `aws/transfer/` 命名您的密钥，则可以使用通配符 (*) 添加必要的权限，如创建密钥的权限示例中所示。

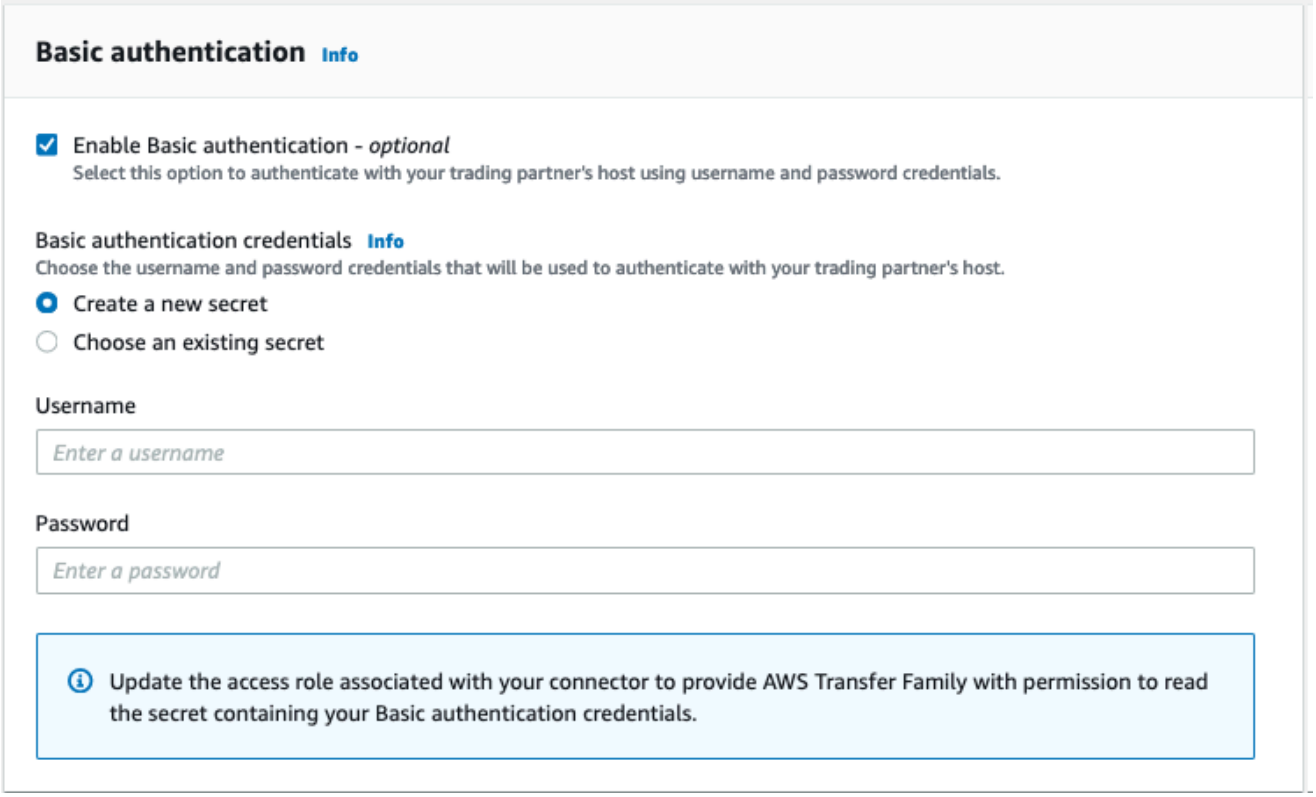
   - 日志角色（可选）-选择连接器用于将事件推送到 CloudWatch 日志的 IAM 角色。
4. 在 AS2 配置部分，选择本地和合作伙伴配置文件、加密和签名算法，以及是否压缩传输的信息。请注意以下几点：

   - 对于加密算法，`DES_EDE3_CBC`除非必须支持需要加密算法的旧版客户端，否则不要选择，因为这是一种较弱的加密算法。
   - 在使用连接器发送的 AS2 消息中，主题用作`subject` HTTP 标头属性。
   - 如果您选择创建不使用加密算法的连接器，则必须指定`HTTPS`为您的协议。
5. 在 MDN 配置部分中，指定以下信息：

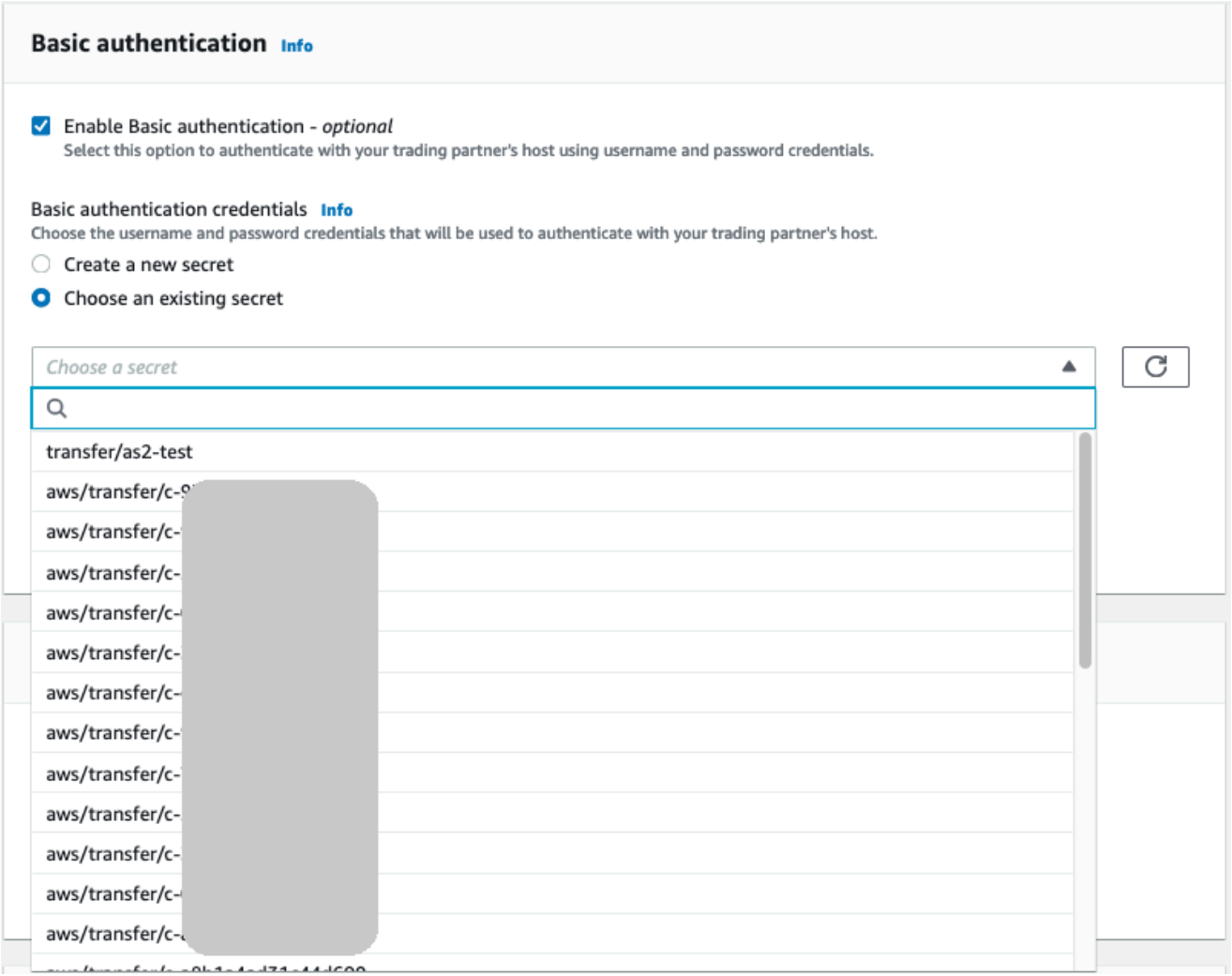
   - 请求 MDN — 您可以选择要求您的交易伙伴在通过 AS2 成功收到您的消息后向您发送 MDN。
   - 已签名的 MDN — 您可以选择要求对 MDN 进行签名。只有选择了请求 MDN，此选项才可用。
6. 在基本身份验证部分中，指定以下信息：

   - 要将登录凭证与出站消息一起发送，请选择启用基本身份验证。如果您不想在出站消息中发送任何凭证，请清除启用基本身份验证。
   - 如果您使用的是身份验证，请选择或创建密钥。

- 要创建新密钥，请选择创建新密钥，然后输入用户名和密码。这些凭证必须与连接到合作伙伴端点的用户相匹配。



- 要使用现有密钥，请选择选择现有秘钥，然后从下拉菜单中选择密钥。有关在 Secrets Manager 中创建格式正确的密钥的详细信息，请参阅启用 AS2 连接器的基本身份验证。



7. 确认所有设置后，选择创建连接器以创建连接器。

连接器页面会出现，其中新连接器的 ID 已添加到列表中。要查看连接器的详细信息，请参阅查看 AS2 连接器详细信息。

AS2 连接器算法

创建 AS2 连接器时，会将以下安全算法附加到该连接器。

类型	算法
TLS 密码	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256

类型	算法
	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384

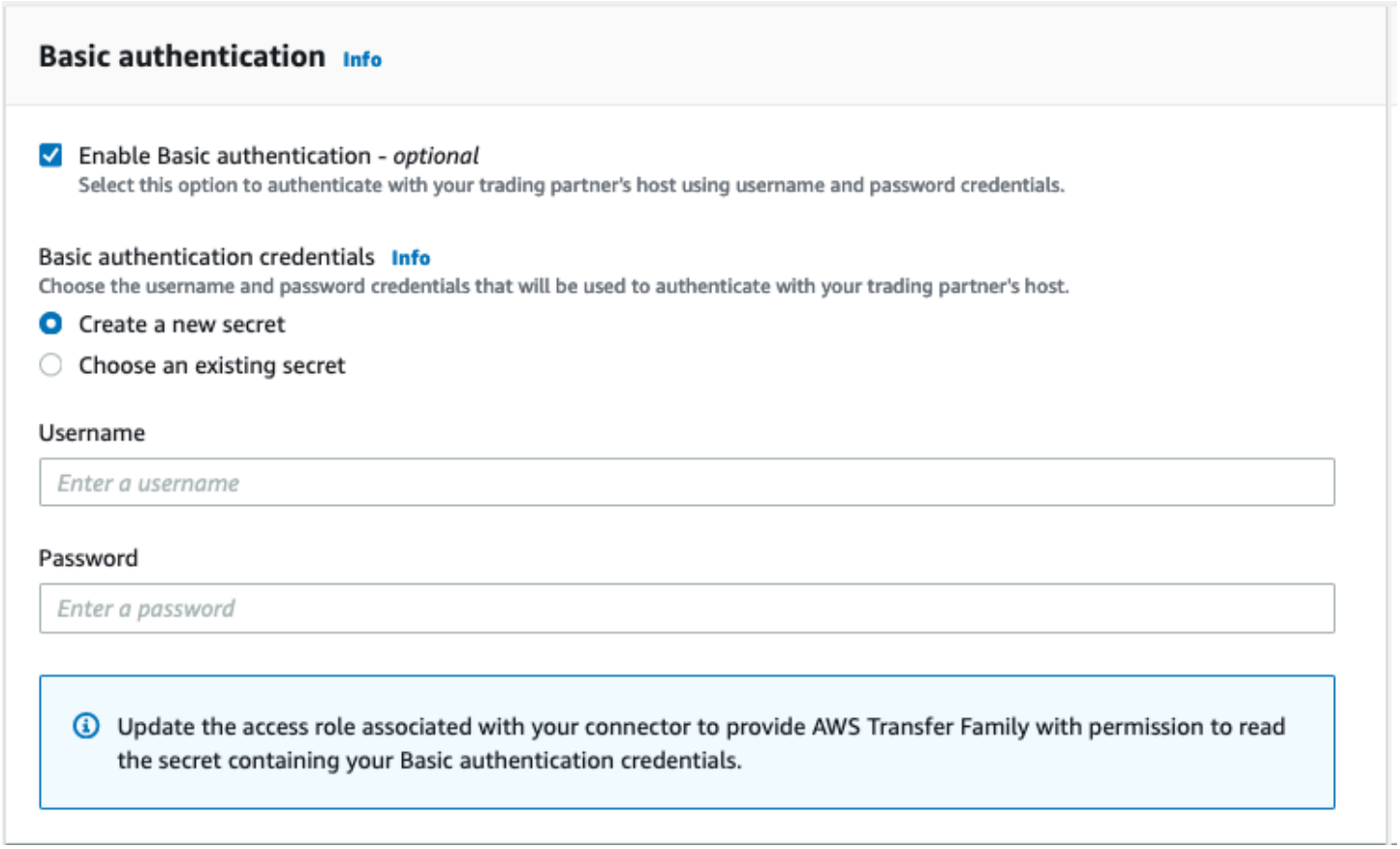
AS2 连接器的基本身份验证

创建或更新使用 AS2 协议的 Transfer Family 服务器时，可以为出站邮件添加基本身份验证。可以通过为连接器添加身份验证信息来完成此操作。

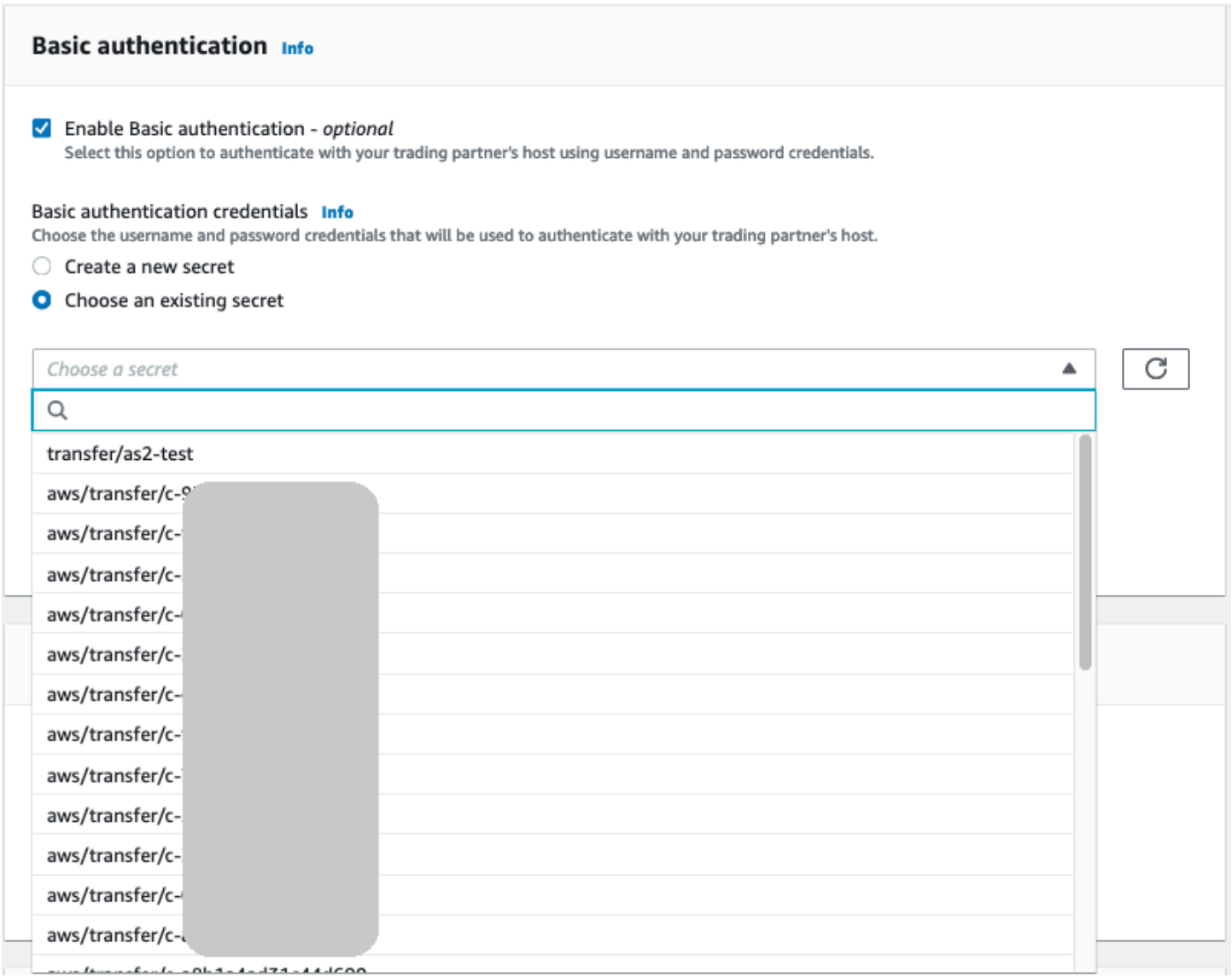
> **Note**
>
> 仅当您使用 HTTPS 时，基本身份验证才可用。

要对连接器使用身份验证，请在“基本身份验证”部分中选择“启用基本身份验证”。启用基本身份验证后，您可以选择创建新密钥，或使用现有密钥。无论哪种情况，密钥中的凭据都与使用此连接器的出站邮件一起发送。这些凭证必须与试图连接到贸易伙伴的远程端点的用户相匹配。

以下屏幕截图显示选中了“启用基本身份验证”，并选择了“创建新密钥”。做出这些选择后，您可以输入密钥的用户名和密码。



以下屏幕截图显示选中了“启用基本身份验证”，并选择了“创建现有密钥”。您的密钥必须为正确格式，如 启用 AS2 连接器的基本身份验证 所述。



启用 AS2 连接器的基本身份验证

为 AS2 连接器启用基本身份验证后，您可以在 Transfer Family 控制台中创建新密钥，也可以使用在 AWS Secrets Manager中创建的密钥。无论哪种情况，您的密钥都存储在 Secrets Manager 中。

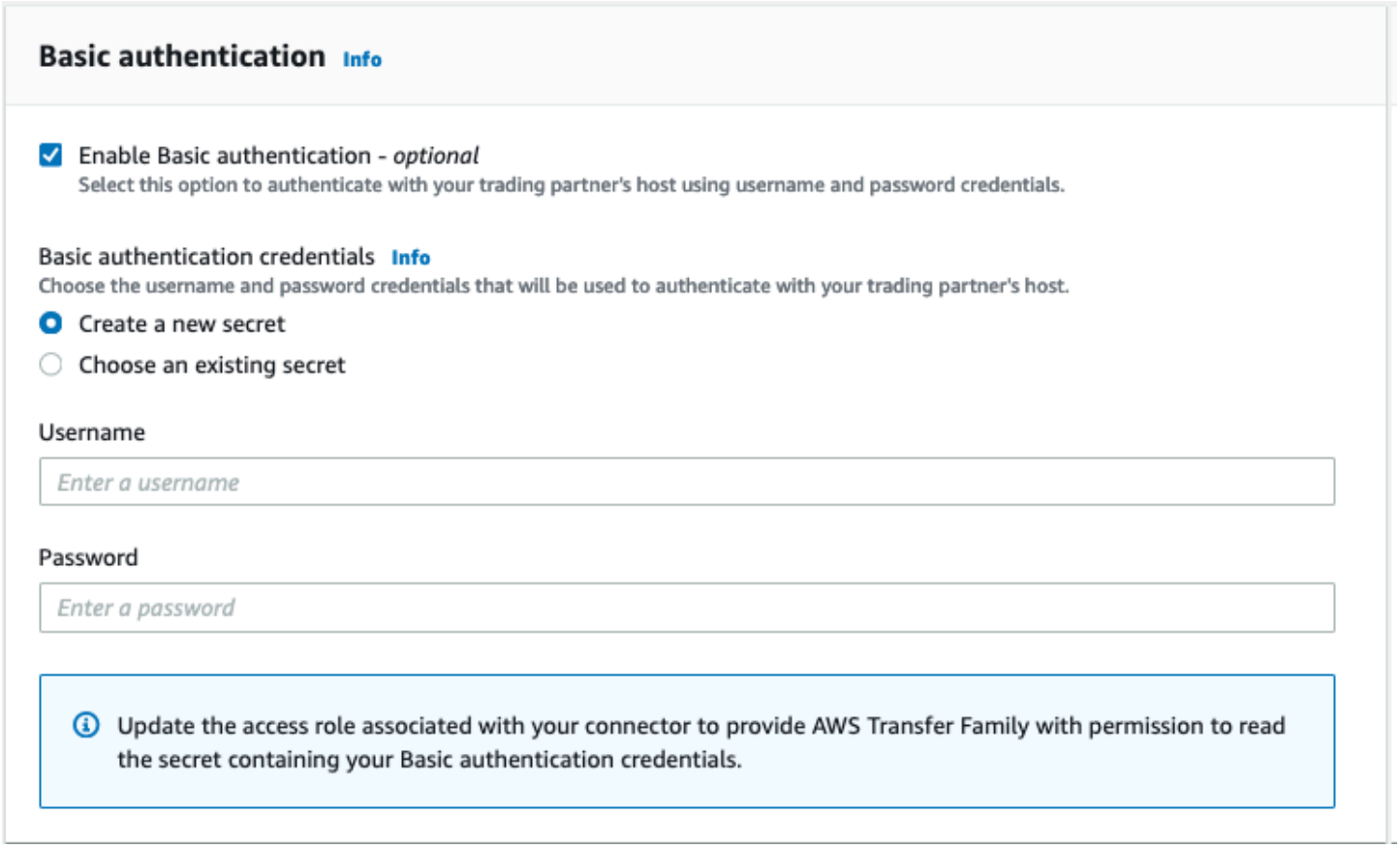
主题

- 在控制台中创建新的密钥
- 使用现有 密钥
- 在中创建密钥 AWS Secrets Manager

在控制台中创建新的密钥

当您在控制台中创建连接器时，您可以创建一个新的密钥。

要创建新密钥，请选择创建新密钥，然后输入用户名和密码。这些凭证必须与连接到合作伙伴端点的用户相匹配。

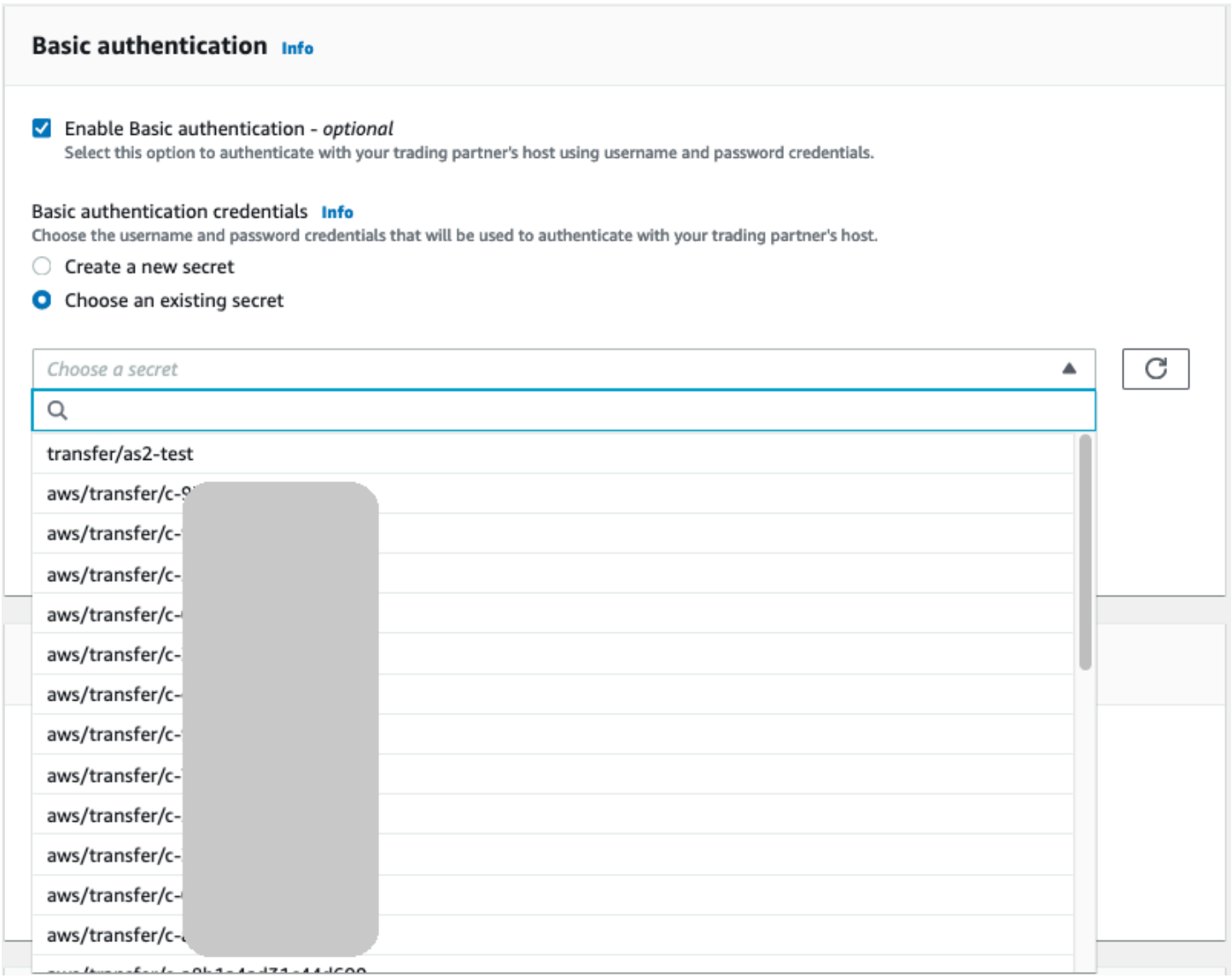


> **Note**
>
> 当您在控制台中创建新密钥时，密钥的名称将遵循以下命名约定：**`/aws/transfer/`*`connector-id`***，其中 *`connector-id`* 是您正在创建的连接器的 ID。当您试图在 AWS Secrets Manager中定位密钥时，请考虑这一点。

使用现有 密钥

当您在控制台中创建连接器时，您可以指定一个现有密钥。

要使用现有密钥，请选择选择现有秘钥，然后从下拉菜单中选择密钥。有关在 Secrets Manager 中创建格式正确的密钥的详细信息，请参阅在中创建密钥 AWS Secrets Manager。



在中创建密钥 AWS Secrets Manager

以下过程介绍了如何创建用于 AS2 连接器的相应密钥。

> **Note**
>
> 仅当您使用 HTTPS 时，基本身份验证才可用。

将用户凭证存储在 Secrets Manager 中进行 AS2 基本身份验证

1. 登录 AWS Management Console 并打开 AWS Secrets Manager 控制台，网址为 https://console.aws.amazon.com/secretsmanager/。

2. 在左侧导航窗格中，选择密钥。
3. 在密钥页面，选择存储新密钥。
4. 在选择密钥类型页面上，对于密钥类型，选择其他类型密钥。
5. 在键/值对部分，选择键/值选项卡。

   - 键 — 输入**Username**。
   - 值 — 输入有权连接到合作伙伴服务器的用户名。
6. 如果要提供密码，请选择添加行，然后在键/值对部分中，选择键/值选项卡。

   选择添加行，然后在键/值对部分选择键/值选项卡。

   - 键 — 输入**Password**。
   - 值 — 输入用户的密码。
7. 如果要提供私钥，请选择添加行，然后在密钥/值对部分，选择密钥/值选项卡。

   - 键 — 输入**PrivateKey**。
   - 值 — 输入用户的私有密钥。此值必须以 OpenSSH 格式存储，并且必须与在远程服务器中为该用户存储的公有密钥相对应。
8. 选择下一步。
9. 在配置密钥页面，输入密钥的名称和描述。建议对名称使用前缀 **aws/transfer/**。例如，您可以将密钥命名为 **aws/transfer/connector-1**。
10. 选择下一步，接受配置轮换页面的默认设置。然后选择下一步。
11. 在审核页面，选择存储以创建和存储密钥。

创建密钥后，您可以在创建连接器时选择密钥（请参阅配置 AS2 连接器）。在启用基本身份验证的步骤中，从可用密钥的下拉列表中选择密钥。

查看 AS2 连接器详细信息

您可以在 AWS Transfer Family 控制台中找到 AS2 AWS Transfer Family 连接器的详细信息和属性列表。AS2 连接器的属性包括其 URL、角色、配置文件、MDN、标签和监控指标。

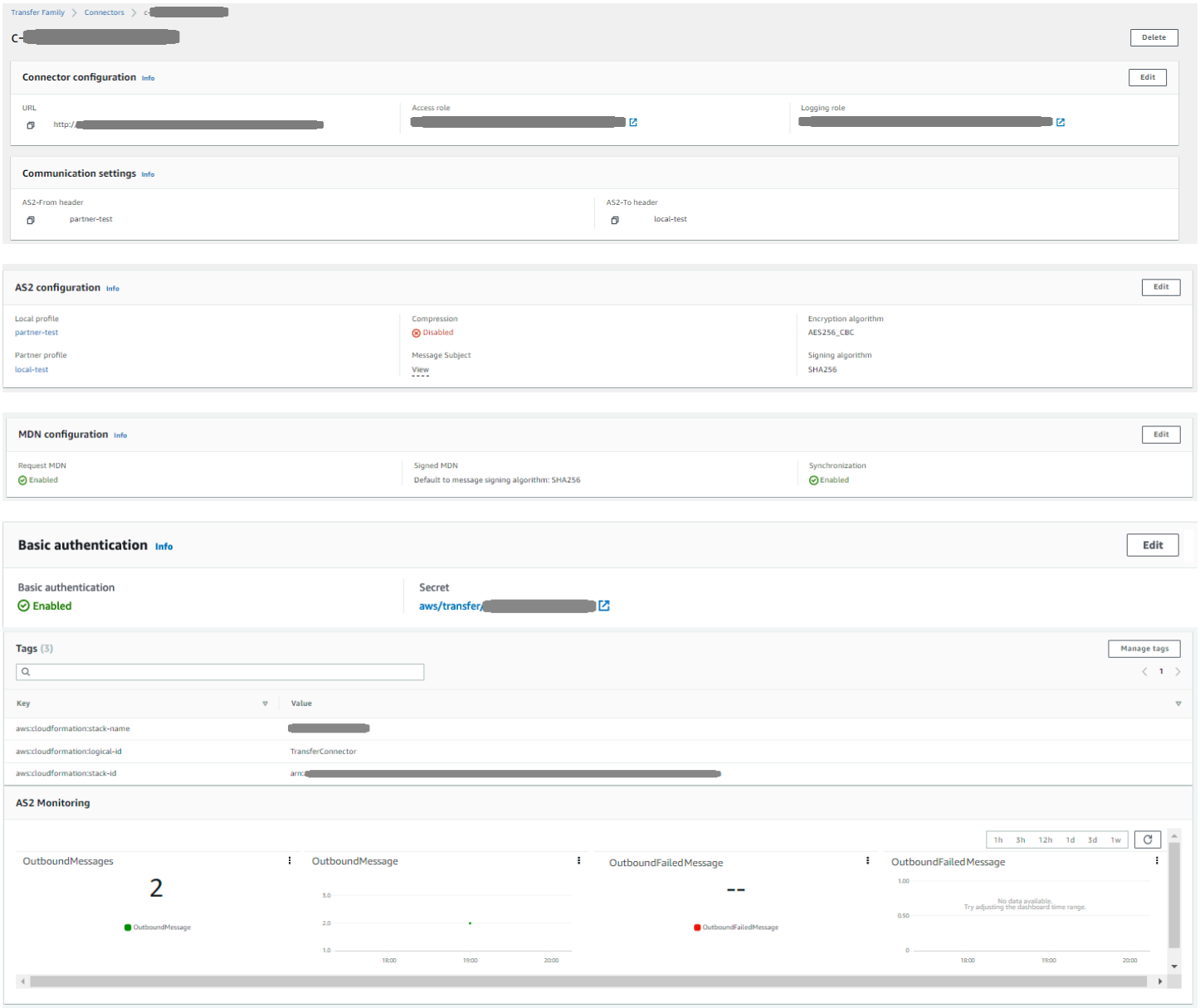
这是查看连接器详细信息的过程。

要查看连接器详细信息

1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。

2. 在左侧导航窗格中，选择 Connectors (连接器)。
3. 在“连接器 ID”列中选择标识符以查看所选连接器的详细信息页面。

通过选择“编辑”，可以在连接器的详细信息页面上更改 AS2 连接器的属性。



> **Note**
>
> 您可以通过运行以下 AWS Command Line Interface (AWS CLI 命令来获取其中的大部分信息，尽管格式不同：

```
aws transfer describe-connector --connector-id your-connector-id
```

有关更多信息，请参阅 API 参考中的DescribeConnector。

管理 AS2 合作伙伴

本主题讨论如何管理 AS2 证书、配置文件和协议。

导入 AS2 证书

Transfer Family AS2 流程使用证书密钥对传输的信息进行加密和签名。合作伙伴可以为两个目的使用相同的密钥，也可以为每个目的使用单独的密钥。如果您的通用加密密钥由受信任的第三方托管，以便在发生灾难或安全漏洞时可以对数据进行解密，我们建议您使用单独的签名密钥。通过使用单独的签名密钥（您不托管），您不会损害数字签名的不可否认性功能。

> **Note**
>
> AS2 证书的密钥长度必须至少为 2048 位，最多为 4096 位。

以下几点详细说明了在此过程中如何使用 AS2 证书。

- 入站的 AS2
  - 交易伙伴发送签名证书的公有密钥，该密钥将导入至合作伙伴配置文件中。
  - 本地方发送用于其加密和签名证书的公有密钥。然后，合作伙伴导入一个或多个私有密钥。本地方可以发送单独的证书密钥进行签名和加密，也可以选择将相同的密钥用于两种用途。
- 出站的 AS2
  - 合作伙伴发送其加密证书的公有密钥，该密钥将导入至合作伙伴配置文件中。
  - 本地方发送证书的公有密钥进行签名，并导入证书的私有密钥进行签名。
  - 如果您使用的是 HTTPS，则可以导入自签名的传输层安全 (TLS) 证书。

有关如何创建证书的详细信息，请参阅the section called “步骤 1：创建 AS2 证书”。

此步骤说明了如何使用 Transfer Family 控制台导入证书。如果要 AWS CLI 改用，请参阅the section called “第 3 步：将证书作为 Transfer Family 证书资源导入”。

要指定启用 AS2 的证书

1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在左侧导航窗格中，在 AS2 交易合作伙伴下选择证书。
3. 选择导入证书。
4. 在证书描述部分，输入易于识别的证书名称。确保您可以通过其描述来识别证书的用途。此外，选择证书的角色。
5. 在证书内容部分，提供交易伙伴提供的公有证书，或本地证书的公有和私有密钥。
6. 在证书使用部分中，选择此证书的用途。它可以用于加密、签名或两者兼而有之。

   > **Note**
   >
   > 如果您选择加密和签名进行使用，Transfer Family 会创建两个相同的证书（每个证书都有自己的 ID）：一个使用值为`ENCRYPTION`，另一个使用值为`SIGNING`。

7. 在证书内容部分填写相应的详细信息。
   - 如果选择自签名证书，则不提供证书链。
   - 粘贴证书的内容。
   - 如果证书不是自签名证书，请提供证书链。
   - 如果此证书是本地证书，请粘贴其私有密钥。
8. 选择导入证书以完成该流程并保存导入证书的详细信息。

> **Note**
>
> TLS 证书只能作为合作伙伴的公共证书导入。如果您选择合作伙伴提供的公共证书，然后为使用选择传输层安全 (TLS)，则会收到警告。此外，TLS 证书必须是自签名的（也就是说，您必须选择自签名证书才能导入 TLS 证书）。

AS2 证书轮换

通常，证书的有效期为六个月至一年。您可能已经设置想要保留更长时间的配置文件。为此，Transfer Family 提供了证书轮换功能。您可以为一个配置文件指定多个证书，以便您可以连续多年使用该配置文件。Transfer Family 使用证书进行签名（可选）和加密（必填）。如果您愿意，可以为这两个目的指定一个证书。

证书轮换是将即将过期的旧证书替换为较新的证书的过程。过渡是渐进的，以避免协议中的合作伙伴尚未为出站传输配置新证书，或者可能在使用新证书的时期发送使用旧证书签名或加密的有效负载，从而避免中断传输。新旧证书均有效的中间期称为宽限期。

X.509 证书有`Not Before`日期和`Not After`日期。但是，这些参数可能无法为管理员提供足够的控制。Transfer Family 提供`Active Date`和`Inactive Date`设置以控制哪些证书用于出站负载，哪些证书被接受用于入站负载。

出站证书选择使用转移日期之前的最大值作为`Inactive Date`。入站流程接受`Not Before`和`Not After`范围内的证书，以及`Active Date`和`Inactive Date`范围内的证书。

下表描述了为单个配置文件配置两个证书的一种可能方法。

两个证书轮换

名称	NOT BEFORE（由证书颁发机构控制）	ACTIVE DATE（由 Transfer Family 设置）	INACTIVE DATE（由 Transfer Family 设置）	NOT AFTER（由证书颁发机构设置）
证书 1（旧证书）	2019-11-01	2020-01-01	2020-12-31	2024-01-01
证书 2（新证书）	2020-11-01	2020-06-01	2021-06-01	2025-01-01

请注意以下几点：

- 为证书指定`Active Date`和`Inactive Date`时，该范围必须介于`Not Before`和`Not After`之间。
- 我们建议您为每个配置文件配置多个证书，确保所有证书的有效日期范围涵盖您要使用该配置文件的时间。
- 我们建议您在旧证书变为非活动状态和新证书处于活动状态之间指定一段宽限时间。在前面的示例中，第一个证书直到 2020 年 12 月 31 才处于非活动状态，而第二个证书在 2020 年 6 月 1 日生效，提供了 6 个月的宽限期。在 2020 年 6 月 1 日至 2020 年 12 月 31 日期间，两个证书均处于活动状态。

创建 AS2 配置文件

使用此步骤创建本地和合作伙伴配置文件。此步骤说明了如何使用 Transfer Family 控制台创建 AS2 配置文件。如果要改用 AWS CLI ，请参阅 the section called “第 4 步：为您和您的交易伙伴创建配置文件”。

要创建 AS2 配置文件

1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在左侧导航窗格的 AS2 交易伙伴下，选择配置文件，然后选择创建配置文件。
3. 在配置文件配置部分，输入配置文件的 AS2 ID。此值用于特定于 AS2 协议的 HTTP 标头`as2-from`和`as2-to`以用于标识交易伙伴关系，后者决定要使用的证书，依此类推。
4. 在配置文件类型部分，选择本地配置文件或合作伙伴配置文件。
5. 在证书部分，从下拉菜单中选择一个或多个证书。

   > **Note**
   >
   > 如果要导入未在下拉菜单中列出的证书，请选择导入新证书。这将在导入证书屏幕上打开一个新的浏览器窗口。有关导入证书的步骤，请参阅导入 AS2 证书。

6. （可选）在标签部分中，指定一个或多个键值对以帮助标识此配置文件。
7. 选择创建配置文件以完成该流程并保存新的配置文件。

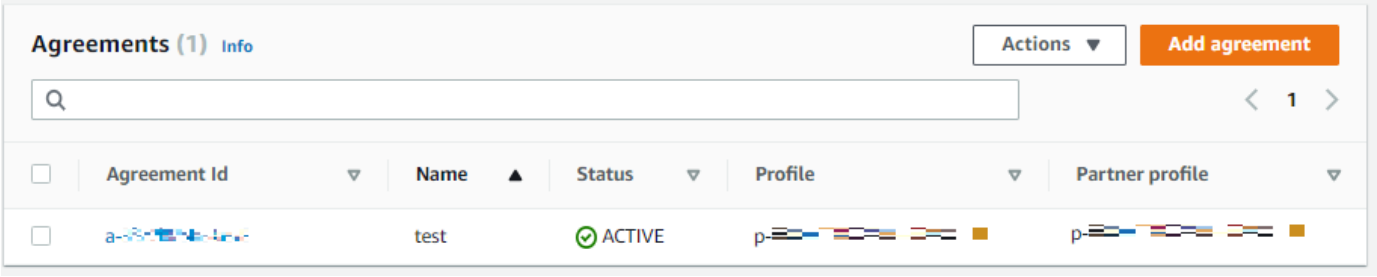
创建 AS2 协议

协议与 Transfer Family 服务器相关联。它们为使用 AS2 协议使用 Transfer Family 交换消息或文件的交易伙伴以及入站传输（将 AS2 文件从合作伙伴拥有的外部源发送到 Transfer Family 服务器）提供了详细信息。

此步骤说明了如何使用 Transfer Family 控制台创建 AS2 协议。如果要 AWS CLI 改用，请参阅the section called “第 5 步：创建您与合作伙伴之间的协议”。

要为 Transfer Family 服务器创建协议

1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在左侧导航窗格中，选择服务器，然后选择使用 AS2 协议的服务器。
3. 在服务器详细信息页面上，向下滚动到协议部分。



4. 选择添加协议。
5. 填写协议参数，如下所示：

   a. 在协议配置部分中，输入描述性名称。确保您可以通过协议名称来识别协议的目的。此外，还要设置协议的状态：活动（默认选中）或非活动。

   b. 在通信配置部分，选择本地配置文件和合作伙伴配置文件。

   c. 在收件箱文件夹配置部分，选择用于存储传入文件的 Amazon S3 存储桶和可以访问该存储桶的 IAM 角色。或者，您可以输入用于在存储桶中存储文件的前缀（文件夹）。

      例如，如果您为存储桶输入**DOC-EXAMPLE-BUCKET**，为前缀输入**incoming**，则传入的文件将保存到该`/DOC-EXAMPLE-BUCKET/incoming`文件夹。

   d. （可选）在标签部分中，添加标签。

   e. 输入协议的所有信息后，选择创建协议。

新协议显示在服务器详细信息页面的协议部分。

发送和接收 AS2 消息

本节介绍发送和接收 AS2 消息的过程。它还提供了与 AS2 消息相关的文件名和位置的详细信息。

下表列出了 AS2 消息的可用加密算法以及何时可以使用这些算法。

加密算法	HTTP	HTTPS	注意事项
AES128_CBC	支持	支持	
AES192_CBC	支持	支持	
AES256_CBC	支持	支持	

加密算法	HTTP	HTTPS	注意事项
DES_EDE3_CBC	支持	支持	只有在必须支持需要此算法的旧版客户端时才使用此算法，因为它是一种弱加密算法。
NONE	不支持	支持	如果您要向 Transfer Family 服务器发送消息，则只能选择NONE是否使用应用程序负载均衡器 (ALB)。

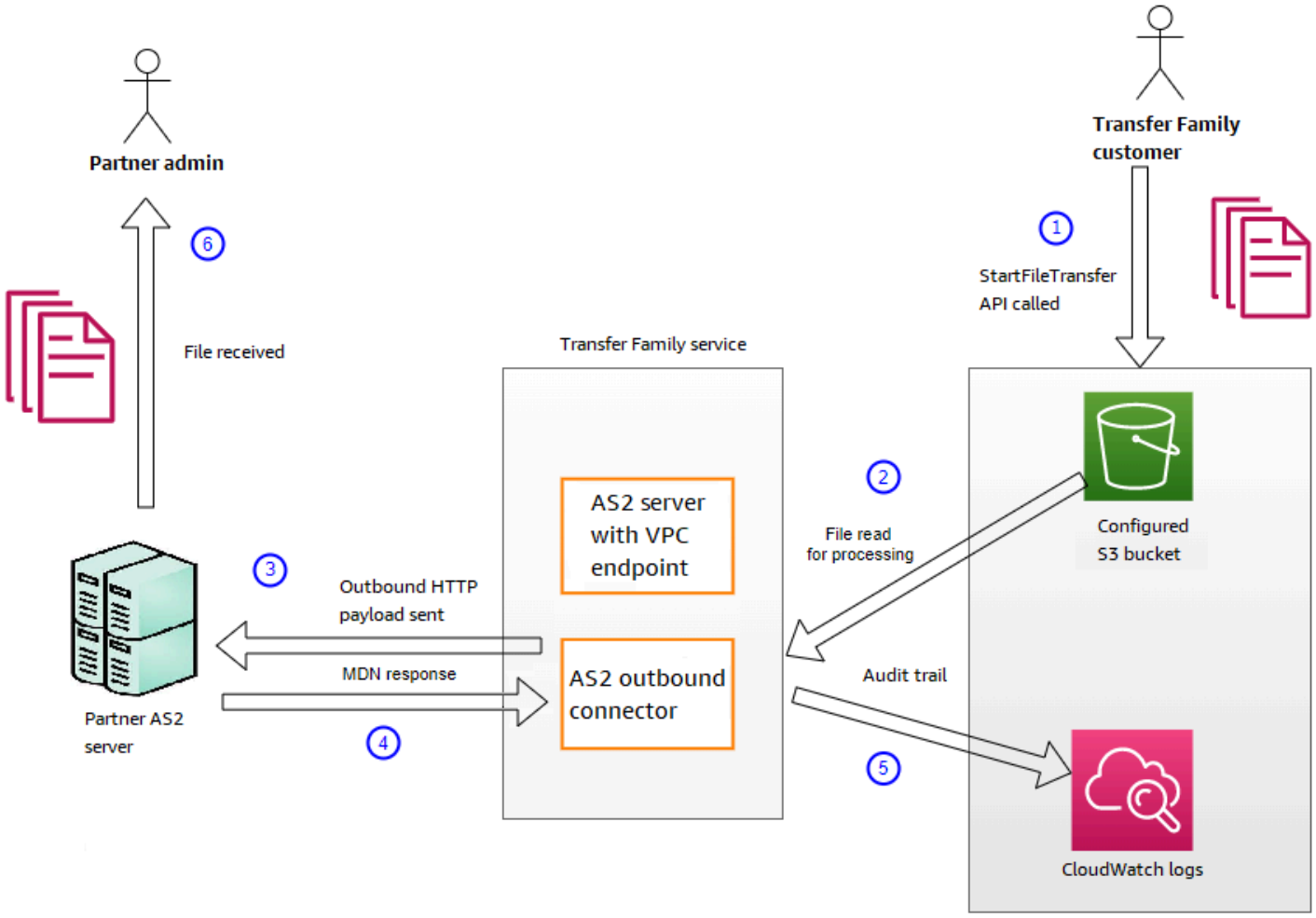
主题

发送 AS2 消息流程

出站进程定义为从 AWS 外部客户端或服务发送的消息或文件。出站消息的顺序如下：

1. 管理员调用 `start-file-transfer` AWS Command Line Interface (AWS CLI) 命令或 `StartFileTransfer` API 操作。此操作引用connector配置。
2. Transfer Family 检测到新的文件请求并定位该文件。文件经过压缩、签名和加密。
3. 传输 HTTP 客户端执行 HTTP POST 请求，将有效负载传输到合作伙伴的 AS2 服务器。
4. 该流程返回已签名的 MDN 响应，该响应与 HTTP 响应（同步 MDN）内联。
5. 当文件在不同的传输阶段之间移动时，该流程会向客户提供 MDN 响应接收和处理详细信息。

6. 远程 AS2 服务器将解密并经过验证的文件提供给合作伙伴管理员。



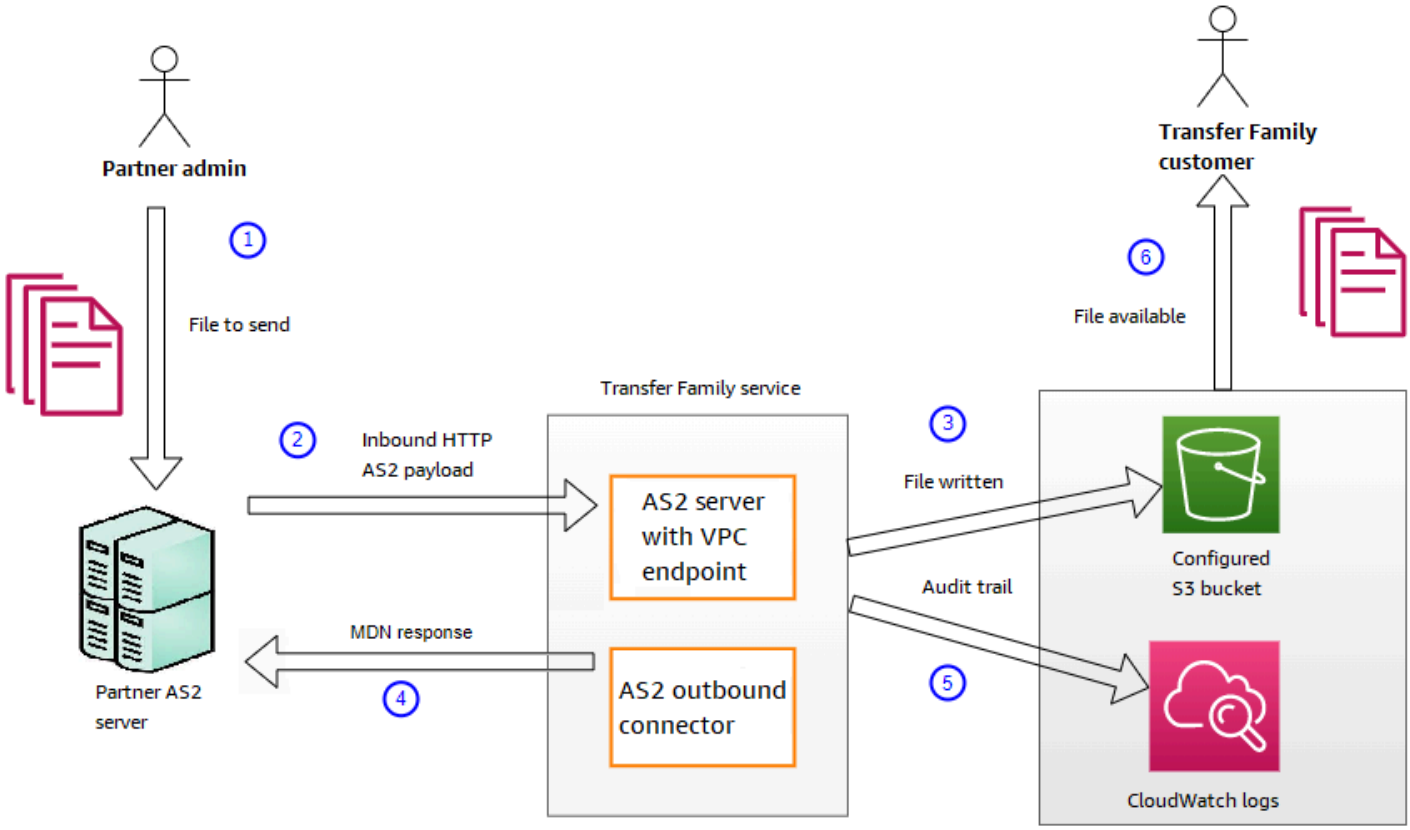
AS2 处理支持许多 RFC 4130 协议，重点关注常见使用案例并与启用 AS2 的现有服务器实现集成。有关支持的配置详细信息，请参阅AS2 支持的配置。

接收 AS2 消息流程

入站流程定义为正在传输到 AWS Transfer Family 服务器的消息或文件。入站消息的顺序如下：

1. 管理员或自动流程在合作伙伴的远程 AS2 服务器上启动 AS2 文件传输。
2. 合作伙伴的远程 AS2 服务器对文件内容进行签名和加密，然后向 Transfer Family 上托管的 AS2 入站端点发送 HTTP POST 请求。
3. Transfer Family 使用服务器、合作伙伴、证书和协议的配置值，解密并验证 AS2 有效负载。文件内容存储在已配置的 Amazon S3 文件存储中。

4. 已签名的 MDN 响应应与 HTTP 响应内联返回，或者通过单独的 HTTP POST 请求异步返回原始服务器。
5. 审计记录已写入Amazon， CloudWatch 其中包含有关交易所的详细信息。
6. 解密后的文件位于名为`inbox/processed`的文件夹中。



通过 HTTPS 发送和接收 AS2 消息

本节介绍如何配置使用 AS2 协议通过 HTTPS 发送和接收消息的 Transfer Family 服务器。

通过 HTTPS 发送 AS2 消息

要使用 HTTPS 发送 AS2 消息，请使用以下信息创建一个连接器：

- 对于网址，请指定 HTTPS URL
- 对于加密算法，请选择任何可用的算法。

> **Note**
>
> 要在不使用加密（即您选择NONE加密算法）的情况下向 Transfer Family 服务器发送消息，则必须使用应用程序负载均衡器 (ALB)。

- 提供连接器的其余值，如 配置 AS2 连接器 中所述。

通过 HTTPS 接收 AS2 消息

AWS Transfer Family AS2 服务器目前仅提供通过端口 5080 的 HTTP 传输。但是，您可以使用自己选择的端口和证书在您的 Transfer Family 服务器 VPC 端点前的负载均衡器上终止 TLS。使用这种方法，您可以让传入的 AS2 消息使用 HTTPS。

先决条件

- VPC 必须与您的 Transfer AWS 区域 Family 服务器位于同一个服务器中。
- 您的 VPC 的子网必须位于您要在其中使用服务器的可用区内。

> **Note**
>
> 每台 Transfer Family 服务器最多可以支持三个可用区。

- 在与您的服务器相同的区域中最多分配三个弹性 IP 地址。或者，您可以选择自带 IP 地址范围（BYOIP）。

> **Note**
>
> 弹性 IP 地址的数量必须与您用于服务器端点的可用区域数量相匹配。

配置网络负载均衡器

在您的 VPC 中设置面向互联网的网络负载均衡器（NLB）。

创建网络负载均衡器并将服务器的 VPC 端点定义为负载均衡器的目标

1. 在 https://console.aws.amazon.com/ec2/ 中打开 Amazon Elastic Compute Cloud 控制台。
2. 在导航窗格中，选择负载均衡器，然后选择创建负载均衡器。

3. 在网络负载均衡器下，选择创建。
4. 在基本配置部分，输入以下信息：
   - 对于名称，为负载均衡器输入一个描述性名称。
   - 对于 Scheme，选择 Internet-facing。
   - 在 IP 地址类型中，选择 IPv4。
5. 在网络映射部分中，输入以下信息：
   - 对于 VPC，请选择您已创建的虚拟私有云 (VPC)。
   - 在映射下，选择与公有子网关联的可用区，这些子网位于您用于服务器端点的同一 VPC 中。
   - 对于每个子网的IPv4 地址，请选择您分配的弹性 IP 地址之一。
6. 在侦听器和路由部分中，输入以下信息：
   - 对于协议，选择 TLS。
   - 对于端口，输入 **5080**。
   - 对于默认操作，选择创建目标组。有关创建新目标组的详细信息，请参阅 创建目标组。

   创建目标组后，在默认操作字段中输入其名称。
7. 在安全侦听器设置部分，在默认 SSL/TLS 证书区域中选择您的证书。
8. 选择创建负载均衡器以创建您的 NLB。
9. （可选，但推荐）打开网络负载均衡器的访问日志以保持完整的审计跟踪记录，如网络负载均衡器的访问日志中所述。

   我们建议执行此步骤，因为 TLS 连接已在 NLB 终止。因此，反映在您的 Transfer Family AS2 CloudWatch 日志组中的源 IP 地址是 NLB 的私有 IP 地址，而不是贸易伙伴的外部 IP 地址。

设置负载均衡器后，客户端通过自定义端口侦听器与负载均衡器进行通信。然后，负载均衡器通过端口 5080 与服务器通信。

创建目标组

1. 在前面的过程中选择创建目标组后，您将进入新目标组的指定组详细信息页面。
2. 在基本配置部分，输入以下信息。
   - 在选择目标类型中，选择 IP 地址。

- 对于目标组名称，输入目标组的名称。
- 对于协议，选择 TCP。
- 对于端口，输入 **5080**。
- 在 IP 地址类型中，选择 IPv4。
- 对于 VPC，选择已为 Transfer Family AS2 服务器创建的 VPC。

3. 在运行状况检查部分，选择 TCP 作为运行状况检查协议。
4. 选择下一步。
5. 在注册目标页面，输入以下信息：

   - 对于网络，请确认已指定您为 Transfer Family AS2 服务器创建的 VPC。
   - 对于 IPv4 地址，请输入 Transfer Family AS2 服务器端点的私有 IPv4 地址。

     如果您的服务器有多个端点，请选择添加 IPv4 地址以添加另一行，用于输入另一个 IPv4 地址。重复此过程，直到输入服务器所有端点的私有 IP 地址。
   - 确保端口设置为**5080**。
   - 选择包含如下待处理事项，将您的条目添加到审核目标部分。
6. 在查看目标部分，查看您的 IP 目标。
7. 选择创建目标组，然后返回之前创建 NLB 的过程，并在指示的位置输入新的目标组。

测试从弹性 IP 地址访问服务器

使用弹性 IP 地址或网络负载均衡器的 DNS 名称通过自定义端口连接到服务器。

> **Important**
>
> 使用负载均衡器上配置的子网的网络访问控制列表（网络 ACL）管理从客户端 IP 地址对服务器的访问。网络 ACL 权限是在子网级别设置的，因此这些规则适用于使用该子网的所有资源。您无法使用安全组控制来自客户端 IP 地址的访问，因为负载均衡器的目标类型设置为 IP 地址而不是实例。因此，负载均衡器不保留源 IP 地址。如果网络负载均衡器的运行状况检查失败，则意味着负载均衡器无法连接到服务器端点。要对此问题进行故障排除，请检查以下步骤：
>
> - 确认服务器端点的关联安全组允许来自负载均衡器上配置的子网的入站连接。负载均衡器必须能够通过端口 5080 连接到服务器端点。
> - 确认服务器的状态为联机。

使用 AS2 连接器传输文件

AS2 连接器在贸易伙伴之间建立关系，将 AS2 消息从 Transfer Family 服务器传输到合作伙伴拥有的外部目的地。

您可以使用 Transfer Family 通过引用连接器 ID 和文件路径发送 AS2 消息，如以下 `start-file-transfer` AWS Command Line Interface (AWS CLI) 命令所示：

```
aws transfer start-file-transfer --connector-id c-1234567890abcdef0 \
--send-file-paths "/DOC-EXAMPLE-SOURCE-BUCKET/myfile1.txt" "/DOC-EXAMPLE-SOURCE-BUCKET/
myfile2.txt"
```

要获取连接器详细信息，请运行以下命令：

```
aws transfer list-connectors
```

该 `list-connectors` 命令会返回连接器的连接器 ID、URL 和 Amazon 资源名称（ARN）。

要返回特定连接器的属性，请使用要使用的 ID 运行以下命令：

```
aws transfer describe-connector --connector-id your-connector-id
```

`describe-connector`命令返回连接器的所有属性，包括其 URL、角色、配置文件、消息处置通知 (MDN)、标签和监控指标。

您可以通过查看 JSON 和 MDN 文件来确认合作伙伴已成功接收文件。这些文件是根据文件名和位置中描述的约定命名的。如果您在创建连接器时配置了日志记录角色，则还可以检查 CloudWatch 日志中是否有 AS2 消息的状态。

要查看 AS2 连接器的详细信息，请参阅 查看 AS2 连接器详细信息。有关创建 AS2 连接器的更多信息，请参阅 配置 AS2 连接器。

文件名和位置

本部分讨论 AS2 传输的文件命名约定。

对于入站文件传输，需要注意以下方面：

- 您可以在协议中指定基本目录。基本目录是 Amazon S3 存储桶名称和前缀（如果有）的组合。例如，`/DOC-EXAMPLE-BUCKET/AS2-folder`。

- 如果成功处理了传入的文件，则该文件（以及相应的 JSON 文件）将保存至该/processed文件夹。例如，/DOC-EXAMPLE-BUCKET/AS2-folder/processed。

  JSON 文件包含以下字段：

  - agreement-id
  - as2-from
  - as2-to
  - as2-message-id
  - transfer-id
  - client-ip
  - connector-id
  - failure-message
  - file-path
  - message-subject
  - mdn-message-id
  - mdn-subject
  - requester-file-name
  - requester-content-type
  - server-id
  - status-code
  - failure-code
  - transfer-size
- 如果无法成功处理传入文件，则该文件（以及相应的 JSON 文件）将保存至该/failed文件夹。例如，/DOC-EXAMPLE-BUCKET/AS2-folder/failed。
- 传输的文件存储在processed文件夹中，名为*original_filename*.*messageId*.*original_extension*。也就是说，传输的消息 ID 会附加到文件名之后，在文件的原始扩展名之前。
- 已创建 JSON 文件并将其另存为*original_filename*.*messageId*.*original_extension*.json。除了要添加的消息 ID 外，还会在传输的文件名后面附加该字符串.json。

- 消息处置通知 (MDN) 文件已创建并另存为*original_filename*.*messageId*.*original_extension*.mdn。除了要添加的消息 ID 外，还会在传输的文件名后面附加该字符串.mdn。
- 如果存在名为ExampleFileInS3Payload.dat的入站文件，则会创建以下文件：
  - File — ExampleFileInS3Payload.c4d6b6c7-23ea-4b8c-9ada-0cb811dc8b35@44313c54b0a46a36.
  - JSON — ExampleFileInS3Payload.c4d6b6c7-23ea-4b8c-9ada-0cb811dc8b35@44313c54b0a46a36.
  - MDN — ExampleFileInS3Payload.c4d6b6c7-23ea-4b8c-9ada-0cb811dc8b35@44313c54b0a46a36.

对于出站传输，命名类似，不同之处在于没有传入的消息文件，而且，已传输消息的传输 ID 会添加到文件名中。传输 ID 由StartFileTransfer API 操作返回（或者当其他流程或脚本调用此操作时）。

- transfer-id是与文件传输关联的标识符。作为StartFileTransfer调用一部分的所有请求共享transfer-id。
- 基本目录与您用于源文件的路径相同。也就是说，基目录是您在 StartFileTransfer API 操作或start-file-transfer AWS CLI 命令中指定的路径。例如：

```
aws transfer start-file-transfer --send-file-paths /DOC-EXAMPLE-BUCKET/AS2-folder/
file-to-send.txt
```

  如果运行此命令，MDN 和 JSON 文件将保存在/DOC-EXAMPLE-BUCKET/AS2-folder/processed中（对于成功传输）或/DOC-EXAMPLE-BUCKET/AS2-folder/failed（对于不成功传输）。
- 已创建 JSON 文件并将其另存为*original_filename*.*transferId*.*messageId*.*original_extension*.json。
- 已创建 MDN 文件并将其另存为*original_filename*.*transferId*.*messageId*.*original_extension*.mdn。
- 如果存在名为ExampleFileOutTestOutboundSyncMdn.dat的出站文件，则会创建以下文件：
  - JSON — ExampleFileOutTestOutboundSyncMdn.dedf4601-4e90-4043-b16b-579af35e0d83.fbe18db8-7361-42ff-8ab6-49ec1e435f34@c9c705f0baaaabaa.dat.j
  - MDN — ExampleFileOutTestOutboundSyncMdn.dedf4601-4e90-4043-b16b-579af35e0d83.fbe18db8-7361-42ff-8ab6-49ec1e435f34@c9c705f0baaaabaa.dat.m

您还可以查看 CloudWatch 日志以查看转账的详细信息，包括任何失败的转账。

状态代码

下表列出了您或您的合作伙伴发送 AS2 消息时可以记录到 CloudWatch 日志中的所有状态代码。不同的消息处理步骤适用于不同的消息类型，并且仅用于监控。“已完成” 和 “失败” 状态表示处理的最后一步，在 JSON 文件中可见。

代码	描述	处理完成了吗？
处理	该消息正在转换为其最终格式。例如，解压缩和解密步骤都具有此状态。	不支持
MDN_TRANSM	消息处理正在发送 MDN 响应。	不支持
MDN_RECEIVE	消息处理正在收到 MDN 响应。	不支持
COMPLETED	消息处理已成功完成。此状态包括为入站消息或出站消息的 MDN 验证发送 MDN 的时间。	支持
FAILED	消息处理失败。有关错误代码的列表，请参阅AS2 错误代码。	支持

示例 JSON 文件

本部分列出了入站和出站传输的示例 JSON 文件，包括传输成功和传输失败的示例文件。

传输成功的示例出站文件：

```
{
  "requester-content-type": "application/octet-stream",
  "mesage-subject": "File xyzTest from MyCompany_OID to partner YourCompany",
  "requester-file-name": "TestOutboundSyncMdn-9lmCr79hV.dat",
```

```
  "as2-from": "MyCompany_OID",
  "connector-id": "c-c21c63ceaaf34d99b",
  "status-code": "COMPLETED",
  "disposition": "automatic-action/MDN-sent-automatically; processed",
  "transfer-size": 3198,
  "mdn-message-id": "OPENAS2-11072022063009+0000-df865189-1450-435b-9b8d-
d8bc0cee97fd@PartnerA_OID_MyCompany_OID",
  "mdn-subject": "Message be18db8-7361-42ff-8ab6-49ec1e435f34@c9c705f0baaaabaa has been
 accepted",
  "as2-to": "PartnerA_OID",
  "transfer-id": "dedf4601-4e90-4043-b16b-579af35e0d83",
  "file-path": "/DOC-EXAMPLE-BUCKET/as2testcell0000/openAs2/
TestOutboundSyncMdn-9lmCr79hV.dat",
  "as2-message-id": "fbe18db8-7361-42ff-8ab6-49ec1e435f34@c9c705f0baaaabaa",
  "timestamp": "2022-07-11T06:30:10.791274Z"
}
```

传输失败的示例出站文件：

```
{
  "failure-code": "HTTP_ERROR_RESPONSE_FROM_PARTNER",
  "status-code": "FAILED",
  "requester-content-type": "application/octet-stream",
  "subject": "Test run from Id da86e74d6e57464aae1a55b8596bad0a to partner
 9f8474d7714e476e8a46ce8c93a48c6c",
  "transfer-size": 3198,
  "requester-file-name": "openAs2TestOutboundWrongAs2Ids-necco-3VYn5n8wE.dat",
  "as2-message-id": "9a9cc9ab-7893-4cb6-992a-5ed8b90775ff@718de4cec1374598",
  "failure-message": "http://Test123456789.us-east-1.elb.amazonaws.com:10080 returned
 status 500 for message with ID 9a9cc9ab-7893-4cb6-992a-5ed8b90775ff@718de4cec1374598",
  "transfer-id": "07bd3e07-a652-4cc6-9412-73ffdb97ab92",
  "connector-id": "c-056e15cc851f4b2e9",
  "file-path": "/testbucket-4c1tq6ohjt9y/as2IntegCell0002/openAs2/
openAs2TestOutboundWrongAs2Ids-necco-3VYn5n8wE.dat",
  "timestamp": "2022-07-11T21:17:24.802378Z"
}
```

传输成功的示例进站文件：

```
{
  "requester-content-type": "application/EDI-X12",
  "subject": "File openAs2TestInboundAsyncMdn-necco-5Ab6bTfCO.dat sent from MyCompany
 to PartnerA",
```

```
  "client-ip": "10.0.109.105",
  "requester-file-name": "openAs2TestInboundAsyncMdn-necco-5Ab6bTfCO.dat",
  "as2-from": "MyCompany_OID",
  "status-code": "COMPLETED",
  "disposition": "automatic-action/MDN-sent-automatically; processed",
  "transfer-size": 1050,
  "mdn-subject": "Message Disposition Notification",
  "as2-message-id": "OPENAS2-11072022233606+0000-5dab0452-0ca1-4f9b-b622-
fba84effff3c@MyCompany_OID_PartnerA_OID",
  "as2-to": "PartnerA_OID",
  "agreement-id": "a-f5c5cbea5f7741988",
  "file-path": "processed/openAs2TestInboundAsyncMdn-
necco-5Ab6bTfCO.OPENAS2-11072022233606+0000-5dab0452-0ca1-4f9b-b622-
fba84effff3c@MyCompany_OID_PartnerA_OID.dat",
  "server-id": "s-5f7422b04c2447ef9",
  "timestamp": "2022-07-11T23:36:36.105030Z"
}
```

传输失败的示例进站文件：

```
{
  "failure-code": "INVALID_REQUEST",
  "status-code": "FAILED",
  "subject": "Sending a request from InboundHttpClientTests",
  "client-ip": "10.0.117.27",
  "as2-message-id": "testFailedLogs-TestRunConfig-Default-inbound-direct-
integ-0c97ee55-af56-4988-b7b4-a3e0576f8f9c@necco",
  "as2-to": "0beff6af56c548f28b0e78841dce44f9",
  "failure-message": "Unsupported date format: 2022/123/456T",
  "agreement-id": "a-0ceec8ca0a3348d6a",
  "as2-from": "ab91a398aed0422d9dd1362710213880",
  "file-path": "failed/01187f15-523c-43ac-9fd6-51b5ad2b08f3.testFailedLogs-
TestRunConfig-Default-inbound-direct-integ-0c97ee55-af56-4988-b7b4-a3e0576f8f9c@necco",
  "server-id": "s-0582af12e44540b9b",
  "timestamp": "2022-07-11T06:30:03.662939Z"
}
```

监控 AS2 的使用情况

您可以使用亚马逊 CloudWatch 和AWS CloudTrail监控 AS2 的活动。要查看其他 Transfer Family 服务器指标，请参阅 Amazon CloudWatch 正在登录 AWS Transfer Family

AS2 指标

指标	描述
InboundMessage	成功从交易伙伴处收到的 AS2 消息总数。 单位：计数 时间：5 分钟
InboundFailedMessage	未成功从交易伙伴处收到的 AS2 消息总数。也就是说，交易伙伴发送了一条消息，但是 Transfer Family 服务器无法成功处理该消息。 单位：计数 时间：5 分钟
OutboundMessage	成功从 Transfer Family 服务器向交易伙伴发送的 AS2 消息总数。 单位：计数 时间 = 5 分钟
OutboundFailedMessage	未成功发送给交易伙伴的 AS2 消息总数。也就是说，它们是从 Transfer Family 服务器发送的，但交易伙伴没有成功接收。 单位：计数 时间：5 分钟

AS2 状态码

下表列出了您或您的合作伙伴发送 AS2 消息时可以记录到 CloudWatch 日志中的所有状态代码。不同的消息处理步骤适用于不同的消息类型，并且仅用于监控。“已完成” 和 “失败” 状态表示处理的最后一步，在 JSON 文件中可见。

代码	描述	处理完成了吗？
处理	该消息正在转换为其最终格式。例如，解压缩和解密步骤都具有此状态。	否
MDN_TRANSM	消息处理正在发送 MDN 响应。	否
MDN_RECEIVE	消息处理正在收到 MDN 响应。	否
COMPLETED	消息处理已成功完成。此状态包括为入站消息或出站消息的 MDN 验证发送 MDN 的时间。	是
FAILED	消息处理失败。有关错误代码的列表，请参阅AS2 错误代码。	是

AS2 错误代码

下表列出并描述了您可能从 AS2 文件传输中收到的错误代码。

AS2 错误代码

代码	错误	描述和解决方法
ACCESS_DENIED	• 访问遭拒绝。检查您的访问角色具有必要的权限。 • 文件路径无效 *send-file-path* • 无法使用 ErrorCode以下##代码获取凭证	在处理其中任何一个 SendFilePaths 都无效或格式错误的StartFileTransfer 请求时发生。也就是说，路径缺少 Amazon S3 存储桶名称，或者路径包含无效字符。如果 Transfer Family 未能担任访问角色或日志记录角色，也会发生这种情况。

代码	错误	描述和解决方法
		确保路径包含有效的 Amazon S3 存储桶名称和密钥名称。
AGREEMENT_NOT_FOUND	未找到协议。	要么找不到协议，要么该协议与非活动配置文件相关联。 在 Transfer Family 服务器中更新协议，使其包含活动的配置文件。
CONNECTOR_NOT_FOUND	找不到连接器或相关配置。	要么找不到连接器，要么该连接器与非活动配置文件相关联。 更新连接器以包含活动配置文件。

代码	错误	描述和解决方法
CREDENTIALS_RETRIEVAL_FAILED	1. 在 Secrets Manager 中找不到密钥。 2. 无法访问 Secrets Manager。 3. 无法解密 Secrets Manager 中的密钥。 4. 由于节流，无法获取密钥值。	对于 AS2 Basic 身份验证，密钥的格式必须正确。以下解决方案对应于上一栏中列出的错误。 1. 确保密钥 ID 正确无误。 2. 确保访问角色具有读取密钥的相应权限。访问权限角色必须提供对 StartFileTransfer 请求中所使用文件位置父目录的读取和写入权限。此外，确保角色对拟定发送StartFileTransfer 的父目录提供读取和写入访问权限。 3. 如果使用客户管理的密钥作为密钥，请确保访问角色拥有对 AWS Key Management Service (AWS KMS) 密钥的权限。 4. 有关适用的限额，请参阅 处理密钥的限额。
DECOMPRESSION_FAILED	无法解压缩消息。	要么发送的文件已损坏，要么压缩算法无效。 重新发送消息并验证使用了 ZLIB 压缩，或者在未启用压缩的情况下重新发送消息。
DECRYPT_FAILED	无法解密消息 *message-ID* 。确保合作伙伴拥有正确的公共加密密钥。	解密失败。 确认合作伙伴使用有效证书发送了有效负载，并且使用有效加密算法执行了加密。

代码	错误	描述和解决方法
DECRYPT_FAILED_INVALID_SMIME_FORMAT	无法解析封装的 mimePart。	MIME 有效负载要么已损坏，要么采用不支持的 SMIME 格式。 发送者应确保他们使用的格式受到支持，然后重新发送有效负载。
DECRYPT_FAILED_NO_DECRYPTION_KEY_FOUND	未找到匹配的解密密钥。	没有为合作伙伴配置文件分配与消息匹配的证书，或者与消息匹配的证书现已过期或不再有效。 您必须更新合作伙伴配置文件并确保其中包含有效的证书。
DECRYPT_FAILED_UNSUPPORTED_ENCRYPTION_ALG	使用 ID 为 *encryption-ID* 的不支持算法请求了 SMIME 有效负载解密。	远程发送者发送了一个 AS2 有效负载，其加密算法不受支持。 发件人必须选择AWS Transfer Family支持的加密算法。
DUPLICATE_MESSAGE	重复或双重处理步骤。	有效负载具有重复的处理步骤。例如，有两个加密步骤。 只需一步即可重新发送消息，完成签名、压缩和加密。

代码	错误	描述和解决方法
ENCRYPT_FAILED_NO_ENCRYPTION_KEY_FOUND	在配置文件*local-profile-ID* 中找不到有效的公共加密证书	Transfer Family 正在尝试加密出站消息，但找不到本地配置文件的加密证书。 解决办法选项： • 确保本地配置文件附有用于加密的证书和私钥。 • 确保加密证书当前处于活动状态。
ENCRYPTION_FAILED	无法加密文件*file-name* 。	要发送的文件不可用于加密。 确认文件位于预期的 AS2 位置，并且AWS Transfer Family有权读取该文件。
FILE_SIZE_TOO_LARGE	文件太大。	当发送或接收超过文件大小限制的文件时，就会发生这种情况。
HTTP_ERROR_RESPONSE_FROM_PARTNER	对于 ID=*message-ID* 的消息，*partner-URL* 返回状态 400。	与合作伙伴的 AS2 服务器通信返回了意外的 HTTP 响应代码。 合作伙伴或许可以从其 AS2 服务器日志中提供更多诊断信息。
INSUFFICENT_MESSAGE_SECURITY_UNENCRYPTED	需要加密。	合作伙伴向 Transfer Family 发送了一封未加密的消息，但该消息不受支持。发件人必须使用加密的有效负载。
INVALID_ENDPOINT_PROTOCOL	仅支持 HTTP 和 HTTPS。	您必须在 AS2 连接器配置中指定 HTTP 或 HTTPS 作为协议。

代码	错误	描述和解决方法
INVALID_REQUEST	1. 消息标题有问题。 2. 无法解析密钥 JSON。 密钥 JSON 与预期格式不符。 3. 密钥必须是 JSON 字符串。 4. 用户名不得包含冒号。 用户名不得包含控制字符。 用户名只能包含 ASCII 字符。 密码不得包含控制字符。 密码只能包含 ASCII 字符。	此错误有多种原因。以下解决方案对应于上一栏中列出的错误。 1. 选中as2-from和as2-to字段。确保 MDN 格式的原始消息 ID 准确无误。还要确保消息 ID 格式不缺少任何 AS2 标头。 2. 确保密钥值与记录的格式相匹配，如启用 AS2 连接器的基本身份验证中所述。 3. 确保密钥以字符串形式提供，而不是以二进制形式提供。 4. 对用户名或密码进行必要的更正。
INVALID_URL_FORMAT	网址格式无效：*URL*	当您使用配置了格式错误的 URL 的连接器发送出站消息时，就会发生这种情况。 确保为连接器配置了有效的 HTTP 或 HTTPS URL。
MDN_RESPONSE_INDICATES_AUTHENTICATION_FAILED	不适用	接收方无法对发送者进行身份验证。交易伙伴向 Transfer Family 返回 MDN，其带有处置修饰符 错误：authentication-failed。

代码	错误	描述和解决方法
MDN_RESPONSE_INDICATES_DECOMPRESSION_FAILED	不适用	当接收者无法解压缩消息内容时，就会发生这种情况。交易伙伴向 Transfer Family 返回 MDN，其带有处置修饰符 错误：decompression-failed。
MDN_RESPONSE_INDICATES_DECRYPTION_FAILED	不适用	接收者无法解密消息内容。交易伙伴向 Transfer Family 返回 MDN，其带有处置修饰符 错误：authentication-failed。
MDN_RESPONSE_INDICATES_INSUFFICIENT_MESSAGE_SECURITY	不适用	接收者希望对消息进行签名或加密，但事实并非如此。贸易伙伴向 Transfer Family 返回带有处置修饰符的 MDN 错误：insufficient-message-security。 在连接器上启用签名和/或加密，以符合交易伙伴的期望。
MDN_RESPONSE_INDICATES_INTEGRITY_CHECK_FAILED	不适用	接收者无法验证内容的完整性。贸易伙伴向 Transfer Family 返回带有处置修饰符的 MDN 错误： integrity-check-failed。
PATH_NOT_FOUND	无法创建目录*file-path* 。找不到父路径。	Transfer Family 正在尝试在客户的 Amazon S3 存储桶中创建目录，但未找到该存储桶。 确保StartFileTransfer 命令中提到的每个路径都包含现有存储桶的名称。

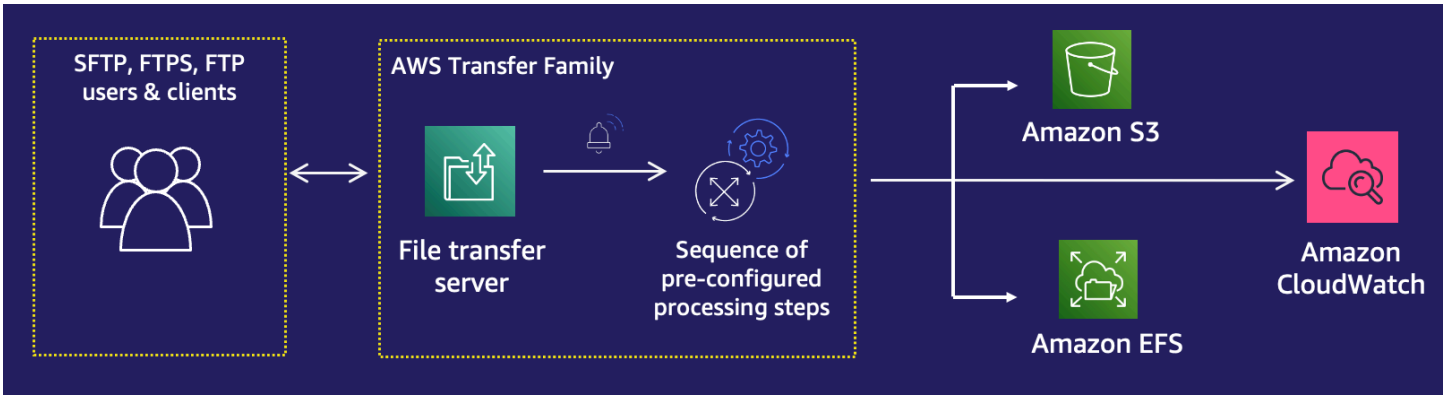
代码	错误	描述和解决方法
SEND_FILE_NOT_FOUND	未找到文件路径 *file-path* 。	Transfer Family 在发送文件操作中找不到该文件。 检查配置的主目录和路径有效，以及 Transfer Family 具有该文件的读取权限。
SERVER_NOT_FOUND	找不到与消息关联的服务器。	Transfer Family 在收到消息时找不到服务器。如果在处理传入消息的过程中删除了服务器，则可能会发生这种情况。
SERVER_NOT_ONLINE	服务器 *server-ID* 不在线。	Transfer Family 服务器处于脱机状态。 启动服务器，使其可以接收和处理消息。
SIGNING_FAILED	对文件签名失败。	要发送的文件不可用于签名，或者无法进行签名。 确认文件位于预期的 AS2 位置，并且AWS Transfer Family有权读取该文件。
SIGNING_FAILED_NO_SIGNING_KEY_FOUND	找不到配置文件*local-profile-ID* 的证书。	正在尝试对出站消息进行签名，但找不到本地配置文件的签名证书。 解决办法选项： • 确保本地配置文件附有证书和用于签名的私钥。 • 确保签名证书当前处于活动状态。

代码	错误	描述和解决方法
UNABLE_RESOLVE_HOST_TO_IP_ADDRESS	无法将主机名解析为 IP 地址。	Transfer Family 无法在 AS2 连接器中配置的公共 DNS 服务器上执行 DNS 到 IP 地址的解析。 更新连接器以指向有效的合作伙伴 URL。
UNABLE_TO_CONNECT_TO_REMOTE_HOST_OR_IP	与端点的连接超时。	Transfer Family 无法与已配置的合作伙伴的 AS2 服务器建立套接字连接。 检查合作伙伴的 AS2 服务器在配置的 IP 地址上可用。
UNABLE_TO_RESOLVE_HOSTNAME	无法解析主机名*hostname*。	Transfer Family 服务器无法使用公共 DNS 服务器解析伙伴的主机名。 检查配置的主机已注册以及 DNS 记录有时间发布。
VERIFICATION_FAILED	AS2 消息*message-ID* 的签名验证失败或 MIC 代码不匹配。	检查发件人的签名证书与远程配置文件的签名证书相匹配。还要检查 MIC 算法兼容 AWS Transfer Family。

代码	错误	描述和解决方法
VERIFICATION_FAILED_NO_MATCHING_KEY_FOUND	• 在配置文件 *partner-profile-ID* 中找不到与消息签名匹配的公共证书。 • 无法为不存在的配置文件 *partner-profile-ID* 获取证书。 • 在配置文件 *partner-profile-ID* 中找不到有效的证书。	AWS Transfer Family正在尝试验证收到的消息的签名，但找不到与合作伙伴配置文件匹配的签名证书。 解决办法选项： • 确保合作伙伴配置文件附有签名证书。 • 确保证书当前处于活动状态。 • 确保证书是合作伙伴的正确签名证书。

AWS Transfer Family 托管工作流程

AWS Transfer Family 支持文件处理的托管工作流程。借助托管工作流程，您可以在通过 SFTP、FTPS 或 FTP 传输文件后启动工作流程。使用此功能，您可以协调文件处理所需的所有必要步骤，从而安全且经济高效地满足 business-to-business (B2B) 文件交换的合规性要求。此外，您还可以从 end-to-end 审计和可见性中受益。



通过协调文件处理任务，托管工作流可帮助您在下游应用程序使用数据之前对其进行预处理。此类文件处理任务可能包括：

- 将文件移动到用户特定的文件夹。
- 作为工作流的一部分对文件进行解密。
- 标记文件
- 通过创建 AWS Lambda 函数并将其附加到工作流程来执行自定义处理。
- 文件成功传输后发送通知。（有关详细介绍此用例的博客文章，请参阅使用 AWS Transfer Family 托管工作流程自定义文件传送通知。）

要快速复制和标准化组织中多个业务部门的常见上传后文件处理任务，您可以使用基础设施即代码 (IaC) 来部署工作流程。您可以指定要在完整上传的文件上启动托管工作流程。对于因会话过早断开连接而仅部分上传的文件，您也可以指定不同的托管工作流程。内置的异常处理功能可帮助您对文件处理结果做出快速反应，同时让您能够控制如何处理故障。此外，每个工作流程步骤都会生成详细的日志，您可以对其进行审核以跟踪数据沿袭。

要开始使用 ，请执行以下步骤：

1. 根据您的要求将工作流程设置为包含预处理操作，例如复制、标记和其他步骤。有关详细信息，请参阅 创建工作流。

2. 配置执行角色，Transfer Family 会使用该角色来运行工作流程。有关详细信息，请参阅 适用于工作流程的 IAM 策略。
3. 将工作流程映射到服务器，以便在文件到达时，实时评估和启动此工作流程中指定的操作。有关详细信息，请参阅 配置和运行工作流程。

相关信息

- 要监控您的工作流程执行情况，请参阅 使用 T CloudWatch ransfer Family 的指标。
- 有关详细的执行日志和故障排除信息，请参阅 使用 Amazon 解决与工作流程相关的错误 CloudWatch。
- 我们有一个可供您参加的研讨会，您可以在其中构建文件传输解决方案。该解决方案利用 AWS Transfer Family 托管 SFTP/FTPS 终端节点，利用 Amazon Cognito 和 DynamoDB 进行用户管理。您可以在此处查看本次研讨会的详细信息。
- 请查看AWS Transfer Family 托管工作流程，了解对 Transfer Family 工作流程的简要介绍。

主题

- 创建工作流
- 使用预定义的步骤
- 使用自定义文件处理步骤
- 适用于工作流程的 IAM 策略
- 工作流程的异常处理
- 监控工作流程执行情况
- 通过模板创建工作流
- 从 Transfer Family 服务器中移除工作流
- 托管工作流限制和局限性

有关托管工作流程入门的更多帮助，请参阅以下资源：

- AWS Transfer Family 托管工作流程演示视频
- 使用 AWS Transfer Family 工作流程构建云原生文件传输平台博客文章

创建工作流

您可以使用创建托管工作流程 AWS Management Console，如本主题所述。为了使工作流程创建过程尽可能简单，控制台中的大多数部分都提供了上下文帮助面板。

工作流程有两种步骤：

- 标称步骤 — 标称步骤是要应用于传入文件的文件处理步骤。如果您选择多个标称步骤，每个步骤将按线性序列处理。
- 异常处理步骤 — 异常处理程序是文件处理步骤，可在任何标称步骤失败或导致验证错误时 AWS Transfer Family 执行。

创建工作流

1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在左侧导航窗格中，选择 工作流。
3. 在 工作流页面，选择 创建工作流。
4. 在创建工作流页面，输入描述。此描述显示在“工作流程”页面上。
5. 在标称步骤部分中，选择添加步骤。添加一个或多个步骤。
    a. 从可用选项中选择步骤类型。有关各种作业状态的更多信息，请参阅 the section called “使用预定义的步骤”。
    b. 选择“下一步”，然后为该步骤配置参数。
    c. 选择“下一步”，然后查看该步骤的详细信息。
    d. 选择“创建步骤”以添加该步骤并继续。
    e. 根据需要继续添加步骤。工作流中的最大步骤数为 8。
    f. 添加完所有必需的标称步骤后，向下滚动到“异常处理程序 - 可选”部分，然后选择“添加步骤”。

    > **Note**
    >
    > 为便于您实时了解故障，我们建议您设置异常处理程序和步骤，以便在工作流程失败时执行。

6. 要配置异常处理程序，请按照与前面所述相同的方式添加步骤。如果某个文件导致任何步骤引发异常，则会逐一调用您的异常处理程序。

7. （可选）向下滚动到“标签”部分，然后为您的工作流程添加标签。
8. 检查配置并选择 创建工作流。

> **Important**
>
> 创建工作流程后，您将无法对其进行编辑，因此请务必仔细查看配置。

配置和运行工作流程

在运行工作流程之前，您需要将其与 Transfer Family 服务器相关联。

将 Transfer Family 配置为对上传的文件运行工作流程

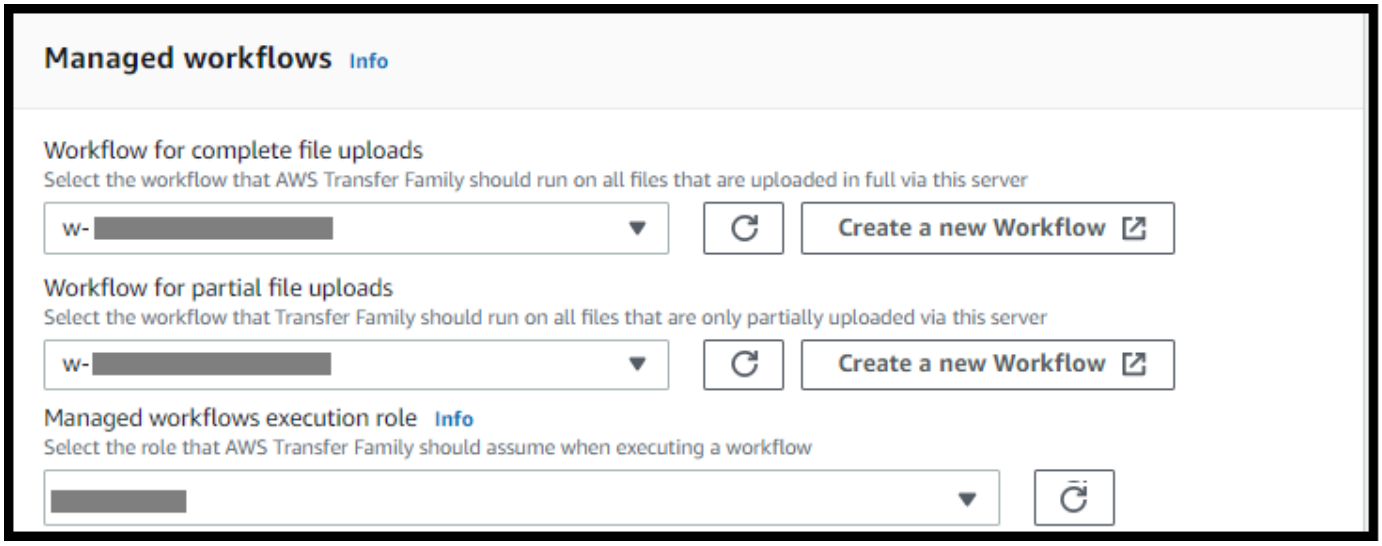
1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在左侧导航窗格中，选择服务器。
   - 要将工作流程添加到现有服务器，请选择要用于工作流程的服务器。
   - 或者，创建一个新服务器并向其添加工作流程。有关更多信息，请参阅 配置 SFTP、FTPS 或 FTP 服务器端点。
3. 在服务器的详细信息页面上，向下滚动到其他详细信息部分，然后选择编辑。

> **Note**
>
> 默认情况下，服务器没有任何关联的工作流程。您可以使用“其他详细信息”部分将工作流与所选服务器相关联。

4. 在编辑其他详细信息页面的托管工作流程部分，选择要在所有上传中运行的工作流程。

> **Note**
>
> 如果您还没有工作流，请选择“创建新的工作流程”来创建工作流程。

   a. 选择要使用的工作流程 ID。
   b. 选择执行角色。这是 Transfer Family 在执行工作流程步骤时所扮演的角色。有关更多信息，请参阅 适用于工作流程的 IAM 策略。选择 Save（保存）。



> **Note**
>
> 如果您不想再将工作流程与服务器关联，则可以移除关联。有关更多信息，请参阅 从 Transfer Family 服务器中移除工作流。

要执行工作流程

要执行工作流程，您需要将文件上传到您配置了关联工作流程的 Transfer Family 服务器。

> **Note**
>
> 无论何时从服务器上移除工作流程并用新的工作流程替换它，或者更新服务器配置（这会影响工作流程的执行角色），都必须等待大约 10 分钟才能执行新的工作流程。Transfer Family 服务器会缓存工作流程细节，服务器需要 10 分钟才能刷新其缓存。
> 此外，您必须注销所有活动的 SFTP 会话，然后等待 10 分钟重新登录才能看到更改。

Example

```
# Execute a workflow
> sftp bob@s-1234567890abcdef0.server.transfer.us-east-1.amazonaws.com

Connected to s-1234567890abcdef0.server.transfer.us-east-1.amazonaws.com.
```

```
sftp> put doc1.pdf
Uploading doc1.pdf to /DOC-EXAMPLE-BUCKET/home/users/bob/doc1.pdf
doc1.pdf                                                                    100% 5013KB
 601.0KB/s   00:08
sftp> exit
>
```

文件上传后，会对您的文件执行定义的操作。例如，如果您的工作流程包含复制步骤，则该文件将会被复制到您在该步骤中定义的位置。您可以使用 Amaz CloudWatch on Logs 来跟踪已执行的步骤及其执行状态。

查看工作流详细信息

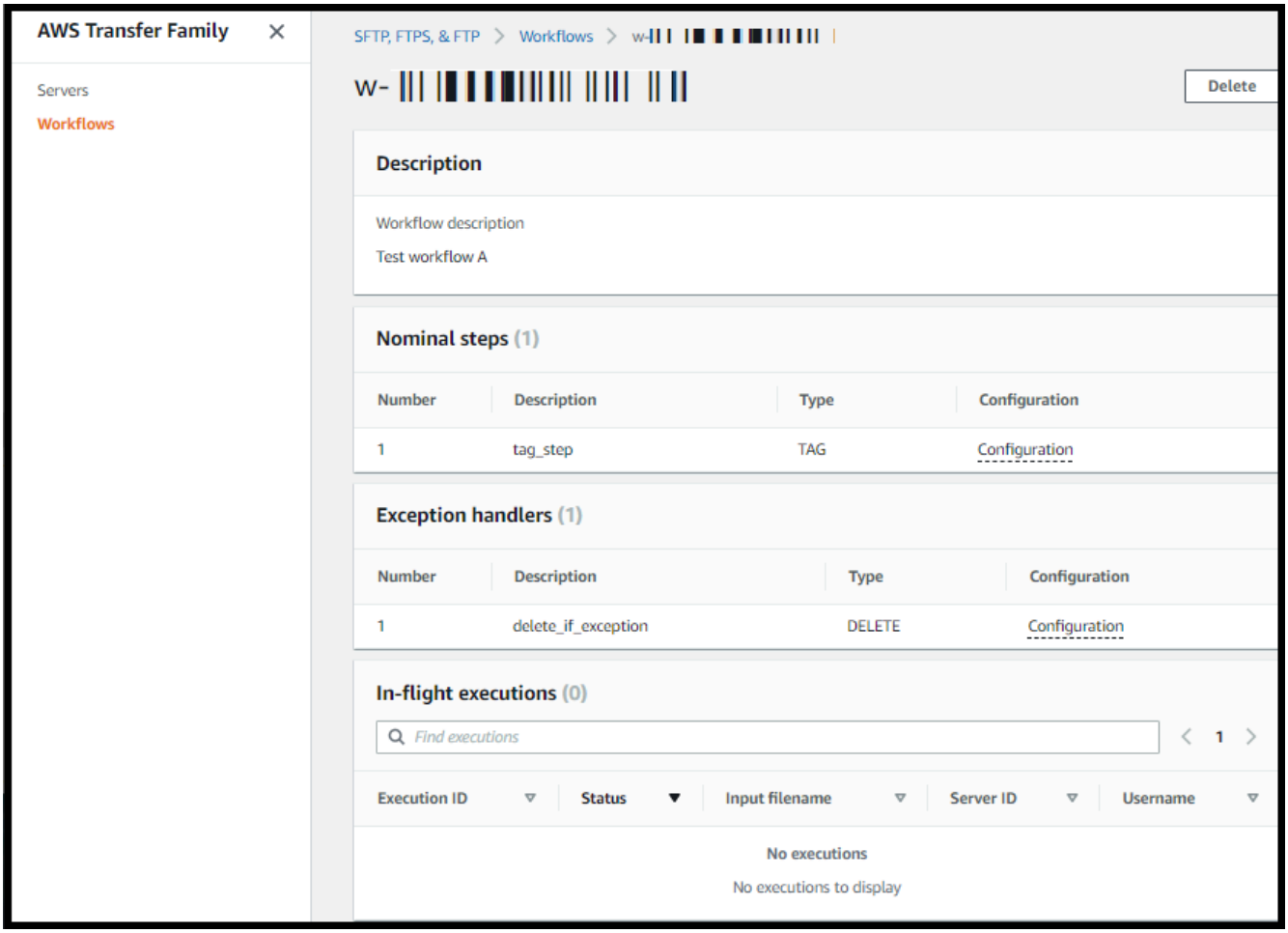
您可以查看有关先前创建的工作流程或工作流程执行的详细信息。要查看这些详细信息，您可以使用控制台或 AWS Command Line Interface (AWS CLI)。

Console

查看工作流详细信息

1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在左侧导航窗格中，选择 工作流。
3. 在“工作流程”页面上，选择一个工作流程。

   工作流程详细信息页面随即打开。



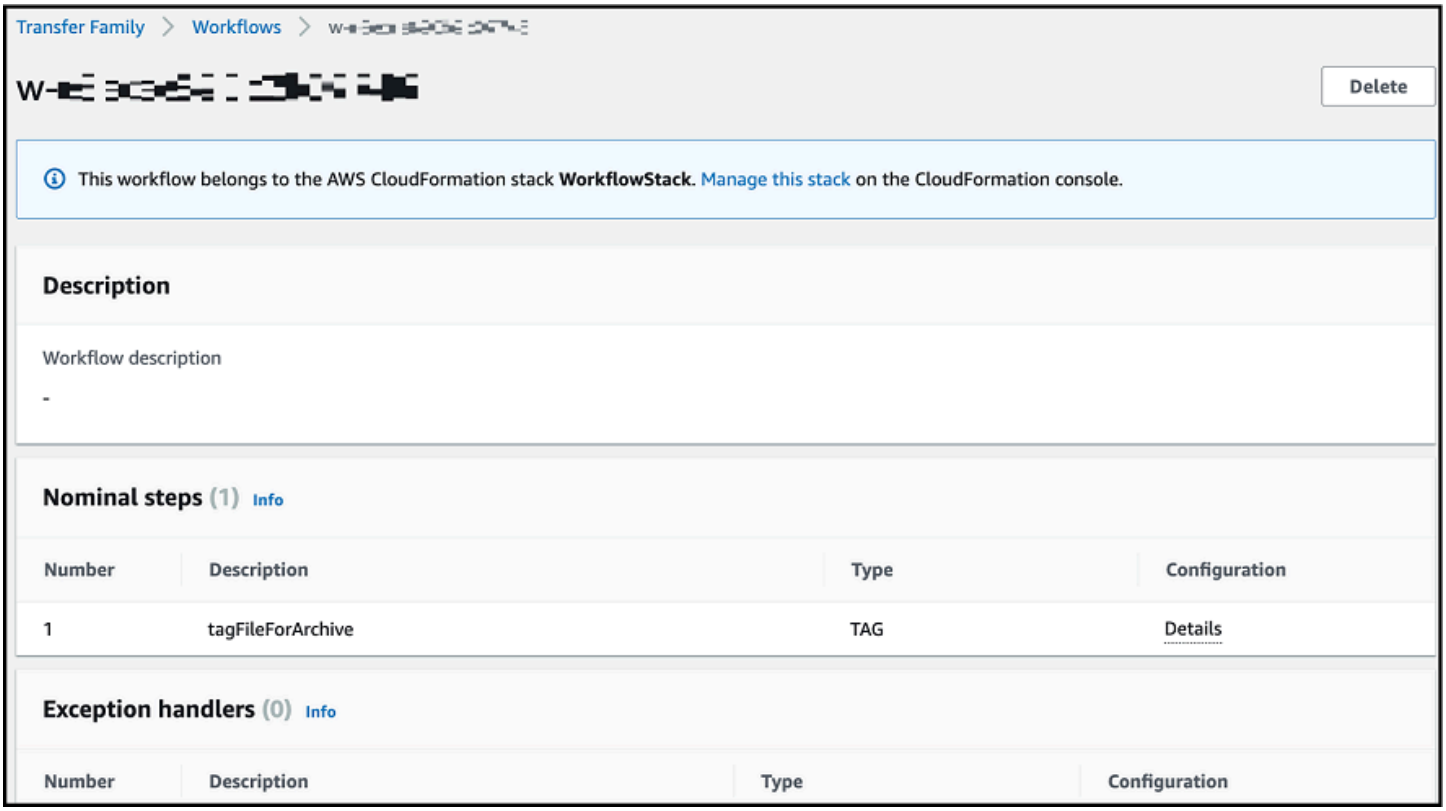
CLI

要查看工作流详细信息，请使用 `describe-workflow` CLI 命令，如以下示例所示。将工作流程 ID *w-1234567890abcdef0* 替换为您自己的值。有关更多信息，请参阅 AWS CLI 命令引用中的 describe-workflow。

```
# View Workflow details
> aws transfer describe-workflow --workflow-id w-1234567890abcdef0
{
    "Workflow": {
        "Arn": "arn:aws:transfer:us-east-1:111122223333:workflow/
w-1234567890abcdef0",
        "WorkflowId": "w-1234567890abcdef0",
        "Name": "Copy file to shared_files",
        "Steps": [
            {
                "Type": "COPY",
```

```
                "CopyStepDetails": {
                "Name": "Copy to shared",
                "FileLocation": {
                    "S3FileLocation": {
                        "Bucket": "DOC-EXAMPLE-BUCKET",
                        "Key": "home/shared_files/"
                    }
                }
                }
            }
        ],
        "OnException": {}
    }
}
```

如果您的工作流程是作为 AWS CloudFormation 堆栈的一部分创建的，则可以使用 AWS CloudFormation 控制台 (https://console.aws.amazon.com/cloudformation) 管理工作流程。



使用预定义的步骤

创建工作流程时，可以选择添加本主题中讨论的以下预定义步骤之一。您还可选择添加自己的自定义文件处理步骤。有关更多信息，请参阅 the section called “使用自定义文件处理步骤”。

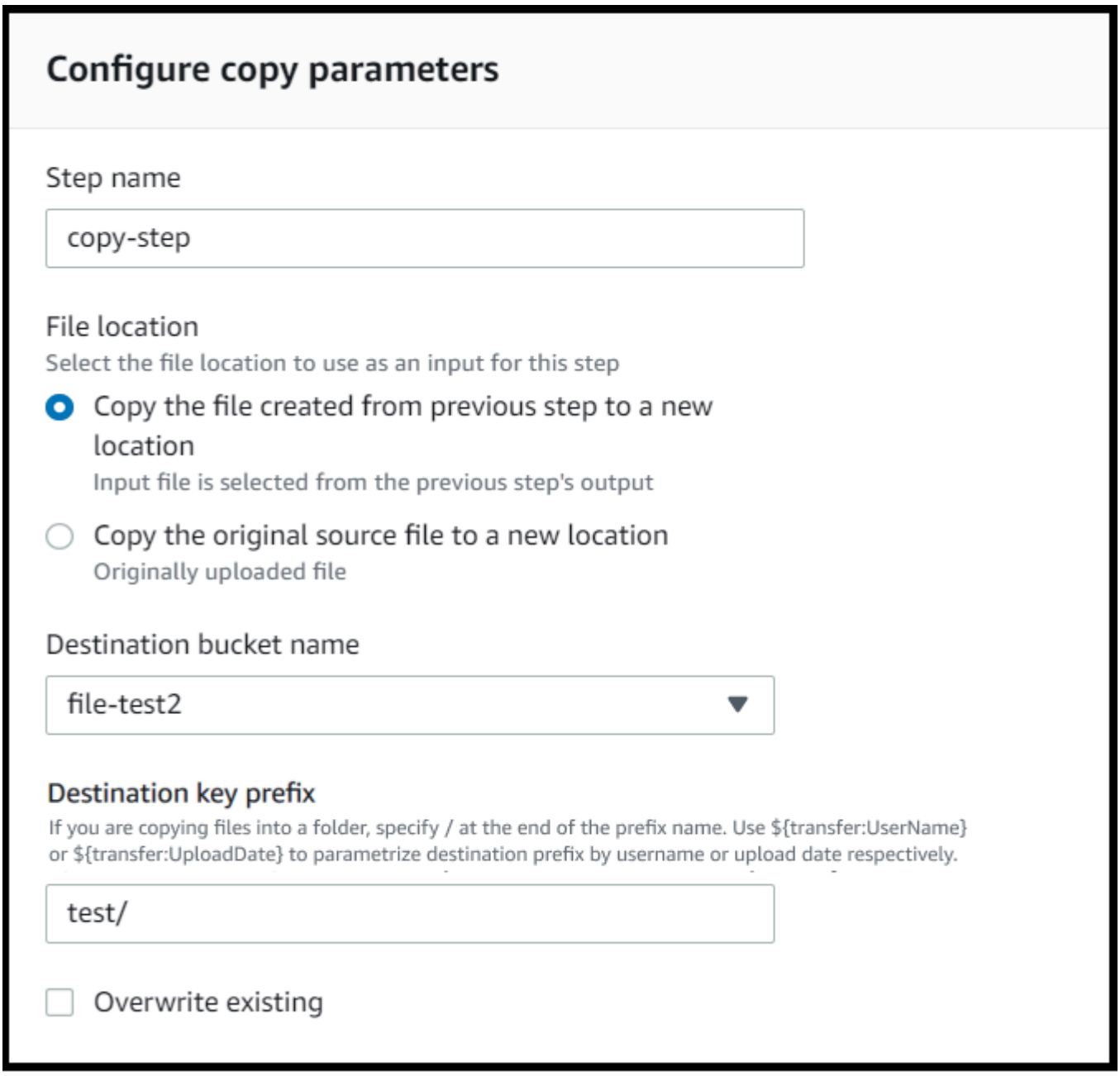
主题

复制文件

复制文件步骤会在新的 Amazon S3 位置创建已上传文件的副本。目前，您只能在 Amazon S3 上使用复制文件步骤。

以下复制文件步骤将文件复制到 `file-test` 目标存储桶中的 `test` 文件夹。

如果复制文件步骤不是工作流程的第一步，则可以指定文件位置。通过指定文件位置，您可以复制上一步中使用的文件或上传的原始文件。您可以使用此功能制作原始文件的多个副本，同时保持源文件完好无损，便于文件存档和记录保留。有关示例，请参阅标记和删除工作流程示例。



提供存储桶和密钥的详细信息

您必须提供存储桶名称和复制文件步骤的目标密钥。密钥可以是路径名或文件名。将密钥视为路径名还是文件名取决于密钥是否以正斜杠 (/) 字符结尾。

如果最后一个字符是 /，则您的文件将被复制到此文件夹，并且其名称不会更改。如果最后一个字符是字母数字，则您上传的文件将被重命名为键值。在这种情况下，如果已存在具有该名称的文件，则相关行为将取决于“覆盖现有文件”字段的设置。

- 如果选择“覆盖现有文件”，则现有文件会被正在处理的文件替换。

- 如果未选择“覆盖现有文件”，则不会发生任何事情，并且工作流将会停止处理。

> **Tip**
>
> 如果在同一文件路径上执行并发写入，则在覆盖文件时可能会导致意外行为。

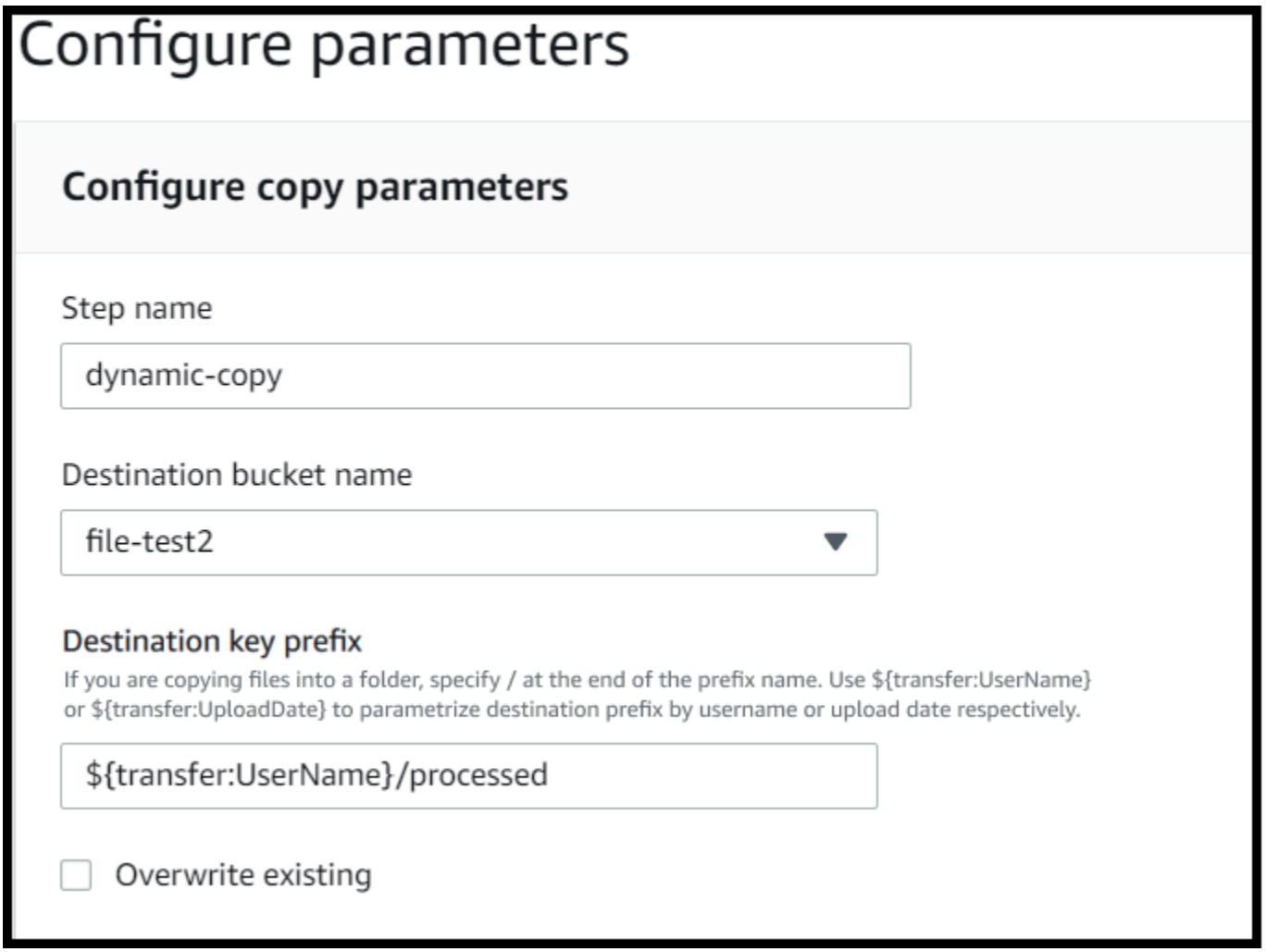
例如，如果您的键值是 `test/`，则您上传的文件将被复制到 `test` 文件夹。如果您的密钥值为 `test/today`，（并且选择了覆盖现有文件），则您上传的每个文件都将复制到该 `test` 文件夹中名为 `today` 的文件中，并且每个后续文件都会覆盖前一个文件。

> **Note**
>
> Amazon S3 支持存储桶和对象且没有层次结构。但是，您可以在对象键名称中使用前缀和分隔符来暗示层次结构，并以类似于文件夹的方式组织数据。

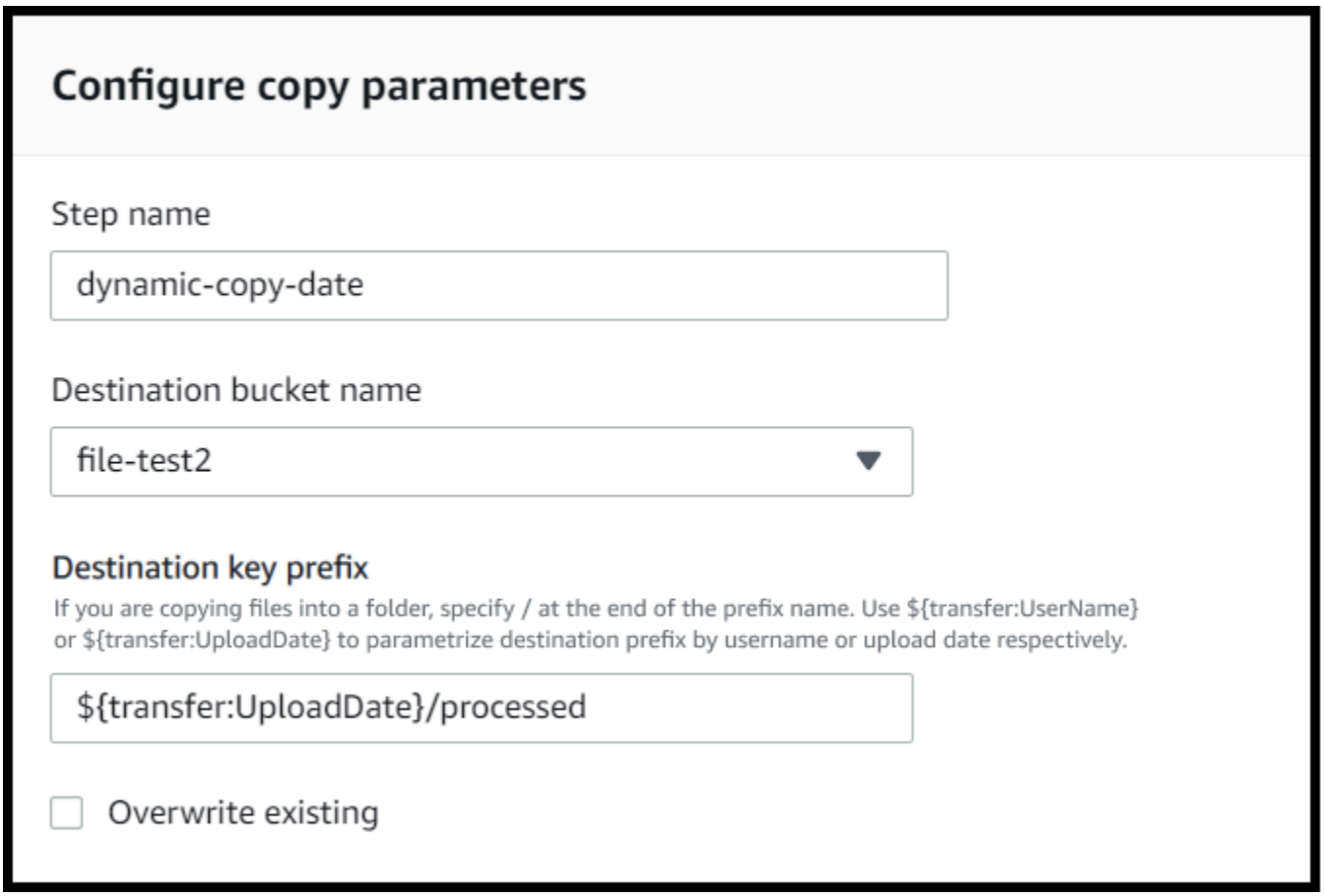
在复制文件步骤中使用命名变量

在复制文件步骤中，您可以使用变量将文件动态复制到用户特定的文件夹中。目前，您可以使用 `${transfer:UserName}` 或 `${transfer:UploadDate}` 作为变量，将文件复制到正在上传文件的给定用户的目标位置，或者根据当前日期将文件复制到目标位置。

在以下示例中，如果用户 `richard-roe` 上传文件，则该文件将被复制到 `file-test2/richard-roe/processed/` 文件夹。如果用户 `mary-major` 上传文件，则该文件将被复制到 `file-test2/mary-major/processed/` 文件夹。



同样，您可以使用 `${transfer:UploadDate}` 作为变量，将文件复制到以当前日期命名的目标位置。在以下示例中，如果您将目标设置为 2022 年 2 月 1 日的 `${transfer:UploadDate}/processed`，则上传的文件将复制到 `file-test2/2022-02-01/processed/` 文件夹。



您也可以同时使用这两个变量，将它们的功能结合起来。例如：

- 例如，您可以将目标键前缀设置为 **`folder/${transfer:UserName}/${transfer:UploadDate}/`**，这样可以创建嵌套文件夹，例如 `folder/marymajor/2023-01-05/`。
- 例如，您可以将目标键前缀设置为 **`folder/${transfer:UserName}-${transfer:UploadDate}/`**，以连接两个变量，例如 `folder/marymajor-2023-01-05/`。

复制步骤的 IAM 权限

要允许复制步骤成功，请确保您的工作流程的执行角色包含以下权限。

```
{
            "Sid": "ListBucket",
            "Effect": "Allow",
            "Action": "s3:ListBucket",
            "Resource": [
                "arn:aws:s3:::destination-bucket-name"
```

```
            ]
        },
        {
            "Sid": "HomeDirObjectAccess",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:DeleteObjectVersion",
                "s3:DeleteObject",
                "s3:GetObjectVersion"
            ],
            "Resource": "arn:aws:s3:::destination-bucket-name/*"
        }
```

> **Note**
>
> 仅当您未选择“覆盖现有文件”时，才需要 `s3:ListBucket` 权限。此权限会检查您的存储桶，以查看是否已存在同名文件。如果您选择了“覆盖现有文件”，则工作流程无需检查文件，只需将其写入即可。
>
> 如果您的 Amazon S3 文件有标签，则需要在 IAM 策略中添加一两个权限。
>
> - 为未进行版本控制的 Amazon S3 文件添加 `s3:GetObjectTagging`。
> - 为进行版本控制的 Amazon S3 文件添加 `s3:GetObjectVersionTagging`。

解密文件

AWS 存储博客上有一篇文章描述了如何加密和解密文件，使用 PGP 加密和解密文件以及。 AWS Transfer Family

在工作流程中使用 PGP 解密

Transfer Family 内置了对 Prety Good Privacy (PGP) 解密的支持。您可以对通过 SFTP、FTPS 或 FTP 上传到 Amazon Simple Storage Service (Amazon S3) 或 Amazon Elastic File System (Amazon EFS) 的文件使用 PGP 解密。

要使用 PGP 解密，必须创建并存储用于解密文件的 PGP 私钥。然后，您的用户可以使用相应的 PGP 加密密钥对文件进行加密，然后再将文件上传到您的 Transfer Family 服务器。收到加密文件后，可以在工作流程中解密这些文件。有关详细教程，请参阅设置用于解密文件的托管工作流程。

若要在工作流程中使用 PGP 解密

1. 确定 Transfer Family 服务器来托管您的工作流，或创建新工作流。您需要先获得服务器 ID，然后才能 AWS Secrets Manager 使用正确的密钥名称存储 PGP 密钥。
2. 将您的 PGP 密钥存储在所需的密钥名称 AWS Secrets Manager 下。有关更多信息，请参阅 管理密钥对。工作流程可以根据 Secrets Manager 中的密钥名称自动找到用于解密的正确 PGP 密钥。

   > **Note**
   >
   > 当你在 Secret AWS 账户 s Manager 中存储密钥时，会产生费用。有关定价的信息，请参阅 AWS Secrets Manager 定价。

3. 使用您的 PGP 密钥对加密文件。（有关受支持的事件的列表，请参阅支持的 PGP 客户端。）如果您使用命令行，请使用以下命令。要使用此命令，请将 *username@example.com* 替换为用于创建 PGP 密钥对的电子邮件地址。将 *testfile.txt* 替换为您要加密的文件名称。

   ```
   gpg -e -r username@example.com testfile.txt
   ```

4. 将加密文件上传至您的 Transfer Family 服务器。
5. 在工作流程中配置解密步骤。有关更多信息，请参阅 添加解密步骤。

添加解密步骤

解密步骤对作为工作流程一部分上传到 Amazon S3 或 Amazon EFS 的加密文件进行解密。有关配置解密的详细信息，请参阅 在工作流程中使用 PGP 解密。

在为工作流程创建解密步骤时，必须指定解密文件的目的地。如果目标位置已存在文件，则还必须选择是否覆盖现有文件。您可以使用 Ama CloudWatch zon Logs 监控解密工作流程结果并实时获取每个文件的审核日志。

为步骤选择解密文件类型后，将出现“配置参数”页面。填写“配置 PGP 解密参数”部分的值。

可用选项如下：

- 步骤名称 - 输入步骤的描述性名称。
- 文件位置 - 通过指定文件位置，您可以解密上一步中使用的文件或上传的原始文件。

> **Note**
>
> 如果此步骤是工作流的第一步，则此参数不可用。

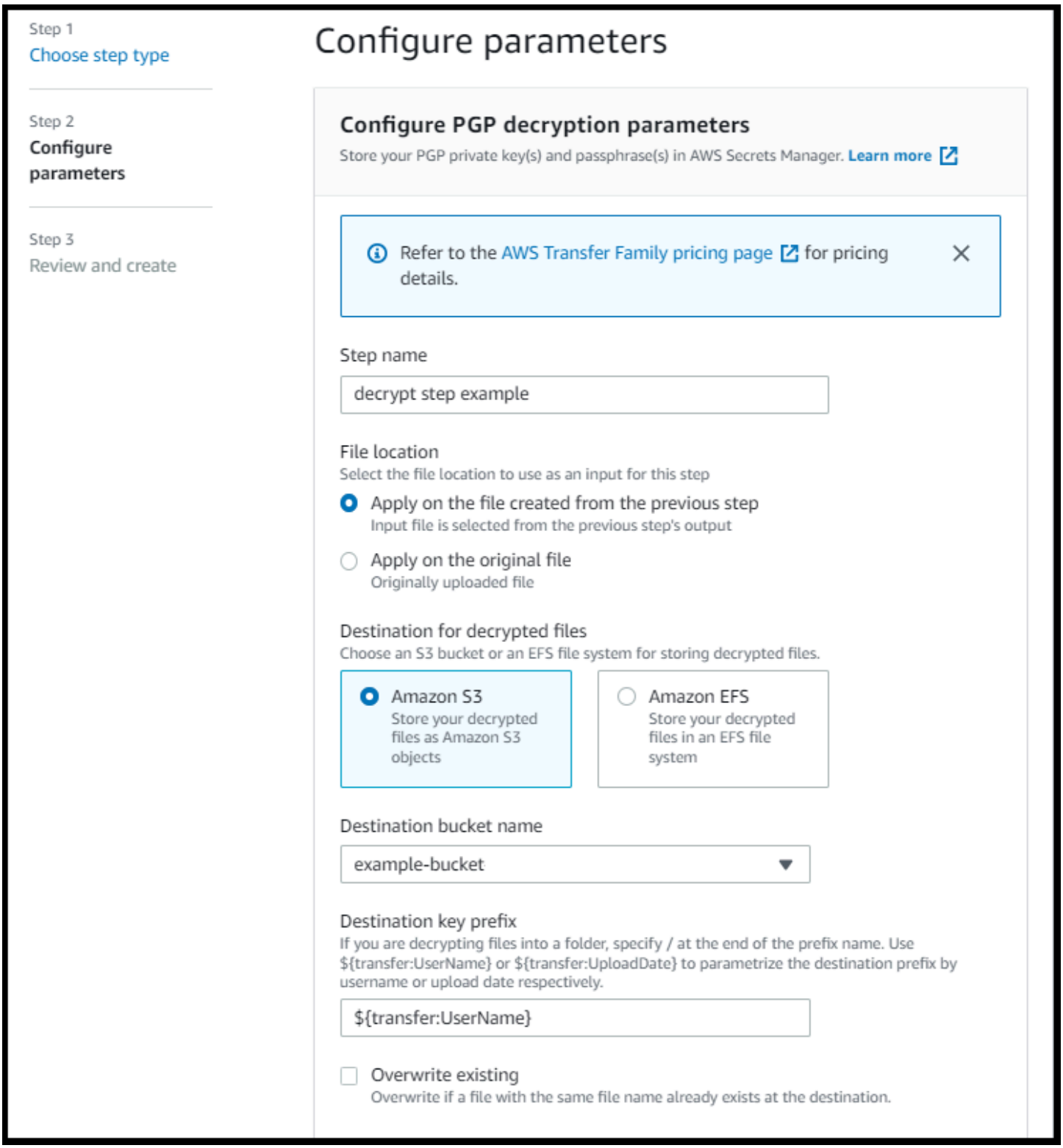
- 解密文件的目标 - 选择 Amazon S3 存储桶或 Amazon EFS 文件系统作为解密文件的目的地。
  - 如果您选择 Amazon S3，则必须提供目标存储桶名称和目标密钥前缀。要按用户名参数化目标密钥前缀，请为“**`${transfer:UserName}`**目标密钥前缀”输入 。同样，要按上传日期参数化目标密钥前缀，请为“**`${Transfer:UploadDate}`**目标密钥前缀”输入 。
  - 如果您选择 Amazon EFS，则必须提供目标文件系统和路径。

> **Note**
>
> 您在此处选择的存储选项必须与与此工作流程关联的 Transfer Family 服务器使用的存储系统相匹配。否则，当您尝试运行此工作流程时会收到错误。

- 覆盖现有文件 - 如果您上传了一个文件，并且目标位置上已经存在具有相同文件名的文件，则相关行为取决于此参数的设置：
  - 如果选择“覆盖现有文件”，则现有文件会被正在处理的文件替换。
  - 如果未选择“覆盖现有文件”，则不会发生任何事情，并且工作流将会停止处理。

> **Tip**
>
> 如果在同一文件路径上执行并发写入，则在覆盖文件时可能会导致意外行为。

以下屏幕截图显示了您可以为解密文件步骤选择的选项示例。



解密步骤的 IAM 权限

若要使解密步骤成功，请确保您的工作流程的执行角色包含以下权限。

```
{
            "Sid": "ListBucket",
            "Effect": "Allow",
            "Action": "s3:ListBucket",
            "Resource": [
                "arn:aws:s3:::destination-bucket-name"
            ]
        },
        {
            "Sid": "HomeDirObjectAccess",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:DeleteObjectVersion",
                "s3:DeleteObject",
                "s3:GetObjectVersion"
            ],
            "Resource": "arn:aws:s3:::destination-bucket-name/*"
        },
        {
            "Sid": "Decrypt",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue",
            ],
            "Resource": "arn:aws:secretsmanager:region:account-id:secret:aws/transfer/
*"
        }
```

Note

仅当您未选择“覆盖现有文件”时，才需要 `s3:ListBucket` 权限。此权限会检查您的存储桶，以查看是否已存在同名文件。如果您选择了“覆盖现有文件”，则工作流程无需检查文件，只需将其写入即可。

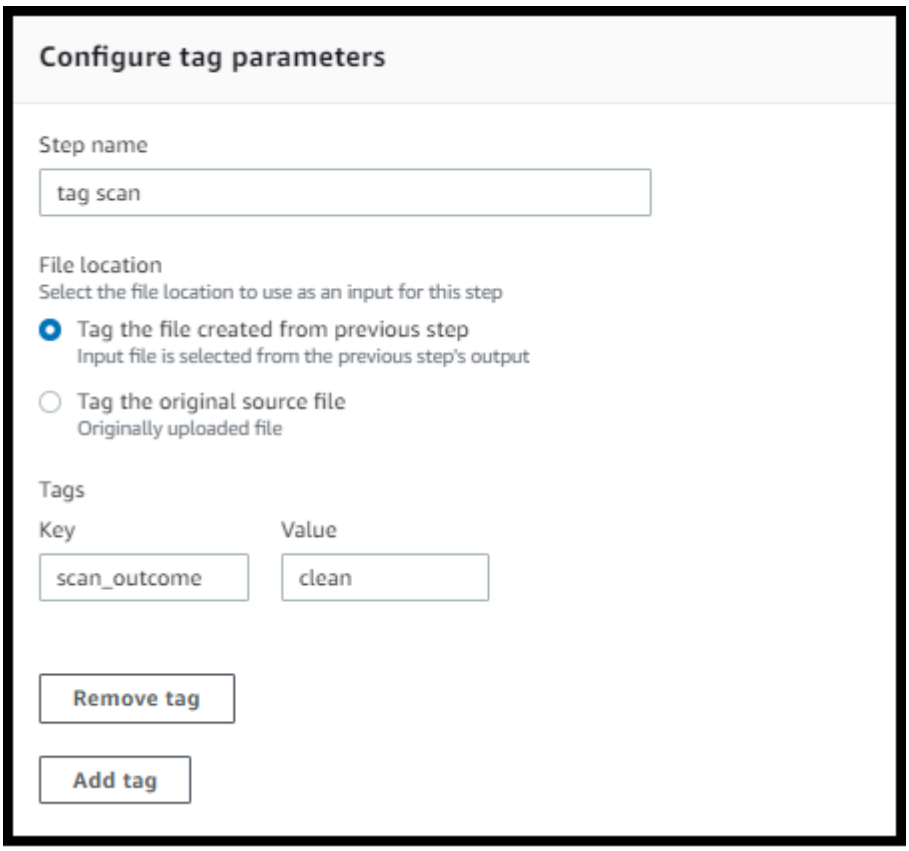
如果您的 Amazon S3 文件有标签，则需要在 IAM 策略中添加一两个权限。

- 为未进行版本控制的 Amazon S3 文件添加 `s3:GetObjectTagging`。
- 为进行版本控制的 Amazon S3 文件添加 `s3:GetObjectVersionTagging`。

标记文件

要标记传入文件以进行进一步的下游处理，请使用标记步骤。输入要分配给传入文件的标签值。当前，只有当您使用 Amazon S3 作为 Transfer Family 服务器存储时，才支持标签操作。

以下示例标签步骤将 `scan_outcome` 和 `clean` 分别指定为标签键和值。



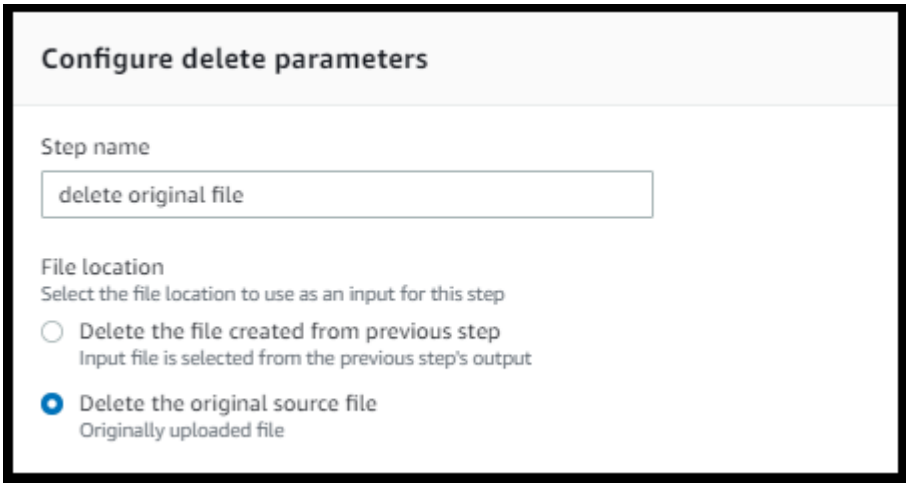
若要使标记步骤成功，请确保您的工作流程的执行角色包含以下权限。

```
{
            "Sid": "Tag",
            "Effect": "Allow",
            "Action": [
                "s3:PutObjectTagging",
                "s3:PutObjectVersionTagging"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ]
}
```

> **Note**
>
> 如果您的工作流程包含在复制或解密步骤之前运行的标签步骤，则需要向 IAM 策略添加一两个权限。
>
> - 为未进行版本控制的 Amazon S3 文件添加 `s3:GetObjectTagging`。
> - 为进行版本控制的 Amazon S3 文件添加 `s3:GetObjectVersionTagging`。

delete-file

要从上一个工作流程步骤中删除已处理的文件或删除最初上传的文件，请使用删除文件步骤。



若要使删除步骤成功，请确保您的工作流程的执行角色包含以下权限。

```
{
            "Sid": "Delete",
            "Effect": "Allow",
            "Action": [
                "s3:DeleteObjectVersion",
                "s3:DeleteObject"
            ],
            "Resource": "arn:aws:secretsmanager:region:account-ID:secret:aws/transfer/
*"
        }
```

工作流程的命名变量

对于复制和解密步骤，您可以使用变量来动态执行操作。目前， AWS Transfer Family 支持以下命名变量。

- 使用 `${transfer:UserName}` 根据上传文件的用户将文件复制或解密到目标位置。
- 使用 `${transfer:UploadDate}` 根据当前日期将文件复制或解密到目标位置。

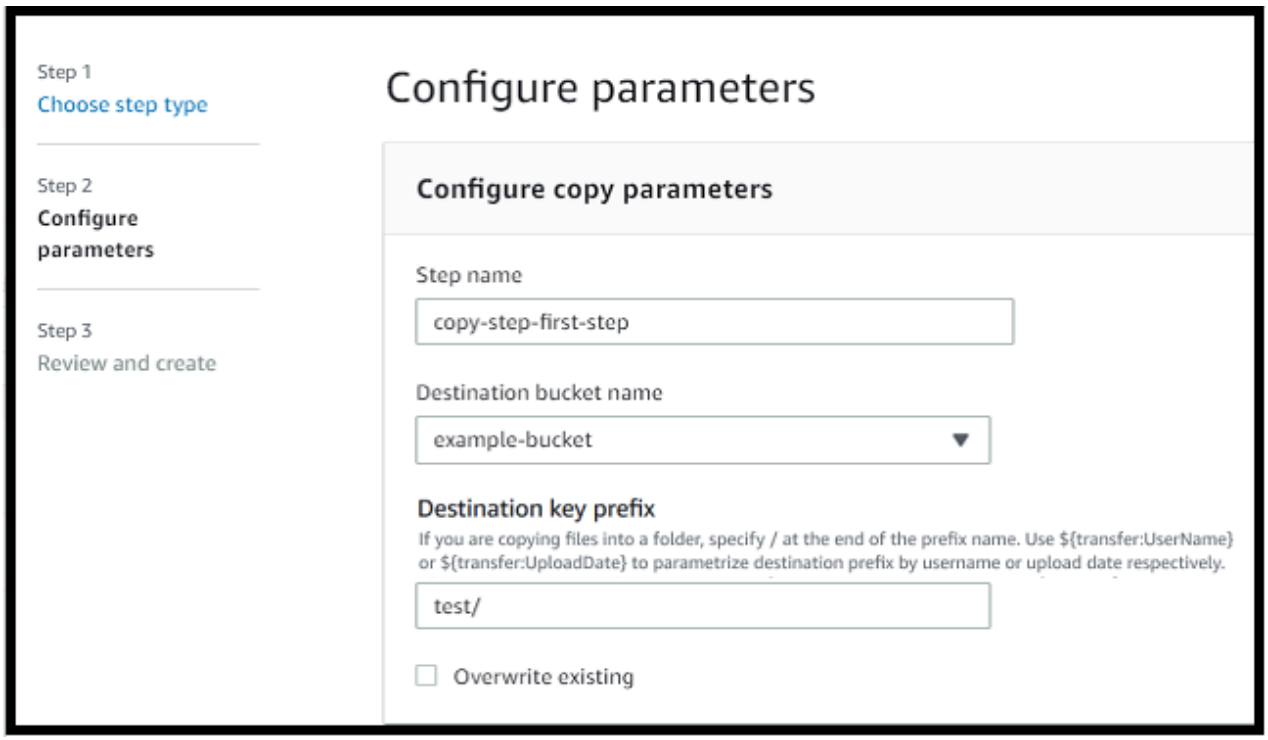
标记和删除工作流程示例

以下示例说明了一个工作流程，该工作流程用于标记需要由下游应用程序（例如数据分析平台）处理的传入文件。标记传入文件后，工作流程会删除最初上传的文件以节省存储成本。

Console

标记和移动工作流程示例

1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在左侧导航窗格中，选择 工作流。
3. 在 工作流页面，选择 创建工作流。
4. 在创建工作流页面，输入描述。此描述显示在“工作流程”页面上。
5. 添加第一步（复制）。
    a. 在标称步骤部分中，选择添加步骤。
    b. 选择复制文件，然后选择 下一步。
    c. 输入步骤名称，然后选择目标存储桶和密钥前缀。



d. 选择“下一步”，然后查看该步骤的详细信息。

e. 选择“创建步骤”以添加该步骤并继续。

6. 添加第二步（标记）。

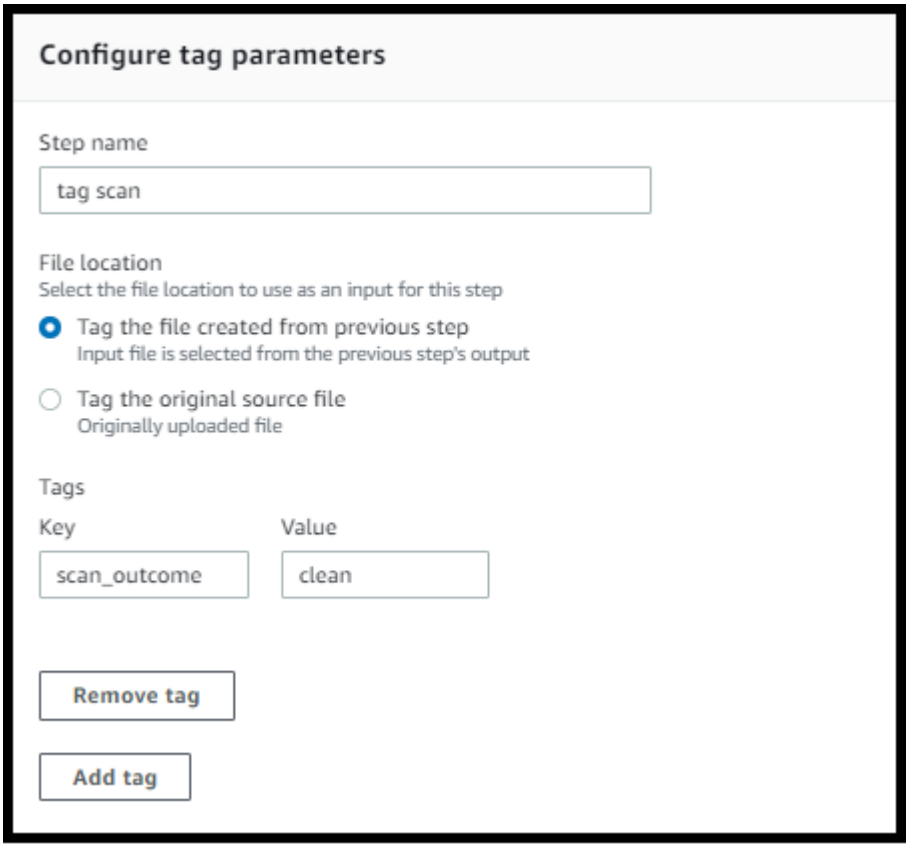
a. 在标称步骤部分中，选择添加步骤。

b. 选择您的标签文件，然后选择 下一步。

c. 输入步骤名称。

d. 在“文件位置”中，选择“标记上一步创建的文件”。

e. 输入键和值。



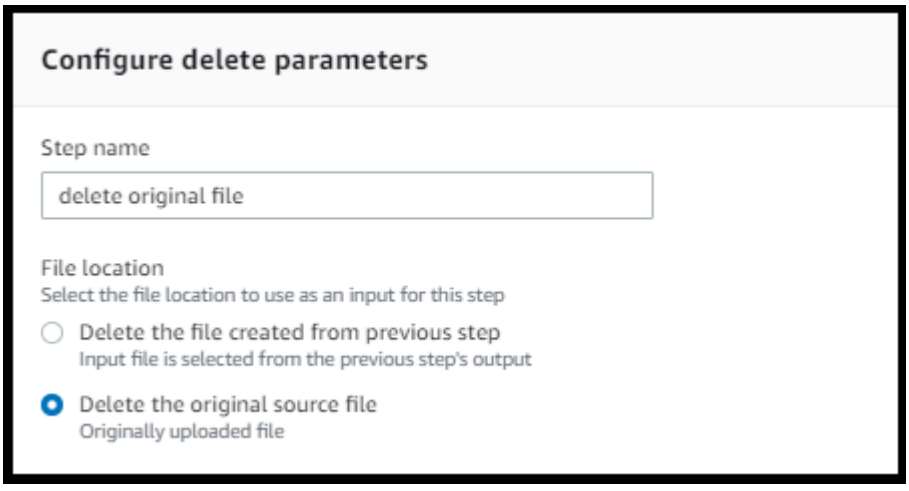
f. 选择“下一步”，然后查看该步骤的详细信息。

g. 选择“创建步骤”以添加该步骤并继续。

7. 添加第三步（删除）。

a. 在标称步骤部分中，选择添加步骤。

b. 选择 删除文件，然后选择 下一步。



c. 输入步骤名称。

d. 在“文件位置”中，选择“删除原始源文件”。

e. 选择“下一步”，然后查看该步骤的详细信息。

f. 选择“创建步骤”以添加该步骤并继续。

8. 查看工作流程配置，然后选择创建工作流程。

CLI

标记和移动工作流程示例

1. 将以下代码保存到文件中；例如，tagAndMoveWorkflow.json。将每个 *user input placeholder* 替换为您自己的信息。

```
[
   {
       "Type": "COPY",
       "CopyStepDetails": {
          "Name": "CopyStep",
          "DestinationFileLocation": {
             "S3FileLocation": {
                "Bucket": "DOC-EXAMPLE-BUCKET",
                "Key": "test/"
             }
          }
       }
   },
   {
       "Type": "TAG",
       "TagStepDetails": {
          "Name": "TagStep",
          "Tags": [
             {
                "Key": "name",
                "Value": "demo"
             }
          ],
          "SourceFileLocation": "${previous.file}"
       }
   },
   {
      "Type": "DELETE",
      "DeleteStepDetails":{
```

```
        "Name":"DeleteStep",
        "SourceFileLocation": "${original.file}"
      }
  }
]
```

第一步是将上传的文件复制到新的 Amazon S3 位置。第二步将标签（键值对）添加到复制到新位置的文件 (`previous.file`)。最后，第三步删除原始文件 (`original.file`)。

2. 使用保存的文件创建工作流程。将每个 *user input placeholder* 替换为您自己的信息。

```
aws transfer create-workflow --description "short-description" --steps
 file://path-to-file --region region-ID
```

例如：

```
aws transfer create-workflow --description "copy-tag-delete workflow" --steps
 file://tagAndMoveWorkflow.json --region us-east-1
```

> **Note**
>
> 有关使用文件加载参数的更多详细信息，请参阅如何从文件加载参数。

3. 更新现有服务器。

> **Note**
>
> 此步骤假设您已经有一台 Transfer Family 服务器，并且想要将工作流程与之关联。如果不是，请参阅 配置 SFTP、FTPS 或 FTP 服务器端点。将每个 *user input placeholder* 替换为您自己的信息。

```
aws transfer update-server --server-id server-ID --region region-ID
  --workflow-details '{"OnUpload":[{ "WorkflowId": "workflow-
ID","ExecutionRole": "execution-role-ARN"}]}'
```

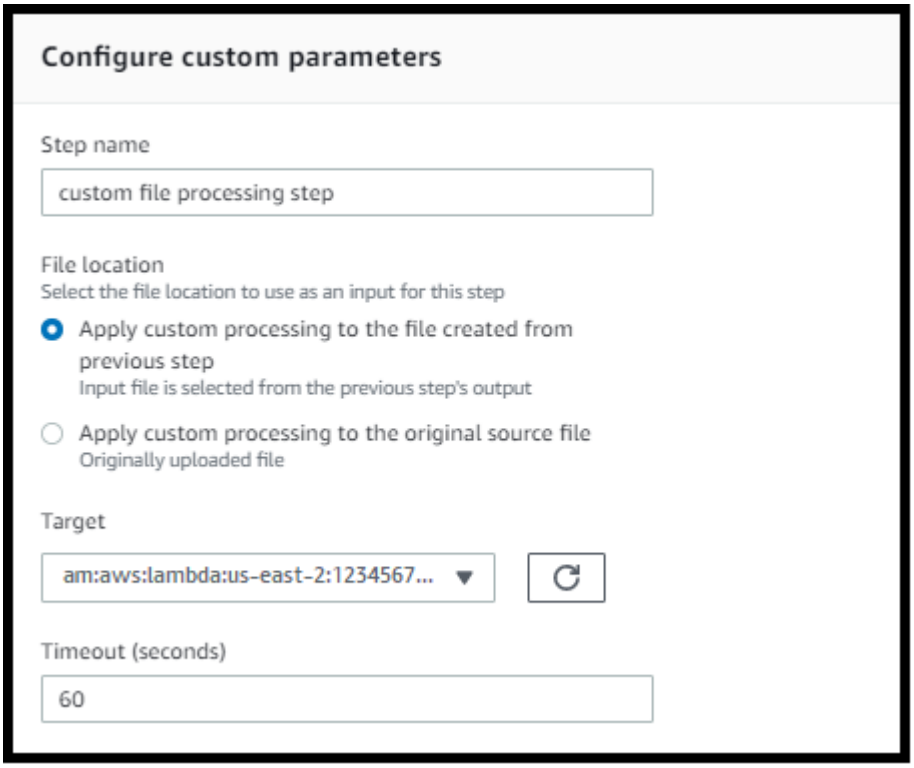
例如：

```
aws transfer update-server --server-id s-1234567890abcdef0 --region us-east-2
```

```
  --workflow-details '{"OnUpload":[{ "WorkflowId": "w-
abcdef01234567890","ExecutionRole": "arn:aws:iam::111111111111:role/nikki-wolf-
execution-role"}]}'
```

使用自定义文件处理步骤

通过使用自定义文件处理步骤，您可以使用 AWS Lambda自带文件处理逻辑。文件到达后，Transfer Family 服务器会调用包含自定义文件处理逻辑的 Lambda 函数，例如加密文件、扫描恶意软件或检查文件类型是否不正确。在以下示例中，目标 AWS Lambda 函数用于处理上一步的输出文件。



> **Note**
>
> 有关示例 Lambda 函数，请参阅 自定义工作流程步骤的 Lambda 函数示例。有关事件示例（包括传递到 Lambda 的文件的位置），请参阅 文件上传 AWS Lambda 时发送到的事件示例。

使用自定义工作流程步骤，您必须配置 Lambda 函数以调用 SendWorkflowStepStateAPI 操作。 SendWorkflowStepState通知工作流程执行该步骤已完成，状态为成功或失

败。SendWorkflowStepState API 操作的状态根据 Lambda 函数的结果调用异常处理程序步骤或线性序列中的标称步骤。

如果 Lambda 函数失败或超时，则该步骤将失败，您将在日志StepErrored中看到。 CloudWatch 如果 Lambda 函数是标称步骤的一部分，并且函数响应 SendWorkflowStepState 为 Status="FAILURE" 或超时，则流程会继续执行异常处理程序步骤。在这种情况下，工作流不会继续执行剩余的（如果有）标称步骤。有关更多详细信息，请参阅工作流程的异常处理。

在调用 SendWorkflowStepState API 操作时，必须发送以下参数：

```
{
    "ExecutionId": "string",
    "Status": "string",
    "Token": "string",
    "WorkflowId": "string"
}
```

您可以从 Lambda 函数执行时传递的输入事件中提取 ExecutionId、Token 和 WorkflowId（以下各节显示了示例）。该Status值可以是 SUCCESS 或FAILURE。

为了能够从 Lambda 函数调用 SendWorkflowStepState API 操作，您必须使用在引入托管工作流程之后发布的 AWS SDK 版本。

连续使用多个 Lambda 函数

当您依次使用多个自定义步骤时，“文件位置”选项的工作方式与仅使用单个自定义步骤时不同。Transfer Family 不支持传回 Lambda 处理过的文件以用作下一步的输入。因此，如果您将多个自定义步骤全部配置为使用 previous.file 选项，则它们都使用相同的文件位置（第一个自定义步骤的输入文件位置）。

> **Note**
>
> 如果您在自定义步骤之后有预定义的步骤（标记、复制、解密或删除），则 previous.file 设置的工作方式也会有所不同。如果将预定义步骤配置为使用 previous.file 设置，则预定义步骤将使用与自定义步骤相同的输入文件。来自自定义步骤的已处理文件不会传递到预定义的步骤。

在自定义处理后访问文件

如果您使用 Amazon S3 作为存储，并且您的工作流程包括对最初上传的文件执行操作的自定义步骤，则后续步骤将无法访问该已处理的文件。也就是说，自定义步骤之后的任何步骤都不能从自定义步骤输出中引用更新的文件。

例如，假设您的工作流程中有以下三个步骤。

- 步骤 1 - 上传名为 `example-file.txt` 的文件。
- 步骤 2 - 调用以某种方式更改`example-file.txt`的 Lambda 函数。
- 步骤 3 - 尝试对 `example-file.txt` 的更新版本执行进一步处理。

如果将步骤 3 的 `sourceFileLocation` 配置为 `${original.file}`，则步骤 3 将使用步骤 1 中服务器将文件上传到存储器时的原始文件位置。如果您在步骤 3 使用 `${previous.file}`，则步骤 3 会重复使用步骤 2 用作输入的文件位置。

因此，步骤 3 会导致错误。例如，如果步骤 3 尝试复制更新的 `example-file.txt`，则会收到以下错误：

```
{
    "type": "StepErrored",
    "details": {
        "errorType": "NOT_FOUND",
        "errorMessage": "ETag constraint not met (Service: null; Status Code: 412;
 Error Code: null; Request ID: null; S3 Extended Request ID: null; Proxy: null)",
        "stepType": "COPY",
        "stepName": "CopyFile"
    },
```

之所以出现此错误，是因为自定义步骤修改了 `example-file.txt` 的实体标签 (ETag)，使其与原始文件不匹配。

> **Note**
>
> 如果您使用的是 Amazon EFS，则不会发生这种情况，因为 Amazon EFS 不使用实体标签来识别文件。

文件上传 AWS Lambda 时发送到的事件示例

以下示例显示了文件上传完成 AWS Lambda 时发送到的事件。一个示例使用 Transfer Family 服务器，其中域名配置了 Amazon S3。另一个示例使用 Transfer Family 服务器，其中域名使用 Amazon EFS。

Custom step that uses an Amazon S3 domain

```
{
    "token": "MzI0Nzc4ZDktMGRmMi00MjFhLTgxMjUtYWZmZmRmODNkYjc0",
    "serviceMetadata": {
        "executionDetails": {
            "workflowId": "w-1234567890example",
            "executionId": "abcd1234-aa11-bb22-cc33-abcdef123456"
        },
        "transferDetails": {
            "sessionId": "36688ff5d2deda8c",
            "userName": "myuser",
            "serverId": "s-example1234567890"
        }
    },
    "fileLocation": {
        "domain": "S3",
        "bucket": "DOC-EXAMPLE-BUCKET",
        "key": "path/to/mykey",
        "eTag": "d8e8fca2dc0f896fd7cb4cb0031ba249",
        "versionId": null
    }
}
```

Custom step that uses an Amazon EFS domain

```
{
    "token": "MTg0N2Y3N2UtNWI5Ny00ZmZlLTk5YTgtZTU3YzViYjllNmZm",
    "serviceMetadata": {
        "executionDetails": {
            "workflowId": "w-1234567890example",
            "executionId": "abcd1234-aa11-bb22-cc33-abcdef123456"
        },
        "transferDetails": {
            "sessionId": "36688ff5d2deda8c",
            "userName": "myuser",
```

```
            "serverId": "s-example1234567890"
        }
    },
    "fileLocation": {
        "domain": "EFS",
        "fileSystemId": "fs-1234567",
        "path": "/path/to/myfile"
    }
}
```

自定义工作流程步骤的 Lambda 函数示例

以下 Lambda 函数提取有关执行状态的信息，然后调用 SendWorkflowStepStateAPI 操作将该步骤SUCCESS的状态返回到工作流程，可以是或。FAILURE在您的函数调用 SendWorkflowStepState API 操作之前，您可以将 Lambda 配置为根据您的工作流程逻辑执行操作。

```
import json
import boto3

transfer = boto3.client('transfer')

def lambda_handler(event, context):
    print(json.dumps(event))

    # call the SendWorkflowStepState API to notify the workflow about the step's
 SUCCESS or FAILURE status
    response = transfer.send_workflow_step_state(
        WorkflowId=event['serviceMetadata']['executionDetails']['workflowId'],
        ExecutionId=event['serviceMetadata']['executionDetails']['executionId'],
        Token=event['token'],
        Status='SUCCESS|FAILURE'
    )

    print(json.dumps(response))

    return {
      'statusCode': 200,
      'body': json.dumps(response)
    }
```

自定义步骤的 IAM 权限

若要使调用 Lambda 的步骤成功，请确保您的工作流程的执行角色包含以下权限。

```
{
            "Sid": "Custom",
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Resource": [
                "arn:aws:lambda:region:account-id:function:function-name"
            ]
        }
```

适用于工作流程的 IAM 策略

向服务器添加工作流程时，必须选择执行角色。服务器在执行工作流程时使用此角色。如果该角色没有适当的权限，则 AWS Transfer Family 无法运行工作流程。

本节介绍一组可能的 AWS Identity and Access Management (IAM) 权限，您可以使用这些权限来执行工作流程。本主题的后续部分中描述了其他示例。

> **Note**
>
> 如果您的 Amazon S3 文件有标签，则需要在 IAM 策略中添加一两个权限。
>
> - 为未进行版本控制的 Amazon S3 文件添加 `s3:GetObjectTagging`。
> - 为进行版本控制的 Amazon S3 文件添加 `s3:GetObjectVersionTagging`。

为您的工作流程创建执行角色

1. 创建新的 IAM 角色，并将 AWS 托管策略`AWSTransferFullAccess`添加到该角色中。有关创建 IAM 角色的更多信息，请参见 the section called “创建 IAM 角色和策略”。
2. 按以下策略创建其他策略，然后将其内联至您的角色。将每个 *user input placeholder* 替换为您自己的信息。

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

```
    "Statement": [
        {
            "Sid": "ConsoleAccess",
            "Effect": "Allow",
            "Action": "s3:GetBucketLocation",
            "Resource": "*"
        },
        {
            "Sid": "ListObjectsInBucket",
            "Effect": "Allow",
            "Action": "s3:ListBucket",
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET"
            ]
        },
        {
            "Sid": "AllObjectActions",
            "Effect": "Allow",
            "Action": "s3:*Object",
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ]
        },
        {
            "Sid": "GetObjectVersion",
            "Effect": "Allow",
            "Action": "s3:GetObjectVersion",
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ]
        },
        {
            "Sid": "Custom",
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Resource": [
                "arn:aws:lambda:region:account-id:function:function-name"
            ]
        },
        {
            "Sid": "Tag",
            "Effect": "Allow",
```

```
            "Action": [
                "s3:PutObjectTagging",
                "s3:PutObjectVersionTagging"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ]
        }
    ]
}
```

3. 保存此角色并在向服务器添加工作流程时将其指定为执行角色。

> **Note**
>
> 在构建 IAM 角色时，AWS 建议您尽可能限制工作流程对资源的访问权限。

工作流程信任关系

工作流程执行角色还需要与 `transfer.amazonaws.com` 建立信任关系。若要为 AWS Transfer Family建立信任关系，请参见建立信任关系。

在建立信任关系的同时，您也可以采取措施避免混淆代理问题。有关此问题的描述以及如何避免该问题的示例，请参见 the section called “防止跨服务混淆代理”。

执行角色示例：解密、复制和标记

如果您的工作流程包括标记、复制和解密步骤，则可以使用以下 IAM 策略。将每个 *user input placeholder* 替换为您自己的信息。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "CopyRead",
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:GetObjectTagging",
                "s3:GetObjectVersionTagging"
            ],
```

```
            "Resource": "arn:aws:s3:::source-bucket-name/*"
        },
        {
            "Sid": "CopyWrite",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:PutObjectTagging"
            ],
            "Resource": "arn:aws:s3:::destination-bucket-name/*"
        },
        {
            "Sid": "CopyList",
            "Effect": "Allow",
            "Action": "s3:ListBucket",
            "Resource": [
                "arn:aws:s3:::source-bucket-name",
                "arn:aws:s3:::destination-bucket-name"
            ]
        },
        {
            "Sid": "Tag",
            "Effect": "Allow",
            "Action": [
                "s3:PutObjectTagging",
                "s3:PutObjectVersionTagging"
            ],
            "Resource": "arn:aws:s3:::destination-bucket-name/*",
            "Condition": {
                "StringEquals": {
                    "s3:RequestObjectTag/Archive": "yes"
                }
            }
        },
        {
            "Sid": "ListBucket",
            "Effect": "Allow",
            "Action": "s3:ListBucket",
            "Resource": [
                "arn:aws:s3:::destination-bucket-name"
            ]
        },
        {
            "Sid": "HomeDirObjectAccess",
```

```
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:DeleteObjectVersion",
                "s3:DeleteObject",
                "s3:GetObjectVersion"
            ],
            "Resource": "arn:aws:s3:::destination-bucket-name/*"
        },
        {
            "Sid": "Decrypt",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": "arn:aws:secretsmanager:region:account-ID:secret:aws/transfer/
*"
        }
    ]
}
```

执行角色示例：运行函数并删除

在此示例中，您有一个 AWS Lambda 调用函数的工作流程。如果工作流程删除了上传的文件，并且有异常处理程序步骤可以对上一步中失败的工作流程执行采取行动，请使用以下 IAM 策略。将每个 *user input placeholder* 替换为您自己的信息。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Delete",
            "Effect": "Allow",
            "Action": [
                "s3:DeleteObject",
                "s3:DeleteObjectVersion"
            ],
            "Resource": "arn:aws:s3:::bucket-name"
        },
        {
            "Sid": "Custom",
            "Effect": "Allow",
```

```
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Resource": [
                "arn:aws:lambda:region:account-id:function:function-name"
            ]
        }
    ]
}
```

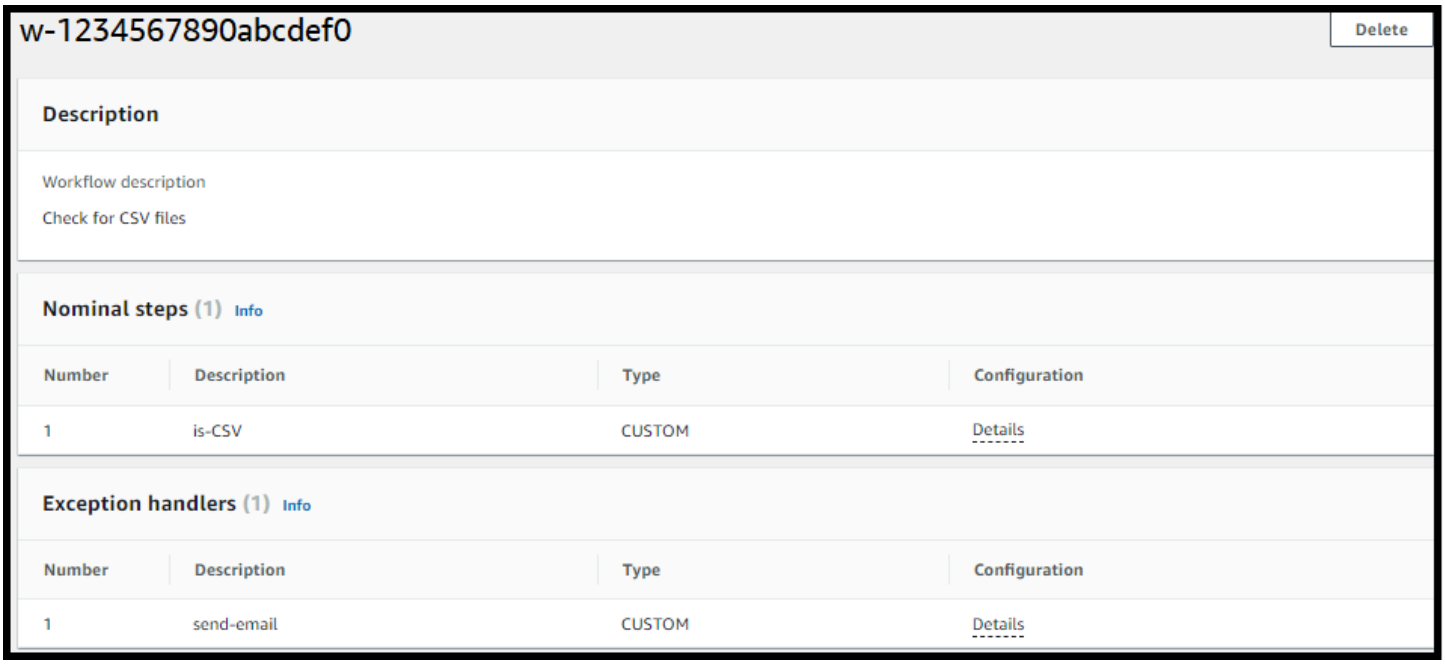
工作流程的异常处理

如果在工作流程执行过程中出现任何错误，则会执行您指定的异常处理步骤。为工作流指定错误处理步骤的方式与为工作流指定标称步骤的方式相同。例如，假设您已按名义步骤配置了自定义处理来验证传入的文件。如果文件验证失败，则异常处理步骤可以向管理员发送电子邮件。

以下示例工作流程包含两个步骤：

- 检查上传文件是否为 CSV 格式的标称步骤
- 一个异常处理步骤，用于在上传的文件不是 CSV 格式且标称步骤失败时发送电子邮件

要启动异常处理步骤，名义步骤中的 AWS Lambda 函数必须使用响应。`Status="FAILURE"`有关工作流错误处理的更多信息，请参阅the section called “使用自定义文件处理步骤”。



监控工作流程执行情况

Amazon 会实时 CloudWatch 监控您的 AWS 资源和您运行 AWS Cloud 的应用程序。您可以使用 Amazon CloudWatch 收集和跟踪指标，这些指标是您可以衡量工作流程的变量。您可以使用 Amazon 查看工作流程指标和整合日志 CloudWatch。

CloudWatch 记录工作流程

CloudWatch 为工作流程进度和结果提供统一的审计和日志记录。

查看 Amazon 工作流程 CloudWatch 日志

1. 打开亚马逊 CloudWatch 控制台，网址为 https://console.aws.amazon.com/cloudwatch/。
2. 在左侧导航窗格中选择日志，然后选择日志组。
3. 在日志组页面的导航栏上，为您的 AWS Transfer Family 服务器选择正确的区域。
4. 选择与您的服务器相对应的日志组。

   例如，如果您的服务器 ID 是 `s-1234567890abcdef0`，则您的日志组是 `/aws/transfer/s-1234567890abcdef0`。
5. 在服务器的日志组详细信息页面上，将显示最新的日志流。您正在探索的用户有两个日志流：

   - 每个 Secure Shell (SSH) 文件传输协议 (SFTP) 会话一个。
   - 一个用于正在为您的服务器执行的工作流程。工作流程的日志流格式为 *`username`*`.`*`workflowID`*`.`*`uniqueStreamSuffix`*。

   例如，如果您的用户是 `mary-major`，您具有以下日志流：

   ```
   mary-major-east.1234567890abcdef0
   mary.w-abcdef01234567890.021345abcdef6789
   ```

   > **Note**
   >
   > 此示例中列出的 16 位字母数字标识符是虚构的。您在 Amazon 上看到 CloudWatch 的值不同。

mary-major-usa-east.1234567890abcdef0 的“日志事件”页面显示每个用户会话的详细信息，mary.w-abcdef01234567890.021345abcdef6789 日志流包含工作流程的详细信息。

以下是基于包含复制步骤的工作流程 (w-abcdef01234567890) 的 mary.w-abcdef01234567890.021345abcdef6789 日志流示例。

```
{
    "type": "ExecutionStarted",
    "details": {
        "input": {
            "initialFileLocation": {
                "bucket": "DOC-EXAMPLE-BUCKET",
                "key": "mary/workflowSteps2.json",
                "versionId": "version-id",
                "etag": "etag-id"
            }
        }
    },
    "workflowId":"w-abcdef01234567890",
    "executionId":"execution-id",
    "transferDetails": {
        "serverId":"s-server-id",
        "username":"mary",
        "sessionId":"session-id"
    }
},
{
    "type":"StepStarted",
    "details": {
        "input": {
            "fileLocation": {
                "backingStore":"S3",
                "bucket":"DOC-EXAMPLE-BUCKET",
                "key":"mary/workflowSteps2.json",
                "versionId":"version-id",
                "etag":"etag-id"
            }
        },
        "stepType":"COPY",
        "stepName":"copyToShared"
    },
    "workflowId":"w-abcdef01234567890",
    "executionId":"execution-id",
```

```
    "transferDetails": {
        "serverId":"s-server-id",
        "username":"mary",
        "sessionId":"session-id"
    }
},
{
    "type":"StepCompleted",
    "details":{
        "output":{},
        "stepType":"COPY",
        "stepName":"copyToShared"
    },
    "workflowId":"w-abcdef01234567890",
    "executionId":"execution-id",
    "transferDetails":{
        "serverId":"server-id",
        "username":"mary",
        "sessionId":"session-id"
    }
},
{
    "type":"ExecutionCompleted",
    "details": {},
    "workflowId":"w-abcdef01234567890",
    "executionId":"execution-id",
    "transferDetails":{
        "serverId":"s-server-id",
        "username":"mary",
        "sessionId":"session-id"
    }
}
```

CloudWatch 工作流程指标

AWS Transfer Family 为工作流程提供了多个指标。您可以查看前一分钟有多少工作流程执行启动、成功完成和失败的指标。中描述了 Transfer Family 的所有 CloudWatch 指标使用 T CloudWatch ransfer Family 的指标。

通过模板创建工作流

您可以部署用于创建工作流的 AWS CloudFormation 堆栈和基于模板的服务器。此过程包含一个示例，您可以使用该示例来快速部署工作流程。

创建用于创建 AWS Transfer Family 工作流程和服务器的 AWS CloudFormation 堆栈

1. 打开 AWS CloudFormation 控制台，网址为 https://console.aws.amazon.com/cloudformation。
2. 将以下代码保存到文件中。

   YAML

```
AWSTemplateFormatVersion: 2010-09-09
Resources:
  SFTPServer:
    Type: 'AWS::Transfer::Server'
    Properties:
      WorkflowDetails:
        OnUpload:
          - ExecutionRole: workflow-execution-role-arn
            WorkflowId: !GetAtt
              - TransferWorkflow
              - WorkflowId
  TransferWorkflow:
    Type: AWS::Transfer::Workflow
    Properties:
      Description: Transfer Family Workflows Blog
      Steps:
        - Type: COPY
          CopyStepDetails:
            Name: copyToUserKey
            DestinationFileLocation:
              S3FileLocation:
                Bucket: archived-records
                Key: ${transfer:UserName}/
            OverwriteExisting: 'TRUE'
        - Type: TAG
          TagStepDetails:
            Name: tagFileForArchive
            Tags:
              - Key: Archive
                Value: yes
```

```
        - Type: CUSTOM
          CustomStepDetails:
            Name: transferExtract
            Target: arn:aws:lambda:region:account-id:function:function-name
            TimeoutSeconds: 60
        - Type: DELETE
          DeleteStepDetails:
            Name: DeleteInputFile
            SourceFileLocation: '${original.file}'
      Tags:
        - Key: Name
          Value: TransferFamilyWorkflows
```

JSON

```
{
    "AWSTemplateFormatVersion": "2010-09-09",
    "Resources": {
        "SFTPServer": {
            "Type": "AWS::Transfer::Server",
            "Properties": {
                "WorkflowDetails": {
                    "OnUpload": [
                        {
                            "ExecutionRole": "workflow-execution-role-arn",
                            "WorkflowId": {
                                "Fn::GetAtt": [
                                    "TransferWorkflow",
                                    "WorkflowId"
                                ]
                            }
                        }
                    ]
                }
            }
        },
        "TransferWorkflow": {
            "Type": "AWS::Transfer::Workflow",
            "Properties": {
                "Description": "Transfer Family Workflows Blog",
                "Steps": [
                    {
                        "Type": "COPY",
```

```
                        "CopyStepDetails": {
                            "Name": "copyToUserKey",
                            "DestinationFileLocation": {
                                "S3FileLocation": {
                                    "Bucket": "archived-records",
                                    "Key": "${transfer:UserName}/"
                                }
                            },
                            "OverwriteExisting": "TRUE"
                        }
                    },
                    {
                        "Type": "TAG",
                        "TagStepDetails": {
                            "Name": "tagFileForArchive",
                            "Tags": [
                                {
                                    "Key": "Archive",
                                    "Value": "yes"
                                }
                            ]
                        }
                    },
                    {
                        "Type": "CUSTOM",
                        "CustomStepDetails": {
                            "Name": "transferExtract",
                            "Target": "arn:aws:lambda:region:account-
id:function:function-name",
                            "TimeoutSeconds": 60
                        }
                    },
                    {
                        "Type": "DELETE",
                        "DeleteStepDetails": {
                            "Name": "DeleteInputFile",
                            "SourceFileLocation": "${original.file}"
                        }
                    }
                ],
                "Tags": [
                    {
                        "Key": "Name",
                        "Value": "TransferFamilyWorkflows"
```

```
                    }
                ]
            }
        }
    }
}
```

3. 将以下值替换为您的实际值。

   - 将 *workflow-execution-role-arn* 替换为实际工作流执行角色的 ARN。例如，arn:aws:transfer:us-east-2:111122223333:workflow/w-1234567890abcdef0
   - 将 arn:aws:lambda:*region*:*account-id*:function:*function-name* 替换为 Lambda 函数的 ARN。例如，arn:aws:lambda:us-east-2:123456789012:function:example-lambda-idp。

4. 按照《AWS CloudFormation 用户指南》中的选择 AWS CloudFormation 堆栈模板中的使用现有模板部署堆栈的说明进行操作。

部署堆栈后，您可以在 CloudFormation 控制台的 Outputs 选项卡中查看有关堆栈的详细信息。该模板创建了一个使用服务管理用户的新 AWS Transfer Family SFTP 服务器和一个新的工作流程，并将该工作流与新服务器相关联。

从 Transfer Family 服务器中移除工作流

如果您已将工作流程与 Transfer Family 服务器关联，而现在想要移除该关联，则可以使用控制台或以编程方式执行此操作。

Console

若要从 Transfer Family 服务器中移除工作流

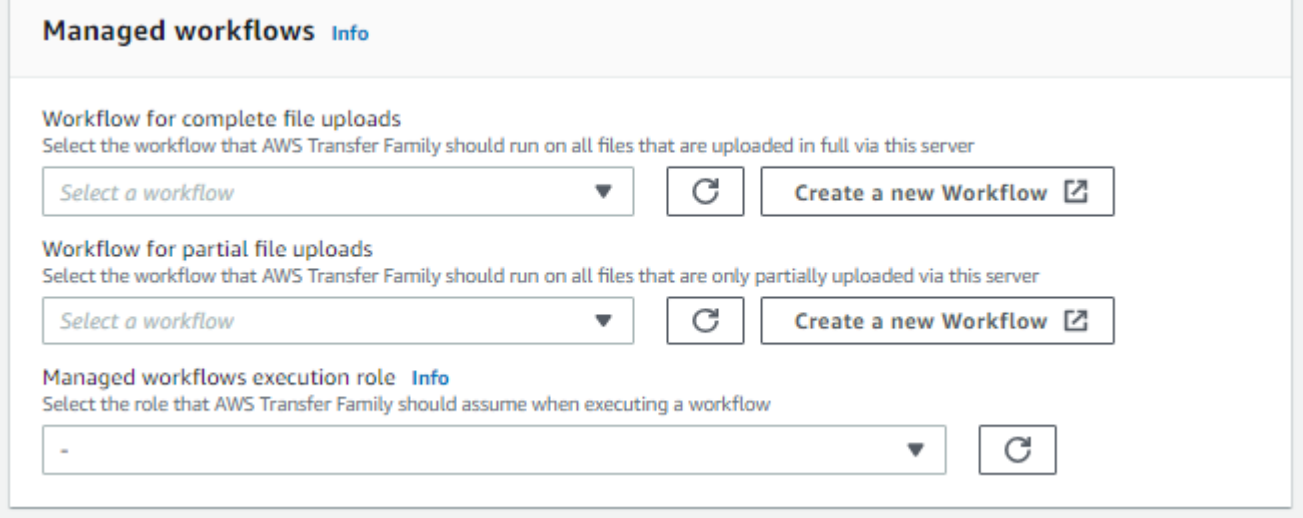
1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在左侧导航窗格中选择服务器。
3. 在“服务器 ID”列中选择服务器的标识符。
4. 在服务器的详细信息页面上，向下滚动到其他详细信息部分，然后选择编辑。
5. 在编辑其他详细信息页面中的托管工作流程部分，清除所有设置的信息：

   - 从用于完整文件上载的工作流的工作流列表中选择短划线 (-)。

- 如果尚未清除，从用于部分文件上载的工作流的工作流列表中选择短划线 (-)。
- 从托管工作流程执行角色的角色列表中选择短划线 (-)。

如果看不到破折号，请向上滚动直到看到它，因为它是每个菜单中的第一个值。

该部分应该类似以下内容。



6. 要保存更改，请向下滚动并选择保存。

CLI

您可以使用 `update-server`（或 `UpdateServer` for API）调用，并为 `OnUpload` 和 `OnPartialUpload` 参数提供空参数。

从中 AWS CLI，运行以下命令：

```
aws transfer update-server --server-id your-server-id --workflow-details
 '{"OnPartialUpload":[],"OnUpload":[]}'
```

将 *`your-server-id`* 替换为服务器的 ID。例如，如果您的服务器 ID 是 `s-01234567890abcdef`，则命令如下所示：

```
aws transfer update-server --server-id s-01234567890abcdef --workflow-details
 '{"OnPartialUpload":[],"OnUpload":[]}'
```

托管工作流限制和局限性

限制

以下限制目前适用于 AWS Transfer Family的上传后处理工作流程。

- 不支持跨账户和跨区域 AWS Lambda 功能。但是，您可以跨账户复制，前提是您的 AWS Identity and Access Management (IAM) 策略配置正确。
- 对于所有工作流程步骤，工作流程访问的任何 Amazon S3 存储桶都必须与工作流程本身位于同一区域。
- 对于解密步骤，解密目标必须与区域和后备存储的来源相匹配（例如，如果要解密的文件存储在 Amazon S3 中，则指定的目标也必须在 Amazon S3 中）。
- 仅支持异步自定义步骤。
- 自定义步骤超时值是近似值。也就是说，超时所需的时间可能比指定时间稍长。此外，工作流程依赖于 Lambda 函数。因此，如果函数在执行过程中出现延迟，则工作流程不会意识到延迟。
- 如果您超过了限制限制，Transfer Family 不会将工作流程操作添加到队列中。
- 不会为大小为 0 的文件启动工作流程。大小大于 0 的文件会启动关联的工作流程。

限制

此外，以下功能限制适用于 Transfer Family 的工作流程：

- 每个区域、每个账户的工作流程数量限制为 10。
- 自定义步骤的最大超时时间为 30 分钟。
- 工作流中的最大步骤数为 8。
- 每个工作组的最大标签数是 50。
- 每个工作流程中包含解密步骤的最大并发执行数为 250 个。
- 在每台 Transfer Family 服务器上，每位用户最多可存储 3 个 PGP 私钥。
- 数据文件的最大大小为 10 GB。
- 我们使用容量暴增为 100、再填充率为 1 的令牌桶系统来限制新的执行率。
- 无论何时从服务器上移除工作流程并用新的工作流程替换它，或者更新服务器配置（这会影响工作流程的执行角色），都必须等待大约 10 分钟才能执行新的工作流程。Transfer Family 服务器会缓存工作流程细节，服务器需要 10 分钟才能刷新其缓存。

  此外，您必须注销所有活动的 SFTP 会话，然后等待 10 分钟重新登录才能看到更改。

管理服务器

在本部分中，您可以找到有关如何查看服务器列表、如何查看服务器详细信息、如何编辑服务器详细信息以及如何更改启用 SFTP 的服务器的主机密钥的信息。

主题

查看服务器列表

在 AWS Transfer Family 控制台上，您可以找到位于所选 AWS 区域内的所有服务器的列表。

要查找某个 AWS 地区中存在的服务器的列表

- 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。

  如果您在当前 AWS 区域有一台或多台服务器，则控制台会打开并显示您的服务器列表。如果未看到服务器列表，请确保您已进入正确的区域。也可以从导航窗格中选择 Servers (服务器)。

  有关查看您服务器详情的更多信息，请参阅 查看 SFTP、FTPS 和 FTP 服务器的详细信息。

删除服务器

此过程说明如何使用 AWS Transfer Family 控制台或删除 Transfer Family 服务器 AWS CLI。

> **Important**
>
> 在您删除服务器之前，您需要为允许访问您的端点的每项协议付费。

> **Warning**
>
> 删除服务器会导致其所有用户都被删除。使用服务器访问的存储桶中的数据不会被删除，拥有这些 Amazon S3 存储桶权限的 AWS 用户仍可以访问这些数据。

Console

使用控制台删除服务器

1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在左侧导航窗格中，选择服务器。
3. 选中您要删除的服务器的复选框。
4. 对于操作，选择删除。
5. 在显示的确认对话框中，输入单词 **delete**，然后选择删除以确认您要删除该用户。

服务器已从服务器页面中删除，您无需再为此付费。

AWS CLI

使用 CLI 删除服务器

1. （可选）运行以下命令查看要永久删除的服务器的详细信息。

```
aws transfer describe-server --server-id your-server-id
```

此`describe-server`命令会返回您的服务器的所有详细信息。

2. 运行以下命令删除服务器。

```
aws transfer delete-server --server-id your-server-id
```

如果成功，该命令将删除服务器并且不返回任何信息。

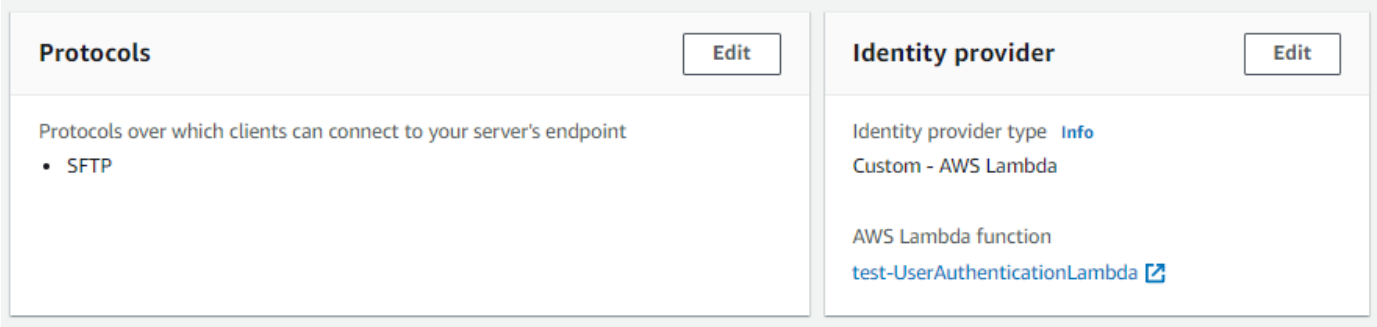
查看 SFTP、FTPS 和 FTP 服务器的详细信息

您可以找到单个 AWS Transfer Family 服务器的详细信息和属性的列表。服务器属性包括协议、身份提供商、状态、端点类型、自定义主机名、端点、用户、日志记录角色、服务器主机密钥和标签。

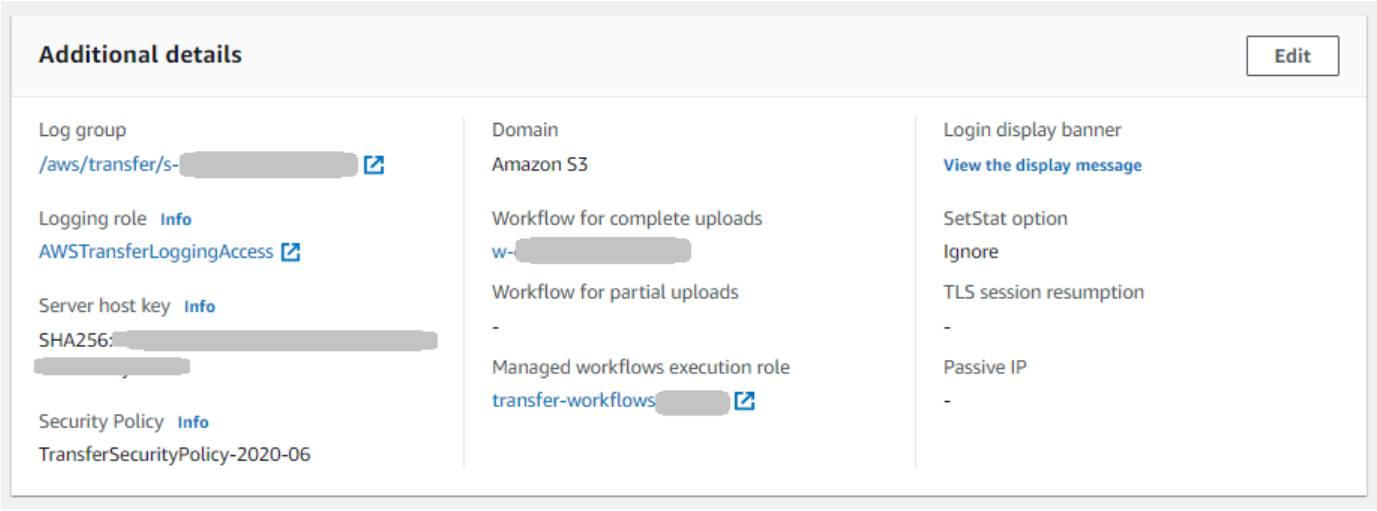
查看服务器详细信息

1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在导航窗格中，选择 Servers (服务器)。
3. 选择服务器 ID 列中的标识符以查看服务器详细信息页面，如下所示。

   您可以通过选择编辑来更改服务器的属性。有关编辑服务器详情的更多信息，请参阅 编辑服务器详细信息。AS2 服务器的详细信息页面略有不同。对于 AS2 服务器，请参见查看 AS2 服务器的详细信息。

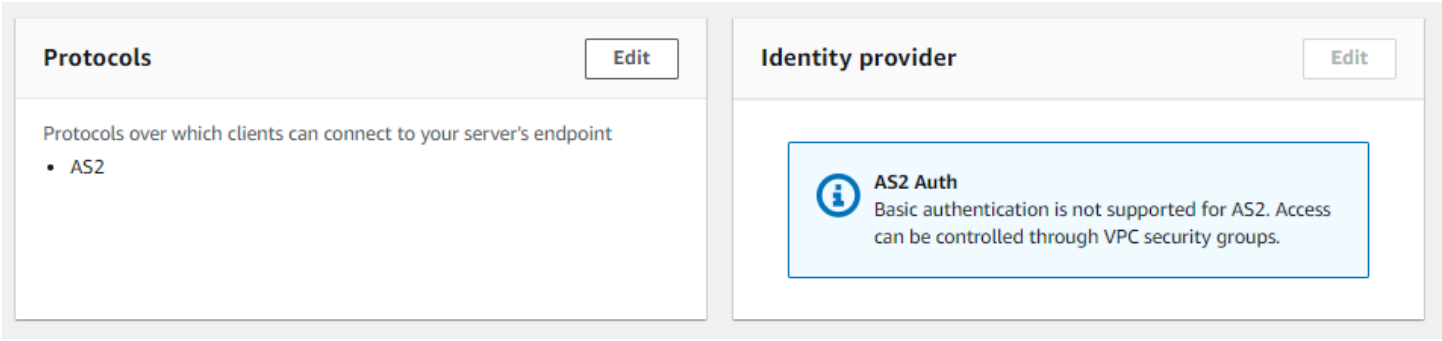


> **Note**
>
> 自 2022 年 9 月起，服务器主机密钥的描述和导入日期值是新的。引入这些值是为了支持多主机密钥功能。此功能需要迁移在引入多个主机密钥之前使用的所有单个主机密钥。
> 已迁移服务器主机密钥的导入日期值设置为服务器的上次修改日期。也就是说，您看到的迁移主机密钥的日期与服务器主机密钥迁移之前上次以任何方式修改服务器的日期相对应。
> 迁移的唯一密钥是您最旧的或唯一的服务器主机密钥。任何其他密钥的实际日期均从您导入时算起。此外，迁移后的密钥具有描述，便于将其识别为已迁移。
> 迁移发生在 9 月 2 日至 9 月 13 日之间。此范围内的实际迁移日期取决于服务器所在的地区。

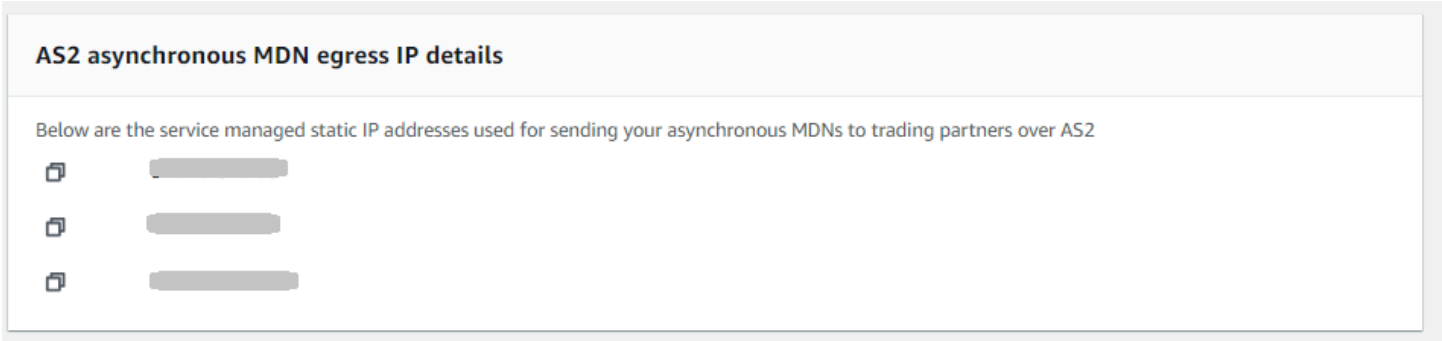


查看 AS2 服务器的详细信息

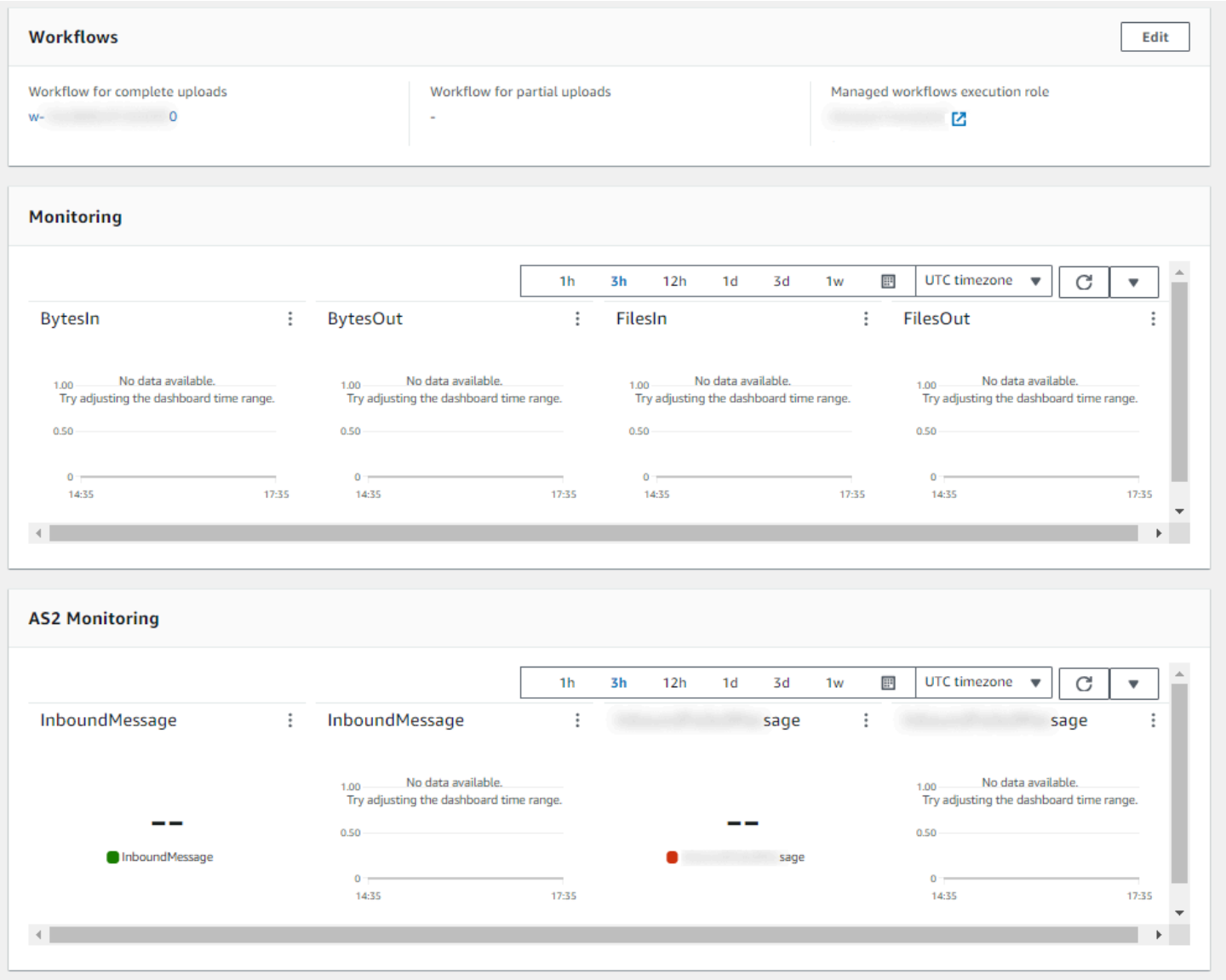
您可以找到单个 AWS Transfer Family 服务器的详细信息和属性的列表。服务器属性包括协议、状态等。对于 AS2 服务器，您还可以查看 AS2 异步 MDN 出口 IP 地址。



每台 AS2 服务器都分配了三个静态 IP 地址。使用这些 IP 地址通过 AS2 向您的贸易伙伴发送异步 mDN。



AS2 服务器详细信息页面的底部包含任何附加工作流程的详细信息以及监控和标记信息。



编辑服务器详细信息

创建 AWS Transfer Family 服务器后，可以编辑服务器配置。

主题

- 编辑文件传输协议
- 编辑自定义身份提供商参数
- 编辑服务器端点
- 编辑日志记录配置
- 编辑安全策略

- 更改服务器的托管工作流程
- 更改服务器的显示横幅
- 将服务器联机或脱机

编辑服务器的配置

1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在左侧导航窗格中，选择服务器。
3. 选择服务器 ID 列中的标识符以查看服务器详细信息页面，如下所示。

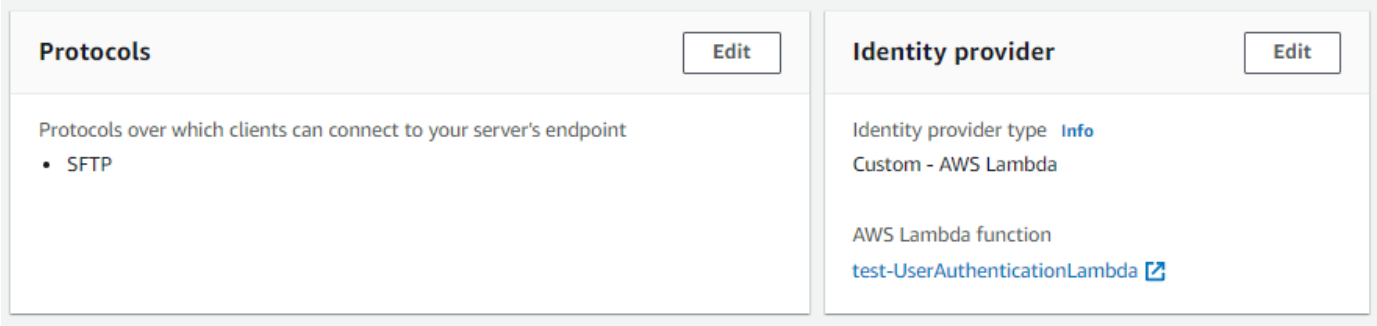
   您可以通过选择编辑来更改服务器的属性：

   - 要更改协议，请参阅编辑文件传输协议。
   - 对于身份提供商，请注意，在创建服务器后，不能更改服务器的身份提供商类型。要更改身份提供商，请删除服务器并使用所需的身份提供商创建新的服务器。

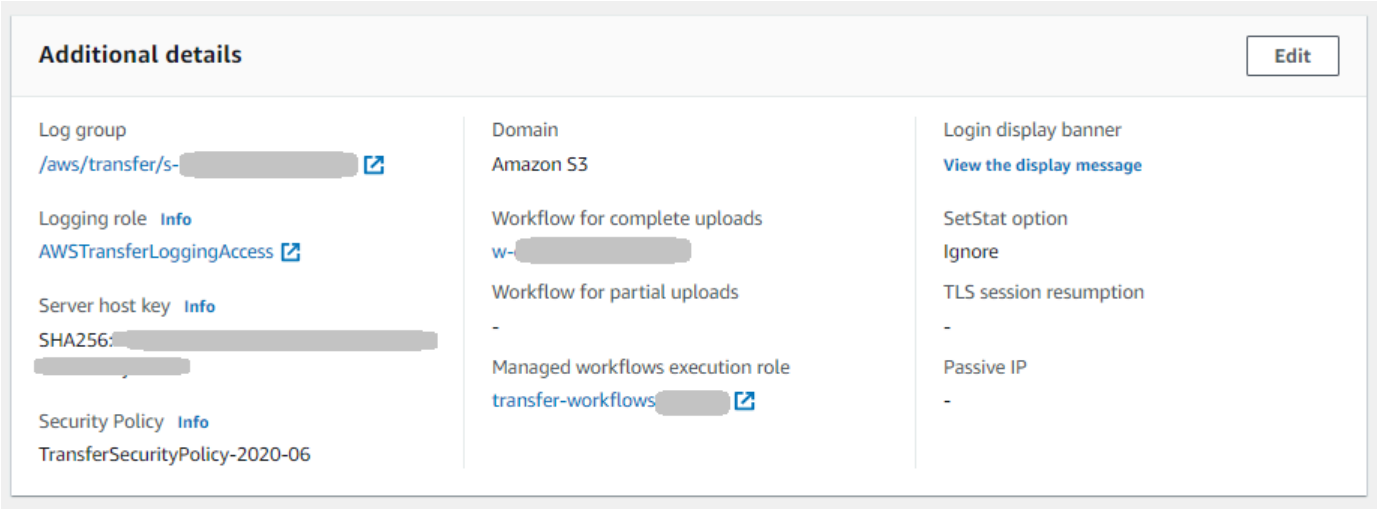
     > **Note**
     >
     > 如果您的服务器使用自定义身份提供商，则可以编辑某些属性。有关更多信息，请参阅编辑自定义身份提供商参数。

   - 要更改端点类型或自定义主机名，请参阅编辑服务器端点。
   - 要添加协议，您需要先将 AS2 作为协议添加到您的服务器。有关更多信息，请参阅 编辑文件传输协议。
   - 要管理服务器的主机密钥，请参阅管理启用 SFTP 的服务器的主机密钥。
   - 在其他详细信息下，您可以编辑以下信息：
     - 要更改日志记录角色，请参阅编辑日志记录配置。
     - 要更改安全策略，请参阅编辑安全策略。
     - 要更改服务器主机密钥，请参阅管理启用 SFTP 的服务器的主机密钥。
     - 要更改服务器的托管工作流程，请参阅更改服务器的托管工作流程。
     - 要编辑服务器的显示横幅，请参阅更改服务器的显示横幅。
   - 在其他配置下，您可以编辑以下信息：

  - SetStat 选项：启用此选项可忽略客户端尝试对您上传到 Amazon S3 存储桶的文件使用SETSTAT时生成的错误。有关其他详细信息，请参阅ProtocolDetails主题中的SetStatOption文档。
  - TLS 会话恢复：提供一种机制来恢复或共享 FTPS 会话的控制和数据连接之间协商的私有密钥。有关其他详细信息，请参阅ProtocolDetails主题中的TlsSessionResumptionMode文档。
  - 被动 IP：表示 FTP 和 FTPS 协议的被动模式。输入一个 IPv4 地址，例如防火墙、路由器或负载均衡器的公有 IP 地址。有关其他详细信息，请参阅ProtocolDetails主题中的PassiveIp文档。
- 要启动或停止服务器，请参阅将服务器联机或脱机。
- 要删除服务器，请参阅删除服务器。
- 要编辑用户的属性，请参阅管理访问控制。



> **Note**
>
> 自 2022 年 9 月起，服务器主机密钥的描述和导入日期值是新的。引入这些值是为了支持多主机密钥功能。此功能需要迁移在引入多个主机密钥之前使用的所有单个主机密钥。
> 已迁移服务器主机密钥的导入日期值设置为服务器的上次修改日期。也就是说，您看到的迁移主机密钥的日期与服务器主机密钥迁移之前上次以任何方式修改服务器的日期相对应。
> 迁移的唯一密钥是您最旧的或唯一的服务器主机密钥。任何其他密钥的实际日期均从您导入时算起。此外，迁移后的密钥具有描述，便于将其识别为已迁移。
> 迁移发生在 9 月 2 日至 9 月 13 日之间。此范围内的实际迁移日期取决于服务器所在的地区。



编辑文件传输协议

在 AWS Transfer Family 控制台上，您可以编辑文件传输协议。文件传输协议将客户端连接到服务器的端点。

编辑协议

1. 在服务器详细信息页面上，选择协议旁边的编辑。
2. 在编辑协议页面，选中或清除协议复选框或复选框以添加或删除以下文件传输协议：

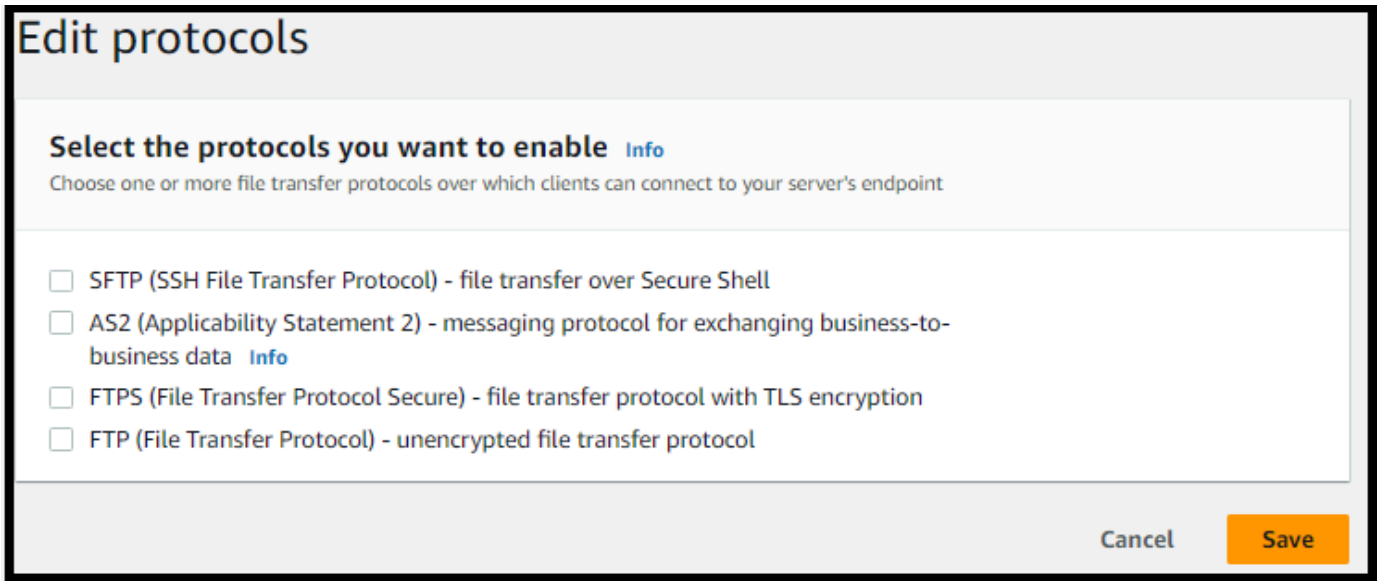
   - Secure Shell (SSH) 文件传输协议 (SFTP) — 通过 SSH 的文件传输

     有关 SFTP 的更多信息，请参阅创建启用 SFTP 的服务器。
   - 文件传输协议安全 (FTPS) — 使用 TLS 加密的文件传输功能

     有关 FTP 的更多信息，请参阅创建启用 FTPS 的服务器。
   - 文件传输协议 (FTP) — 未加密的文件传输功能

     有关 FTPS 的更多信息，请参阅创建启用 FTP 的服务器。

> **Note**
>
> 如果现有服务器仅为 SFTP 启用，并且要添加 FTPS 和 FTP，则必须确保具有与 FTPS 和 FTP 兼容的正确身份提供商和端点类型设置。



如果选择 FTPS，则必须选择存储在 AWS Certificate Manager (ACM) 中的证书，当客户端通过 FTPS 连接到服务器时，该证书将用于识别您的服务器。

要请求新的公有证书，请参阅AWS Certificate Manager 用户指南中的请求公有证书。

要将现有证书导入到 ACM 中，请参阅AWS Certificate Manager 用户指南中的将证书导入到 ACM。

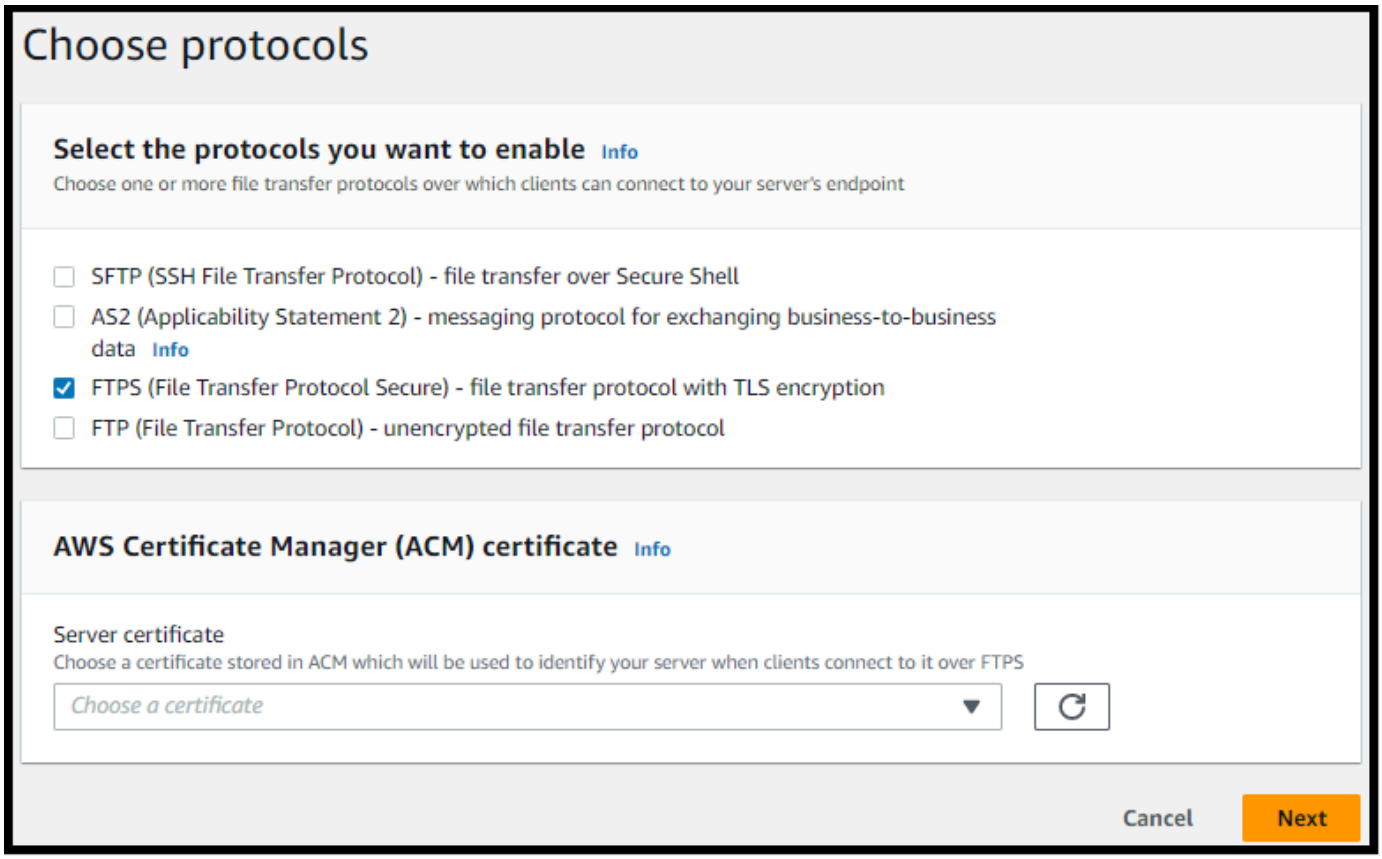
要请求私有证书以通过私有 IP 地址使用 FTPS，请参阅AWS Certificate Manager 用户指南中的请求私有证书。

支持具有以下加密算法和密钥大小的证书：

- 2048 位 RSA (RSA_2048)
- 4096 位 RSA (RSA_4096)
- Elliptic Prime Curve 256 位 (EC_prime256v1)
- Elliptic Prime Curve 384 位 (EC_secp384r1)

- Elliptic Prime Curve 521 位 (EC_secp521r1)

> **Note**
>
> 证书必须是指定了 FQDN 或 IP 地址且具有有关颁发者的信息的有效 SSL/TLS X.509 版本 3 证书。

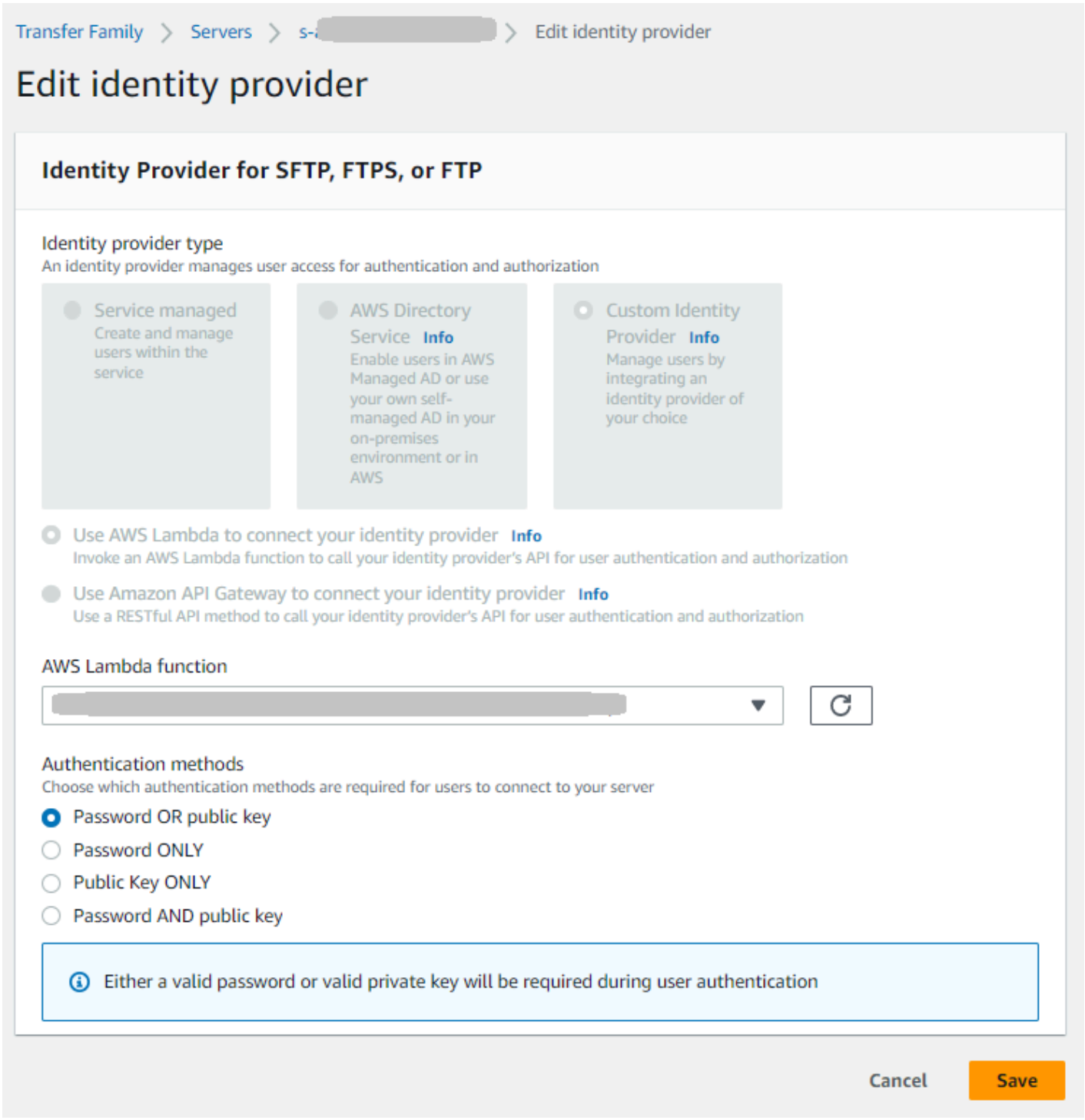


3. 选择保存。您将返回到服务器详细信息页面。

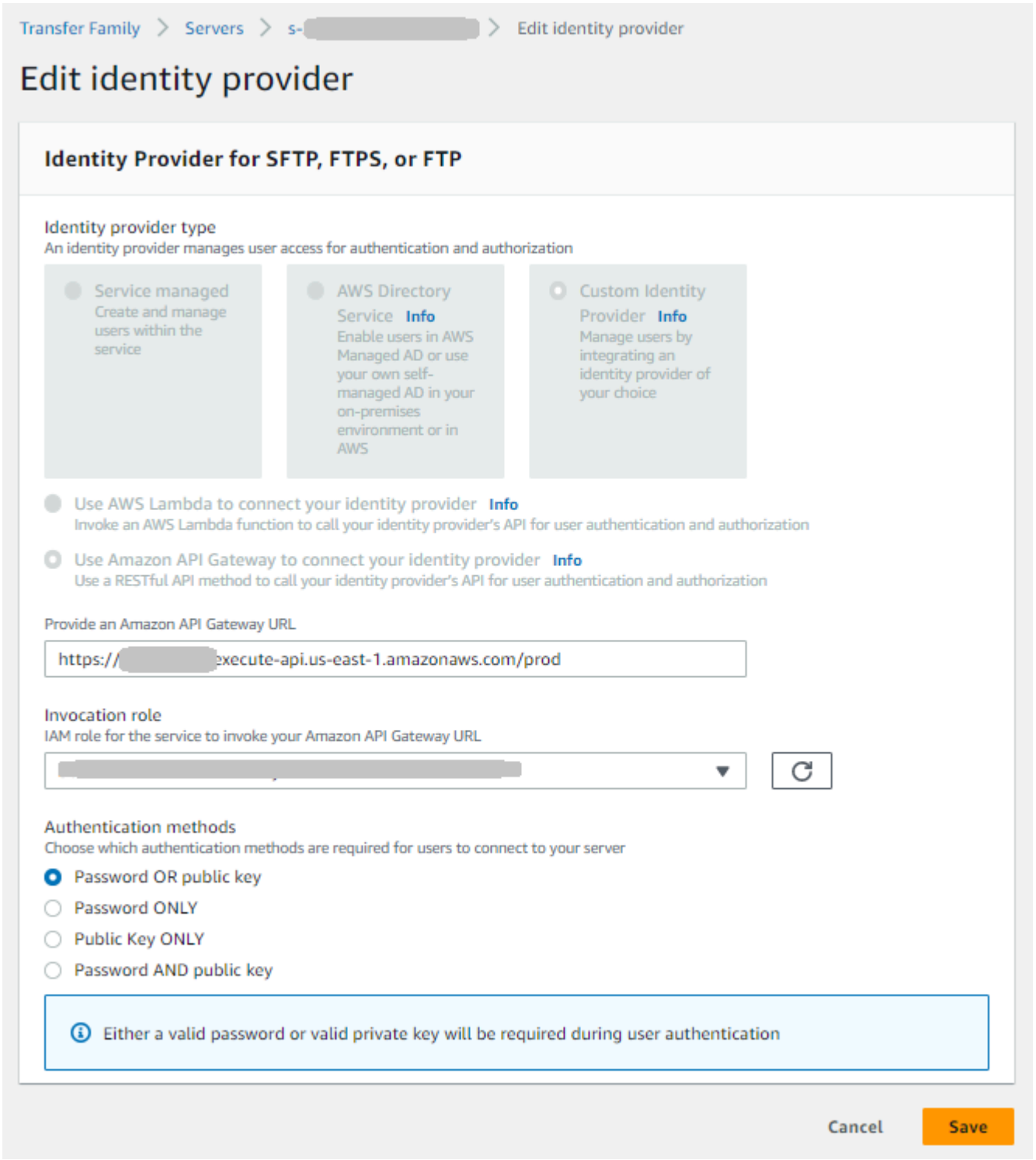
编辑自定义身份提供商参数

在 AWS Transfer Family 控制台上，对于自定义身份提供商，您可以更改某些设置，具体取决于您使用的是 Lambda 函数还是 API Gateway。无论哪种情况，如果您的服务器使用 SFTP 协议，您都可以编辑身份验证方法。

- 如果您使用 Lambda 作为身份提供商，则可以更改底层 Lambda 函数。



- 如果您使用 API Gateway 作为身份提供商，则可以更新 Gateway URL 或调用角色，或同时更新两者。



编辑服务器端点

在 AWS Transfer Family 控制台上，您可以修改服务器端点类型和自定义主机名。

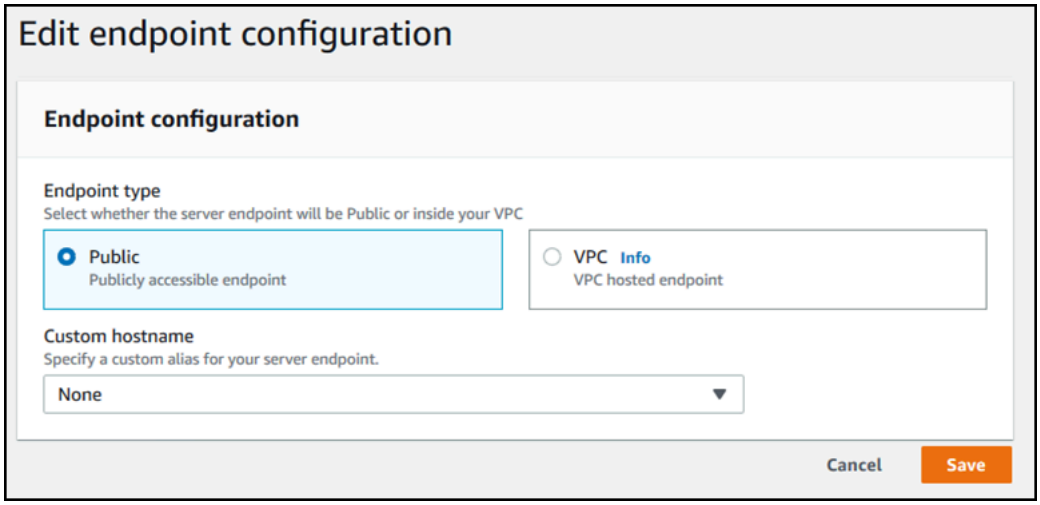
要编辑服务器端点详细信息

1. 在服务器详细信息页面上，选择端点详细信息旁边的编辑。
2. 在编辑端点配置页面上，对于端点类型，选择以下选项之一：
   - 公有 — 通过此选项，您可以通过互联网访问服务器。
   - VPC — 通过此选项，您可以访问虚拟私有云 (VPC) 中的服务器。有关 VPC 的信息，请参阅 在虚拟私有云中创建服务器。
3. 对于自定义主机名，请选择以下选项之一：
   - 无 — 如果您不想使用自定义域，请选择无。

     您将获得由提供的服务器主机名 AWS Transfer Family。服务器主机名使用格式 *serverId*.server.transfer.*regionId*.amazonaws.com。
   - Amazon Route 53 DNS 别名 — 要使用在 Route 53 中自动为您创建的 DNS 别名，请选择此选项。
   - 其他 DNS — 要使用您在外部 DNS 服务中已经拥有的主机名，请选择其他 DNS。

   选择 Amazon Route 53 DNS 别名或其他 DNS 可指定与服务器端点关联的名称解析方法。

   例如，您的自定义域可能是 sftp.inbox.example.com。自定义主机名使用由您提供并且 DNS 服务可以解析的 DNS 名称。您可以使用 Route 53 作为您 DNS 的解析程序，或者使用您自己的 DNS 服务提供商。要了解 AWS Transfer Family 如何使用 Route 53 从自定义域将流量路由到服务器端点，请参阅使用自定义主机名。



4. 选择保存。您将返回到服务器详细信息页面。

编辑日志记录配置

在 AWS Transfer Family 控制台上，您可以更改日志配置。

> **Note**
>
> 如果 Transfer Family 在你创建服务器时为你创建了 CloudWatch 日志 IAM 角色，则会调用该 IAM 角色`AWSTransferLoggingAccess`。您可以将其用于所有的 Transfer Family 服务器。

要编辑日志记录配置

1. 在服务器详细信息页面上，选择其他详细信息旁边的编辑。
2. 根据您的配置，在日志记录角色、结构化 JSON 日志记录或两者之间进行选择。有关更多信息，请参阅 更新服务器的日志记录。

编辑安全策略

此过程说明如何使用 AWS Transfer Family 控制台或更改 Transfer Family 服务器的安全策略 AWS CLI。

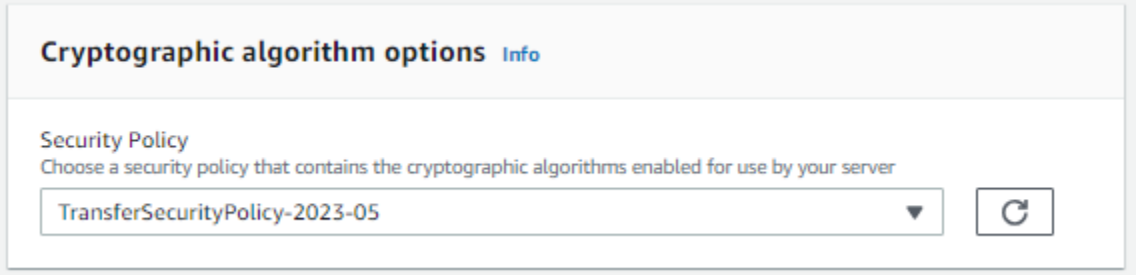
> **Note**
>
> 如果您的终端节点已启用 FIPS，则无法将 FIPS 安全策略更改为非 FIPS 安全策略。

Console

使用控制台编辑安全策略

1. 在服务器详细信息页面上，选择其他详细信息旁边的编辑。
2. 在加密算法选项部分中，请选择包含允许服务器使用的加密算法的安全策略。

   有关安全策略的更多信息，请参阅AWS Transfer Family 服务器的安全策略。



3. 选择保存。

   您将返回到服务器详细信息页面，您可以在其中查看更新的安全策略。

AWS CLI

使用 CLI 编辑安全策略

1. 运行以下命令以查看附加到您的服务器的当前安全策略。

```
aws transfer describe-server --server-id your-server-id
```

   此describe-server命令返回服务器的所有详细信息，包括以下行：

```
"SecurityPolicyName": "TransferSecurityPolicy-2018-11"
```

   在这种情况下，服务器的安全策略是TransferSecurityPolicy-2018-11。

2. 确保为命令提供安全策略的确切名称。例如，运行以下命令将服务器更新为TransferSecurityPolicy-2023-05。

```
aws transfer update-server --server-id your-server-id --security-policy-name
 "TransferSecurityPolicy-2023-05"
```

> **Note**
>
> 中列出了可用安全策略的名称AWS Transfer Family 服务器的安全策略。

如果成功，该命令将返回以下代码，并更新服务器的安全策略。

```
{
```

```
    "ServerId": "your-server-id"
}
```

更改服务器的托管工作流程

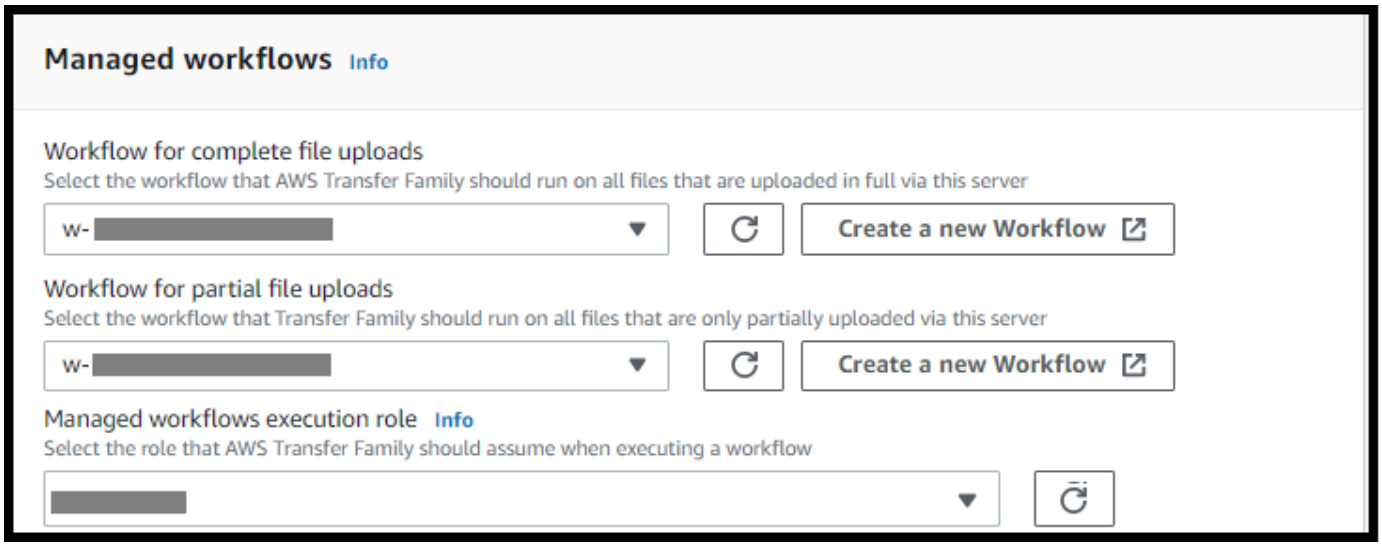
在 AWS Transfer Family 控制台上，您可以更改与服务器关联的托管工作流程。

要更改托管工作流程

1. 在服务器详细信息页面上，选择其他详细信息旁边的编辑。
2. 在编辑其他详细信息页面的托管工作流程部分，选择要在所有上传中运行的工作流程。

> **Note**
>
> 如果您还没有工作流程，请选择创建新的工作流程来创建工作流程。

   a. 选择要使用的工作流程 ID。
   b. 选择执行角色。这是 Transfer Family 在执行工作流程步骤时所扮演的角色。有关更多信息，请参阅 适用于工作流程的 IAM 策略。选择 Save（保存）。



3. 选择保存。您将返回到服务器详细信息页面。

更改服务器的显示横幅

在 AWS Transfer Family 控制台上，您可以更改与服务器关联的显示横幅。

要更改显示横幅

1. 在服务器详细信息页面上，选择其他详细信息旁边的编辑。
2. 在编辑其他详细信息页面的显示横幅部分，输入可用显示横幅的文本。
3. 选择保存。您将返回到服务器详细信息页面。

将服务器联机或脱机

在 AWS Transfer Family 控制台上，您可以将服务器联机或使其离线。

让您的服务器联机

1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在导航窗格中，选择 Servers (服务器)。
3. 选中离线服务器的复选框。
4. 对于操作，选择启动。

服务器可能需要几分钟的时间才能从脱机状态切换到联机状态。

> **Note**
>
> 当您停止服务器以使其脱机时，目前仍在为该服务器累积服务费用。要消除基于服务器的附加费用，请删除该服务器。

让服务器离线

1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在导航窗格中，选择 Servers (服务器)。
3. 选中在线服务器的复选框。
4. 对于操作，请选择停止。

如果服务器正在启动或关闭，则该服务器无法用于文件操作。控制台不显示正在启动和正在停止状态。

如果您发现错误情况`START_FAILED`或`STOP_FAILED`，请联系 AWS Support 以帮助解决您的问题。

管理启用 SFTP 的服务器的主机密钥

> **Important**
>
> 如果您不打算将现有用户从启用了 SFTP 的现有服务器迁移到启用了 SFTP 的新服务器，请忽略本部分。
>
> 意外更改服务器的主机密钥会导致中断。根据您的 SFTP 客户端的配置方式，它可能会立即失败，并显示不存在可信主机密钥的消息，或者出现威胁性提示。如果有用于自动连接的脚本，它们很可能也会失败。

默认情况下， AWS Transfer Family 为启用 SFTP 的服务器提供主机密钥。您可以使用来自其他服务器的主机密钥替换默认主机密钥。只有当您计划将现有用户从现有启用了 SFTP 的服务器迁移到新的启用了 SFTP 的服务器时才执行此操作。

为避免用户再次被提示验证启用了 SFTP 的服务器的真实性，请将本地服务器的主机密钥导入到启用了 SFTP 的服务器。这样做还可以防止您的用户收到有关潜在 man-in-the-middle 攻击的警告。

作为一项额外的安全措施，您也可以定期轮换主机密钥。

> **Note**
>
> 尽管 Transfer Family 控制台允许您为所有服务器指定和添加服务器主机密钥，但这些密钥仅适用于使用 SFTP 协议的服务器。

主题

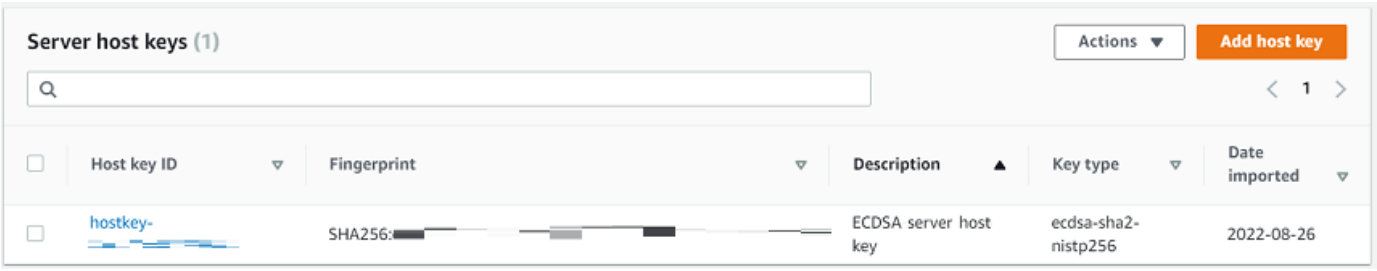
添加其他的服务器主机密钥

在 AWS Transfer Family 控制台上，您可以添加其他服务器主机密钥。添加其他不同格式的主机密钥对于在客户端连接到服务器时识别服务器以及改善您的安全配置文件非常适用。例如，如果您的原始密钥是 RSA 密钥，则可以添加其他 ECDSA 密钥。

> **Note**
>
> SFTP 客户端使用其拥有的第一个公有密钥进行连接，该公有密钥可以与活动服务器密钥之一相匹配。

要添加其他的服务器主机密钥

1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。
2. 在左侧导航窗格中，选择服务器，然后选择使用 SFTP 协议的服务器。
3. 在服务器详细信息页面上，向下滚动到服务器主机密钥部分。

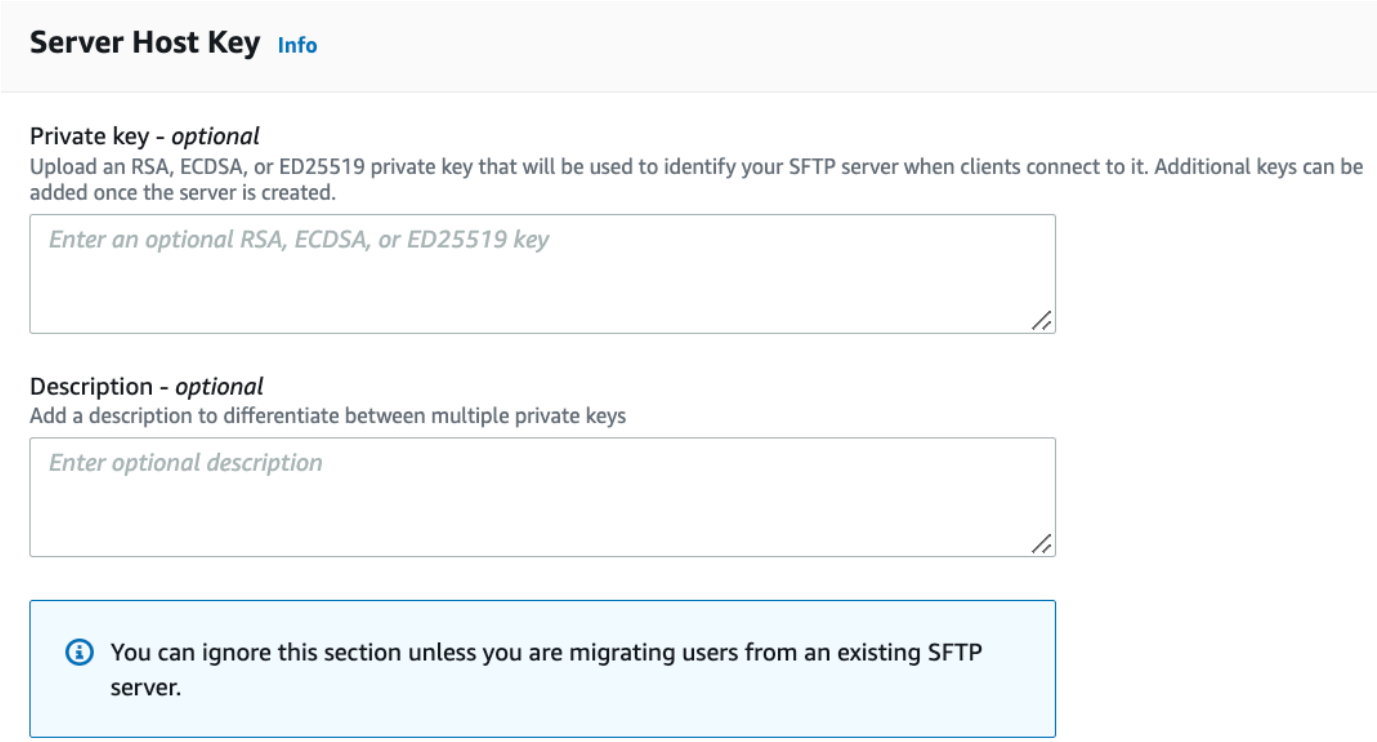


4. 选择添加主机密钥。

   将显示添加服务器主机密钥页面。

5. 在服务器主机密钥部分中，输入 RSA、ECDSA 或 ED25519 私有密钥，用于在客户端通过启用 SFTP 的服务器连接到服务器时标识您的服务器。

> **Note**
>
> 创建服务器主机密钥时，请务必指定`-N ""`（无密码）。有关如何生成密钥对的详细信息，请参阅在 macOS、Linux 或 Unix 系统创建 SSH 密钥。



6. （可选）添加描述以区分多个服务器主机密钥。您可以为密钥添加标签。
7. 选择 Add key (添加密钥)。您将返回到服务器详细信息页面。

要使用 AWS Command Line Interface (AWS CLI) 添加主机密钥，请使用 the section called “ImportHostKey” API 操作并提供新的主机密钥。如果创建新的启用 SFTP 的服务器，则在 the section called “CreateServer”API 操作中提供主机密钥作为参数。您也可以使用 AWS CLI 更新现有主机密钥的描述。

以下示例`import-host-key` AWS CLI 命令导入指定启用 SFTP 的服务器的主机密钥。

```
aws transfer import-host-key --description key-description --server-id your-server-id
 --host-key-body file://my-host-key
```

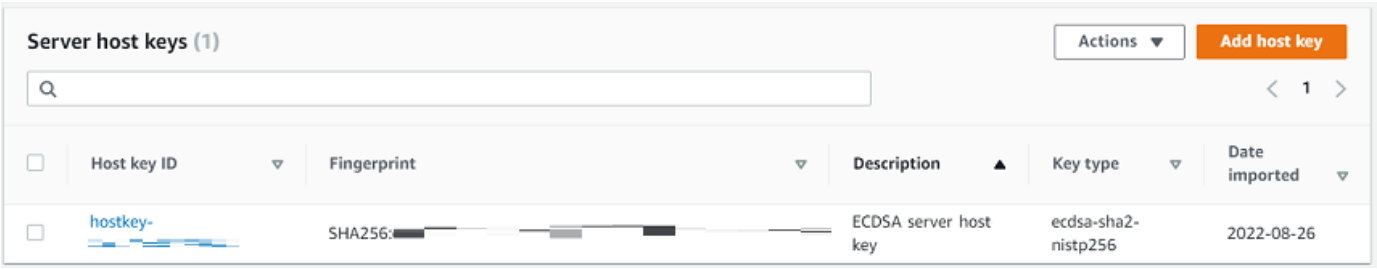
删除服务器主机密钥

在 AWS Transfer Family 控制台上，您可以删除服务器主机密钥。

要删除服务器主机密钥

1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。

2. 在左侧导航窗格中，选择服务器，然后选择使用 SFTP 协议的服务器。
3. 在服务器详细信息页面上，向下滚动到服务器主机密钥部分。



4. 在服务器主机密钥部分中，选择一个密钥，然后在操作下选择删除。
5. 在出现的确认对话框中，输入单词**delete**，然后选择删除以确认要删除主机密钥。

主机密钥已从服务器页面中删除。

要使用删除主机密钥 AWS CLI，请使用 the section called “DeleteHostKey” API 操作并提供服务器 ID 和主机密钥 ID。

以下示例`delete-host-key` AWS CLI 命令删除指定启用 SFTP 的服务器的主机密钥。

```
aws transfer delete-host-key --server-id your-server-id --host-key-id your-host-key-id
```

轮换服务器主机密钥

您可以定期轮换服务器主机密钥。

客户端如何选择服务器主机密钥

Transfer Family 选择应用哪个服务器密钥的方式取决于 SFTP 客户端的条件，如下所述。假设有一个较旧的密钥和一个较新的密钥。

- SFTP 客户端之前没有服务器的公用主机密钥。客户端首次连接到服务器时，会发生以下任一情况：
  - 如果配置为连接失败，则客户端会导致连接失败。
  - 或者，客户端选择与可能的可用算法相匹配的第一个密钥，并询问用户该密钥是否可信。如果是，则客户端会自动更新`known_hosts`文件（或客户端用来记录信任决策的任何本地配置文件或资源）并输入该密钥。
- SFTP 客户端`known_hosts`的文件中有一个较旧的密钥。即使存在较新的密钥，客户端也倾向于将此密钥用于此密钥的算法或其他算法。这是因为客户端对其`known_hosts`文件中的密钥具有更高的信任度。

- SFTP 客户端的密钥文件中包含新密known_hosts钥（采用任何可用的算法）。客户端会忽略旧密钥，因为它们不受信任，而是使用新密钥。
- SFTP 客户端known_hosts的文件中包含两个密钥。客户端通过索引选择与服务器提供的可用密钥列表相匹配的第一个密钥。

Transfer Family 更喜欢 SFTP 客户端在其known_hosts文件中包含所有密钥，因为这样在连接到 Transfer Family 服务器时可以获得最大的灵活性。密钥轮换基于这样一个事实，即同一个 Transfer Family 服务器known_hosts的文件中可能存在多个条目。

轮换服务器主机密钥程序

例如，假设您已将以下一组服务器主机密钥添加到 Transfer Family 服务器中。

服务器主机密钥

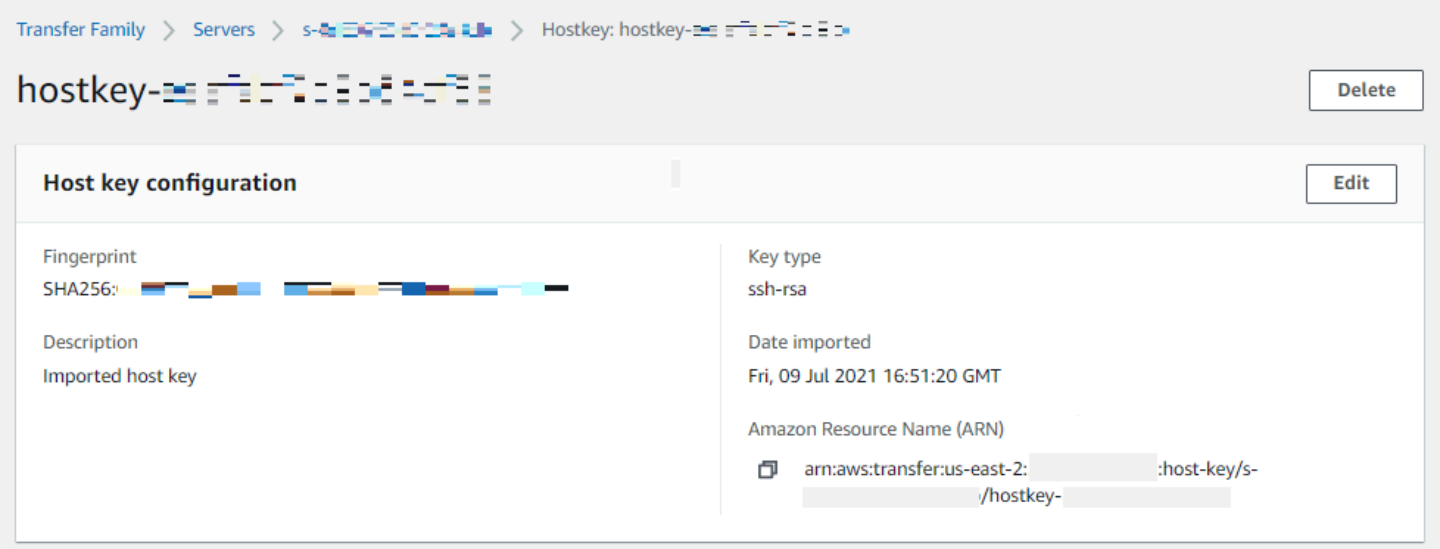
主机密钥类型	添加到服务器的日期
RSA	2020 年 4 月 1 日
ECDSA	2020 年 2 月 1 日
ED25519	2019 年 12 月 1 日
RSA	2019 年 10 月 1 日
ECDSA	2019 年 6 月 1 日
ED25519	2019 年 3 月 1 日

要轮换服务器主机密钥

1. 添加新的服务器主机密钥。有关此过程的说明，请参阅添加其他的服务器主机密钥。
2. 删除您之前添加的一个或多个相同类型的主机密钥。有关此过程的说明，请参阅删除服务器主机密钥。
3. 所有按键均可见，并且可以处于活动状态，具体取决于前面中描述的行为客户端如何选择服务器主机密钥。

其他服务器主机密钥信息

您可以选择主机密钥以显示该密钥的详细信息。



您可以删除主机密钥，也可以从服务器详细信息屏幕上的操作菜单中编辑其描述。选择主机密钥，然后从菜单中选择相应的操作。

在控制台中监控使用情况

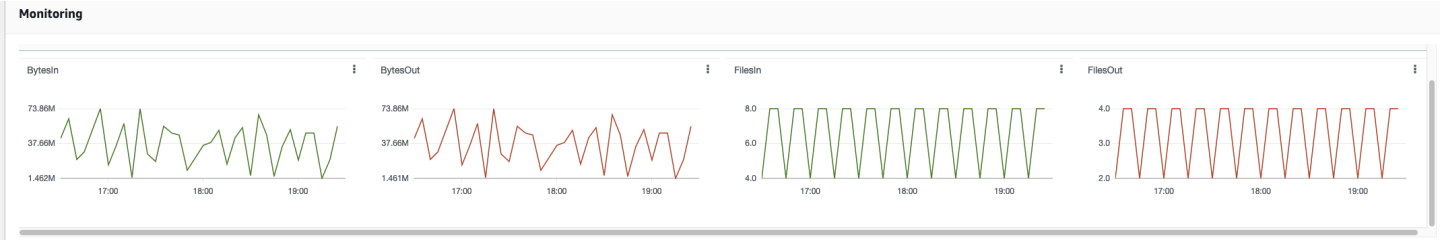
您可以在服务器详细信息页面上获取有关服务器指标的信息。这为您提供了一处位置以监控文件传输工作负载。您可以使用集中式控制面板跟踪与合作伙伴交换文件的数量，并密切跟踪其使用情况。有关更多信息，请参阅 查看 SFTP、FTPS 和 FTP 服务器的详细信息。下表描述了可用于 Transfer Family 的指标。

命名空间	指标	描述
AWS/Transfer	BytesIn	传输至服务器的字节总数。

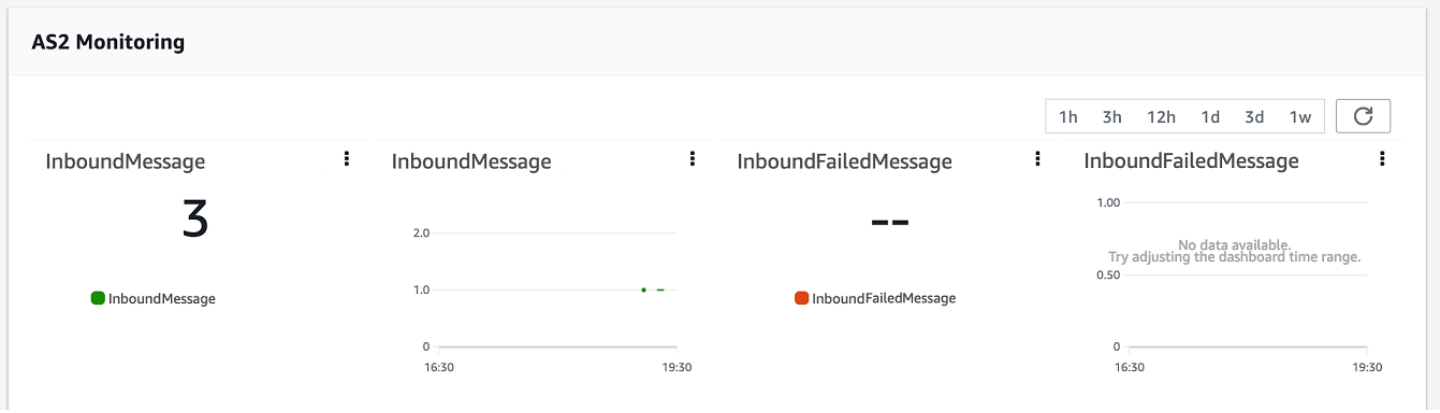
命名空间	指标	描述
		单位：计数 时间：5 分钟
	BytesOut	从服务器传出来的字节总数。 单位：计数 时间：5 分钟
	FilesIn	传输至服务器的字节总数。 对于使用 AS2 协议的服务器，此指标表示所收到的消息数量。 单位：计数 时间：5 分钟
	FilesOut	从服务器传出来的字节总数。 单位：计数 时间：5 分钟
	InboundMessage	成功从交易伙伴处收到的 AS2 消息总数。 单位：计数 时间：5 分钟
	InboundFailedMessage	未成功从交易伙伴处收到的 AS2 消息总数。也就是说，交易伙伴发送了一条消息，但是 Transfer Family 服务器无法成功处理该消息。 单位：计数 时间：5 分钟

命名空间	指标	描述
	OnPartialUploadExecutionsStarted	服务器上启动的工作 on-partial-upload 流程执行总数。 单位：计数 时间：1 分钟
	OnPartialUploadExecutionsSuccess	服务器上成功执行 on-partial-upload 的工作流程总数。 单位：计数 时间：1 分钟
	OnPartialUploadExecutionsFailed	服务器上执行失败 on-partial-upload 的工作流程总数。 单位：计数 时间：1 分钟
	OnUploadExecutionsStarted	服务器上启动的工作流程执行总数。 单位：计数 时间：1 分钟
	OnUploadExecutionsSuccess	服务器上执行成功的工作流程总数。 单位：计数 时间：1 分钟
	OnUploadExecutionsFailed	服务器上执行失败的工作流程总数。 单位：计数 时间：1 分钟

监控部分包含四个单独的图表。这些图表显示了字节输入、字节输出、文件输入和文件输出。



对于启用了 AS2 协议的服务器，监控信息下方有一个 AS2 监控部分。本节包含成功和失败的入站消息数的详细信息。



要在自己的窗口中打开所选图表，请选择展开图标
()。
您也可以单击图表的垂直省略号图标
()
以打开包含以下项目的下拉菜单：

- 放大 — 在自己的窗口中打开所选图表。
- 刷新 — 使用最新数据重新加载图表。
- 在指标中查看-在 Amazon 中打开相应的指标详情 CloudWatch。
- 查看日志-在中打开相应的日志组 CloudWatch。

管理访问控制

您可以使用 AWS Identity and Access Management (IAM) 策略控制用户对 AWS Transfer Family 资源的访问权限。 策略是一个语句（通常采用 JSON 格式），它允许对资源进行特定级别的访问。您可以使用 策略来定义希望允许 SFTP 用户执行和不执行哪些文件操作。您还可以使用 策略来定义希望允许用户访问哪些 存储桶。要为用户指定这些策略，您可以为 创建一个 角色，该角色具有与之关联的 策略和信任关系。

为每个 SFTP 用户分配一个 角色。 AWS Transfer Family 使用的 IAM 角色类型称为服务角色。当用户登录到您的服务器时， AWS Transfer Family 将使用映射到该用户的 IAM 角色。要了解如何创建向用户提供对 Amazon S3 存储桶的访问权限的 IAM 角色，请参阅 IAM 用户指南中的创建向 AWS 服务委派权限的角色。

您可以使用 IAM 策略中的特定权限授予对 Amazon S3 对象的只写访问权限。有关更多信息，请参阅授予仅写入和列出文件的权限。

AWS 存储博客包含一篇详细介绍如何设置最低权限访问权限的文章。有关详细信息，请参阅在AWS Transfer Family 工作流程中实现最低权限访问权限。

> **Note**
>
> 如果您的 Amazon S3 存储桶使用 AWS Key Management Service (AWS KMS) 进行加密，则必须在策略中指定其他权限。有关更多信息，请参阅 Amazon S3 中的数据加密。此外，您可以在 IAM 用户指南中查看有关 会话策略的更多信息。

主题

- 允许对 存储桶的读取和写入访问权限
- 为 Amazon S3 存储桶创建会话策略
- 阻止用户在 S3 存储桶中运行mkdir

允许对 存储桶的读取和写入访问权限

此部分说明了如何创建 IAM 策略，以允许对特定 Amazon S3 存储桶进行读写访问。通过将具有此 策略的 角色分配给 SFTP 用户，将允许该用户对指定的 S3 存储桶进行读/写访问。

以下策略允许通过编程方式对 存储桶进行读写访问。只有当您需要启用跨账户存取时，才需要GetObjectACL和PutObjectACL语句。也就是说，您的 Transfer Family 服务器需要访问其他账户中的存储桶。

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid":"ReadWriteS3",
      "Action": [
            "s3:ListBucket"
                ],
      "Effect": "Allow",
      "Resource": ["arn:aws:s3:::DOC-EXAMPLE-BUCKET"]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:GetObjectTagging",
        "s3:DeleteObject",
        "s3:DeleteObjectVersion",
        "s3:GetObjectVersion",
        "s3:GetObjectVersionTagging",
        "s3:GetObjectACL",
        "s3:PutObjectACL"
      ],
      "Resource": ["arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"]
    }
  ]
}
```

ListBucket 操作需要对存储桶本身的权限。PUT、GET 和 DELETE 操作需要对象权限。由于这些是不同的实体，因此使用不同的 Amazon 资源名称 (ARN) 指定它们。

要进一步缩小用户的访问权限以使其仅能访问指定的 S3 存储桶的 home 目录，请参阅为 Amazon S3 存储桶创建会话策略。

为 Amazon S3 存储桶创建会话策略

会话策略是一项 AWS Identity and Access Management (IAM) 策略，它限制用户访问 Amazon S3 存储桶的某些部分。它通过实时评估访问来做到这一点。

> **Note**
>
> 会话策略仅适用于 Amazon S3。对于 Amazon EFS，您可以使用 POSIX 文件权限限制访问权限。

当您需要向一组用户授予对 S3 存储桶的特定部分的相同访问权限时，可以使用范围缩小策略。例如，一组用户可能仅需访问 home 目录。该组用户共享相同的 角色。

> **Note**
>
> 路径的长度上限是 4000 个字符。有关更多详细信息，请参阅 API 参考中`CreateUser`操作的策略请求参数。

要创建范围缩小策略，请在 IAM 策略中使用以下策略变量：

- `${transfer:HomeBucket}`
- `${transfer:HomeDirectory}`
- `${transfer:HomeFolder}`
- `${transfer:UserName}`

> **Important**
>
> 您不能在托管策略中使用前述变量。也不能在 IAM 角色定义中将其用作策略变量。您可以在策略中创建这些变量，并在设置用户时直接提供这些变量。另外，您不能在此范围缩小策略中使用 `${aws:Username}` 变量。此变量引用了 用户名而不是 所需的用户名。

以下代码所示为会话策略示例。

```
{
```

```
  "Version": "2012-10-17",
  "Statement": [
      {
          "Sid": "AllowListingOfUserFolder",
          "Action": [
              "s3:ListBucket"
          ],
          "Effect": "Allow",
          "Resource": [
              "arn:aws:s3:::${transfer:HomeBucket}"
          ],
          "Condition": {
              "StringLike": {
                  "s3:prefix": [
                      "${transfer:HomeFolder}/*",
                      "${transfer:HomeFolder}"
                  ]
              }
          }
      },
      {
          "Sid": "HomeDirObjectAccess",
          "Effect": "Allow",
          "Action": [
              "s3:PutObject",
              "s3:GetObject",
              "s3:DeleteObjectVersion",
              "s3:DeleteObject",
              "s3:GetObjectVersion",
              "s3:GetObjectACL",
              "s3:PutObjectACL"
          ],
          "Resource": "arn:aws:s3:::${transfer:HomeDirectory}/*"
       }
  ]
}
```

> **Note**
>
> 前面的策略示例假设用户的主目录设置为包含尾部斜杠，以表示它是一个目录。另一方面，如果您设置的用户HomeDirectory不带尾部的斜杠，则应将其作为策略的一部分。

在前面的示例策略中，请注意使用transfer:HomeFolder、transfer:HomeBucket和transfer:HomeDirectory策略参数。这些参数是为用户配置的设置的，如HomeDirectory和中所述实施您的 API Gateway 方法。HomeDirectory这些参数具有以下定义：

- transfer:HomeBucket参数将替换为的HomeDirectory第一个组件。
- transfer:HomeFolder参数将替换为HomeDirectory参数的其余部分。
- transfer:HomeDirectory参数删除了前导正斜杠 (/)，因此可以在Resource语句中将其用作 S3 Amazon 资源名称（ARN）的一部分。

> **Note**
>
> 如果您使用的是逻辑目录（即用户的homeDirectoryType是LOGICAL），则不支持这些策略参数（HomeBucket、HomeDirectory和HomeFolder）。

例如，假设为 Transfer Family 用户配置的HomeDirectory参数是/home/bob/amazon/stuff/。

- transfer:HomeBucket 设置为 /home。
- transfer:HomeFolder 设置为 /bob/amazon/stuff/。
- transfer:HomeDirectory 变为 home/bob/amazon/stuff/。

第一个"Sid"允许用户列出从/home/bob/amazon/stuff/开始的所有目录。

第二个"Sid"限制用户对同一路径/home/bob/amazon/stuff/的put和get访问权限。

借助上述策略，当用户登录时，他们只能访问其主目录中的对象。在连接时， AWS Transfer Family 将这些变量替换为适合用户的值。这样做可以更轻松地将相同的策略文档应用于多个用户。此方法减少了用于管理用户对 存储桶的访问的 角色和策略管理的开销。

您还可以使用范围缩小策略以根据业务需求自定义每个用户的访问权限。有关更多信息，请参阅 IAM 用户指南 AssumeRoleWithWebIdentity中的权限 AssumeRole、 AssumeRoleWith SAM L 和。

> **Note**
>
> AWS Transfer Family 存储策略 JSON，而不是策略的亚马逊资源名称 (ARN)。因此，当您在 IAM 控制台中对策略进行更改时，需要返回到 AWS Transfer Family 控制台并使用最新策略内容更新您的用户。您可以在 用户配置 部分的 策略信息 选项卡上更新用户。

如果您使用的是 AWS CLI，则可以使用以下命令来更新策略。

```
aws transfer update-user --server-id server --user-name user --policy \
   "$(aws iam get-policy-version --policy-arn policy --version-id version --
output json)"
```

阻止用户在 S3 存储桶中运行mkdir

您可限制用户在 Amazon S3 存储桶中创建目录。为此，您应创建 IAM 策略，允许 s3:PutObject 操作，但在密钥以“/”（反斜杠）结尾时拒绝该操作。以下示例策略允许用户将文件上传至 Amazon S3 存储桶，但拒绝在 Amazon S3 存储桶中执行mkdir命令。

```
{
   "Sid":"DenyMkdir",
   "Action":[
      "s3:PutObject"
   ],
   "Effect":"Deny",
   "Resource":[
      "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*/",
      "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*/*"
    ]
}
```

Note

用户可通过第二行资源，运行put my-file *DOC-EXAMPLE-BUCKET*/new-folder/my-file等命令创建子文件夹。

AWS Transfer Family 的日志记录

AWS Transfer Family与两者AWS CloudTrail和 Amazon 集成 CloudWatch。 CloudTrail 并 CloudWatch 用于不同但互补的目的：

- CloudTrail 是一项AWS服务，用于记录您内部所执行的操作AWS 账户。它会持续监控和记录控制台登录、AWS Command Line Interface命令和 SDK/API 调用等活动的 API 调用。这使您可以记录谁在何时何地采取了什么行动。 CloudTrail 通过提供AWS环境中所有活动的历史记录，帮助审计、访问管理和监管合规性。有关详细信息，请参阅《AWS CloudTrail用户指南》。
- CloudWatch 是AWS资源和应用程序的监控服务。它收集指标和日志，以提供对资源利用率、应用程序性能和整体系统运行状况的可见性。 CloudWatch 帮助完成操作任务，例如故障排除、设置警报和自动缩放。有关详情，请参阅 Amazon CloudWatch 用户指南。

主题

AWS CloudTrail正在登录 AWS Transfer Family

AWS Transfer Family与AWS CloudTrail一项服务集成，该服务提供用户、角色或AWS服务在中执行的操作的记录AWS Transfer Family。 CloudTrail 将所有 API 调用捕获AWS Transfer Family为事件。捕获的调用包含来自 AWS Transfer Family 控制台和代码的 AWS Transfer Family API 操作调用。

跟踪是一种配置，允许将事件作为日志文件传输到您指定的 Amazon S3 存储桶。 CloudTrail 日志文件包含一个或多个日志条目。事件代表来自任何来源的单个请求，包括有关请求的操作、操作的日期和时间、请求参数等的信息。 CloudTrail 日志文件不是公共 API 调用的有序堆栈跟踪，因此它们不会按任何特定的顺序出现。

要持续记录 AWS 账户中的事件（包括 AWS Transfer Family 的事件），请创建跟踪。跟踪允许 CloudTrail 将日志文件传输到 Amazon S3 存储桶。预设情况下，在控制台中创建跟踪时，此跟踪应用于所有 AWS 区域。此跟踪记录在 AWS 分区中记录所有区域中的事件，并将日志文件传送至您指定的 Simple Storage Service（Amazon S3）存储桶。此外，您可以配置其他AWS服务，以进一步分析和处理 CloudTrail 日志中收集的事件数据。有关更多信息，请参阅下列内容：

- Overview for creating a trail
- CloudTrail 支持的服务和集成

- 配置 Amazon SNS 通知 CloudTrail
- 接收来自多个地区的 CloudTrail 日志文件和接收来自多个账户的 CloudTrail 日志文件

所有AWS Transfer Family操作均由记录 CloudTrail 并记录在ActionsAPI reference。例如，调用ListUsers和StopServer操作会在 CloudTrail 日志文件中生成条目。CreateServer

每个事件或日记账条目都包含有关生成请求的人员信息。身份信息有助于您确定以下内容：

- 请求是使用根用户凭证还是 AWS Identity and Access Management 用户凭证发出的。
- 请求是使用角色还是联合用户的临时安全凭证发出的。
- 请求是否由其它 AWS 服务发出。

有关更多信息，请参阅 CloudTrail userIdentity 元素。

如果您创建了跟踪，则可以允许将 CloudTrail 事件持续传输到 Amazon S3 存储桶，包括的事件AWS Transfer Family。如果您未配置跟踪，您仍然可以在 CloudTrail 控制台的 “事件历史记录” 中查看最新的事件。

使用收集的信息 CloudTrail，您可以确定向哪个请求发出AWS Transfer Family、发出请求的 IP 地址、谁发出了请求、何时发出请求以及其他详细信息。

要了解更多信息 CloudTrail，请参阅AWS CloudTrail用户指南。

主题

- 启用日志记录
- 创建服务器的日志条目示例

启用日志记录

您可使用 AWS CloudTrail 监控 AWS Transfer Family API 调用。通过监控 API 调用，您可以获取有用的安全性和操作信息。如果您启用了 Amazon S3 对象级日志记录RoleSessionName，则以[AWS:Role Unique Identifier]/username.sessionid@server-id形式包含在“请求者”字段。有关唯一标识符的更多信息，请参阅《IAM 用户指南》中的 IAM 标识符。

> **Important**
>
> 名称的长度上限是 255 个字符。如果RoleSessionName较长，则server-id会被截断。

创建服务器的日志条目示例

以下示例显示了演示CreateServer操作的 CloudTrail 日志条目（JSON 格式）。


```
{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AAAA4FFF5HHHHH6NNWWWW:user1",
        "arn": "arn:aws:sts::123456789102:assumed-role/Admin/user1",
        "accountId": "123456789102",
        "accessKeyId": "AAAA52C2WWWWWW3BB4Z",
        "sessionContext": {
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2018-12-18T20:03:57Z"
            },
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AAAA4FFF5HHHHH6NNWWWW",
                "arn": "arn:aws:iam::123456789102:role/Admin",
                "accountId": "123456789102",
                "userName": "Admin"
            }
        }
    },
    "eventTime": "2024-02-05T19:18:53Z",
    "eventSource": "transfer.amazonaws.com",
    "eventName": "CreateServer",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "11.22.1.2",
    "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
 like Gecko) Chrome/121.0.0.0 Safari/537.36",
    "requestParameters": {
        "domain": "S3",
        "hostKey": "HIDDEN_DUE_TO_SECURITY_REASONS",
        "protocols": [
            "SFTP"
        ],
        "protocolDetails": {
            "passiveIp": "AUTO",
            "tlsSessionResumptionMode": "ENFORCED",
            "setStatOption": "DEFAULT"
```

```
        },
        "securityPolicyName": "TransferSecurityPolicy-2020-06",
        "s3StorageOptions": {
            "directoryListingOptimization": "ENABLED"
        }
    },
    "responseElements": {
        "serverId": "s-1234abcd5678efghi"
    },
    "requestID": "6fe7e9b1-72fc-45b0-a7f9-5840268aeadf",
    "eventID": "4781364f-7c1e-464e-9598-52d06aa9e63a",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789102",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.3",
        "cipherSuite": "TLS_AES_128_GCM_SHA256",
        "clientProvidedHostHeader": "transfer.us-east-1.amazonaws.com"
    },
    "sessionCredentialFromConsole": "true"
}
```


Amazon CloudWatch 正在登录 AWS Transfer Family

Amazon 会实时 CloudWatch 监控您的 AWS Transfer Family 资源和您运行 AWS 的应用程序。您可以使用 CloudWatch 来收集和跟踪指标，这些指标是您可以衡量资源和应用程序的变量。

CloudWatch 主页会自动显示有关Transfer Family和您使用的所有其他 AWS 服务的指标。此外，您还可以创建自定义控制面板，以显示有关自定义应用程序的指标，并显示您选择的指标的自定义集合。

您可以创建警报，这些警报监视指标，当超出阈值时，它们会发送通知或者对您所监控的资源自动进行更改。例如，您可以监控传输到 Transfer Family 服务器的文件，并使用该数据来确定是否需要部署其他服务器来处理增加的负载。您还可以使用这些数据来停止或删除未充分使用的实例以节省资金。

主题

- 创建、更新和查看服务器的日志记录
- 管理工作流程的日志记录
- 配置 CloudWatch 日志记录角色

- 查看 Transfer Family 日志流
- 创建亚马逊 CloudWatch 警报
- Amazon S3 API 调用 S3 访问日志的记录
- 混淆代理问题限制示例
- CloudWatch Transfer Family 的日志结构
- CloudWatch 日志条目示例
- 使用 T CloudWatch ransfer Family 的指标
- AWS 用户通知服务 与一起使用 AWS Transfer Family

创建、更新和查看服务器的日志记录

对于所有 AWS Transfer Family 服务器，您可以在两个日志记录选项之间进行选择：`LoggingRole`（用于记录连接到服务器的工作流程）或`StructuredLogDestinations`。使用 `StructuredLogDestinations` 具有以下好处：

- 接收结构化 JSON 格式的日志。
- 使用 Amazon Logs Insights 查询您的 CloudWatch 日志，它会自动发现 JSON 格式的字段。
- 跨 AWS Transfer Family 资源共享日志组允许您将来自多个服务器的日志流合并到一个日志组中，从而更轻松地管理监控配置和日志保留设置。
- 创建可添加到 CloudWatch仪表板的聚合指标和可视化效果。
- 使用日志组创建整合的日志指标、可视化效果和控制面板，从而跟踪使用情况和性能数据。

`LoggingRole`或的选项`StructuredLogDestinations`是单独配置和控制的。对于每台服务器，您可以设置一种或两种日志记录方法，或者将服务器配置为不进行任何日志记录（尽管不建议采取这种方法）。

如果使用 Transfer Family 控制台创建新的服务器，将默认启用日志记录。创建服务器后，您可以使用`UpdateServer` API 调用来更改日志记录配置。有关更多信息，请参阅 StructuredLogDestinations。

目前，对于工作流程，如果要启用日志记录，则必须指定日志记录角色：

- 如果您使用`CreateServer`或 `UpdateServer` API 调用将工作流程与服务器关联，则系统不会自动创建日志记录角色。如果要记录工作流程事件，则需要将日志记录角色显式附加到服务器。

- 如果您使用 Transfer Family 控制台创建服务器并附加工作流程，则日志将发送到名称中包含服务器 ID 的日志组。例如/aws/transfer/s-1111aaaa2222bbbb3，格式为/aws/transfer/*server-id*。服务器日志可以发送到同一个日志组或另一个日志组。

在控制台中创建和编辑服务器的日志记录注意事项

- 除非将工作流程附加到服务器，否则通过控制台创建的新服务器仅支持结构化 JSON 日志记录。
- 无日志记录不是您在控制台中创建的新服务器的选项。
- 现有服务器可以随时通过控制台启用结构化 JSON 日志记录。
- 通过控制台启用结构化 JSON 日志记录会禁用现有的日志记录方法，以免向客户重复收费。如果将工作流程附加到服务器，则例外。
- 如果启用结构化 JSON 日志记录，则以后无法通过控制台将其禁用。
- 如果启用结构化 JSON 日志记录，则可以随时通过控制台更改日志组目标。
- 如果启用结构化 JSON 日志记录，则如果通过 API 启用了两种日志记录类型，则无法通过控制台编辑日志记录角色。如果服务器已附加工作流程，则例外。但是，日志记录角色会继续出现在其他详细信息中。

使用 API 或 SDK 创建和编辑服务器的日志记录注意事项

- 如果您通过 API 创建新服务器，则可以配置其中一种或两种类型的日志记录，或者选择不记录日志。
- 对于现有服务器，可以随时启用和禁用结构化 JSON 日志记录。
- 您可以随时通过 API 更改日志组。
- 您可以随时通过 API 更改日志记录角色。

若要启用结构化日志记录，您必须登录到具有以下权限的账户

- `logs:CreateLogDelivery`
- `logs:DeleteLogDelivery`
- `logs:DescribeLogGroups`
- `logs:DescribeResourcePolicies`
- `logs:GetLogDelivery`
- `logs:ListLogDeliveries`

- `logs:PutResourcePolicy`
- `logs:UpdateLogDelivery`

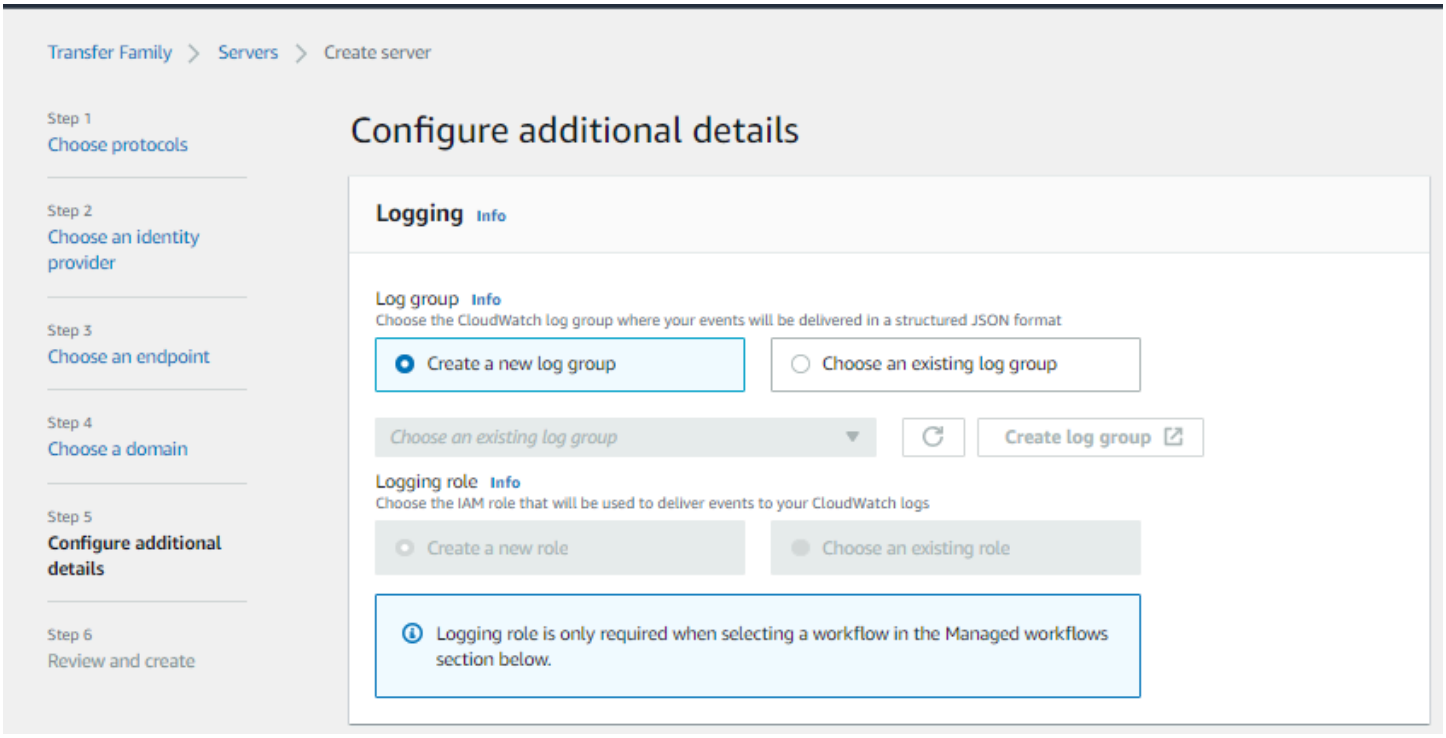
该部分提供了策略示例配置 CloudWatch 日志记录角色。

主题

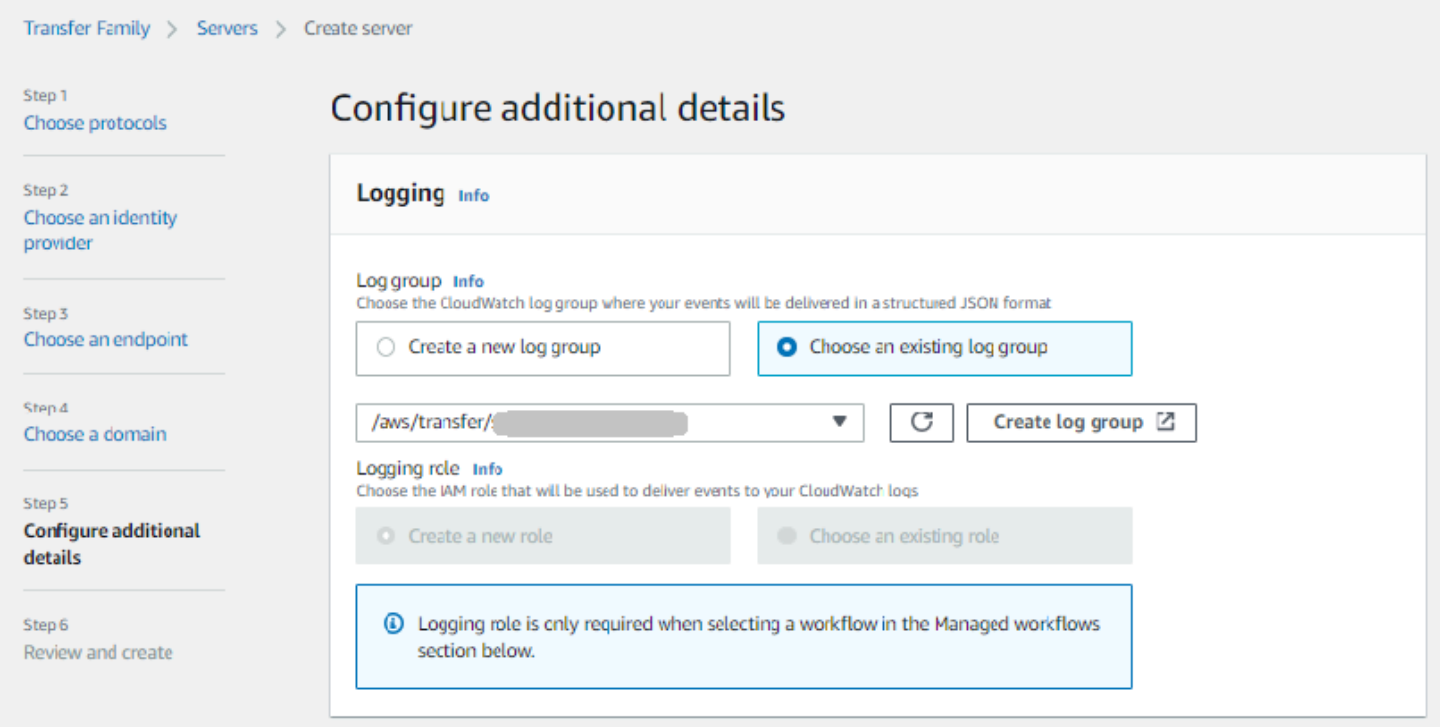
- 为服务器创建日志
- 更新服务器的日志记录
- 查看服务器配置

为服务器创建日志

创建新服务器时，可以在配置其他详细信息页面上指定现有日志组或创建新日志组。



如果您选择现有日志组，则必须选择与您的日志组关联的日志组 AWS 账户。



如果选择 “创建日志组”，则 CloudWatch 控制台 (https://console.aws.amazon.com/cloudwatch/) 将打开 “创建日志组” 页面。有关详细信息，请参阅在日志中创建 CloudWatch 日志组。

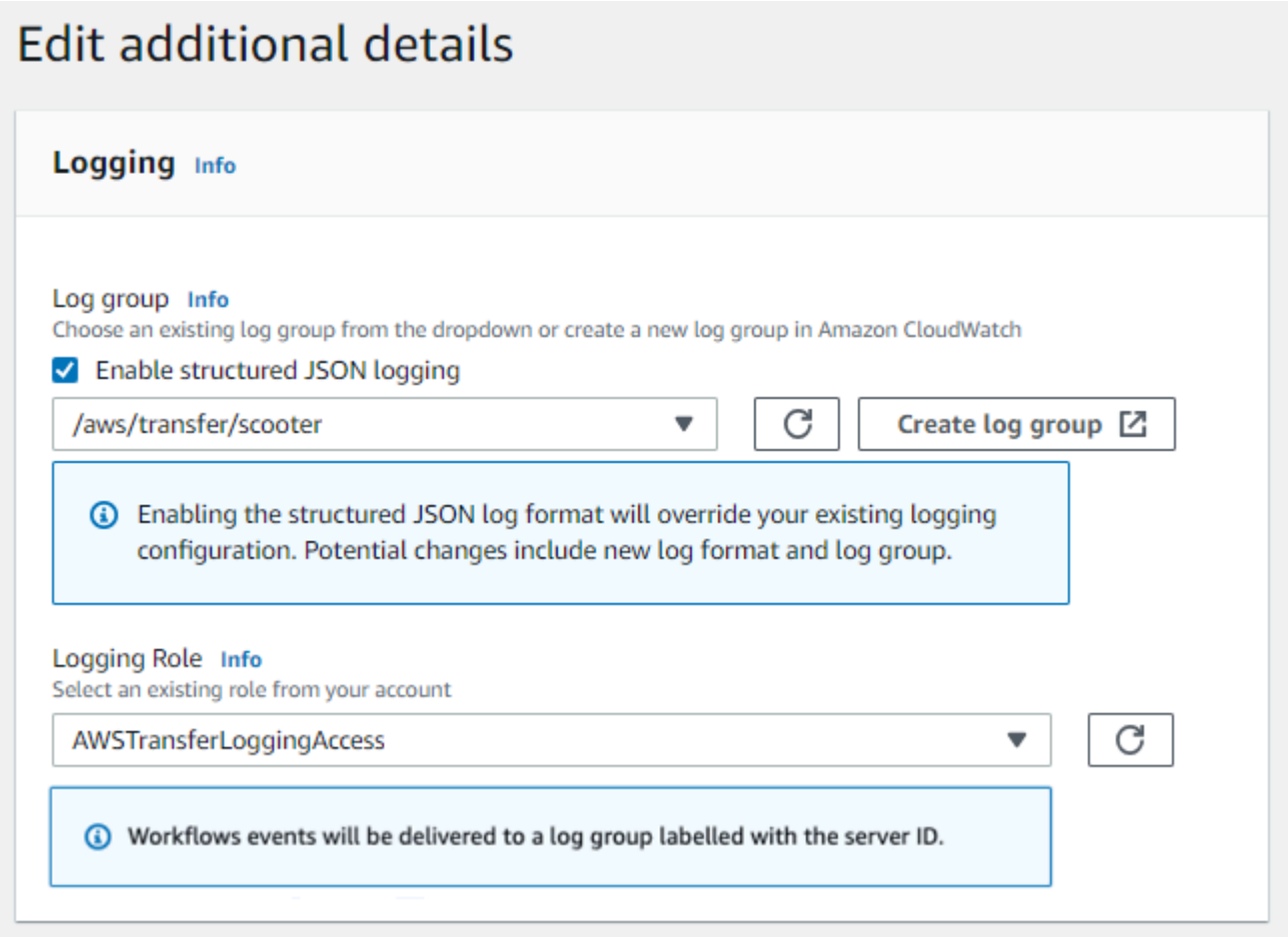
更新服务器的日志记录

日志记录的详细信息取决于您的更新场景。

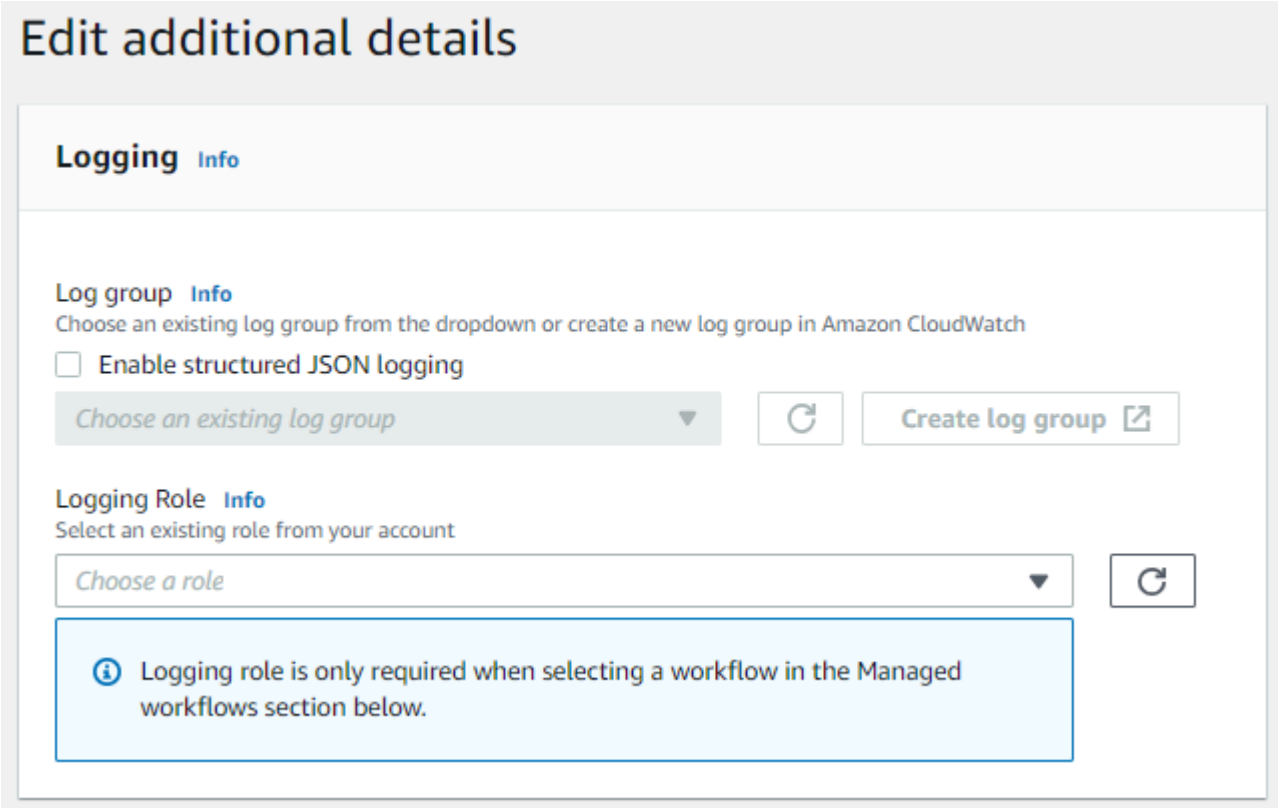
> **Note**
>
> 当您选择使用结构化 JSON 日志记录时，在极少数情况下，Transfer Family 会停止使用旧格式进行日志记录，但需要一些时间才能开始使用新的 JSON 格式进行日志记录。这可能会导致事件不被记录。不会出现任何服务中断，但是在更改日志记录方法后的第一个小时内，您应该谨慎传输文件，因为日志可能会被丢弃。

如果您正在编辑现有服务器，则选项取决于服务器的状态。

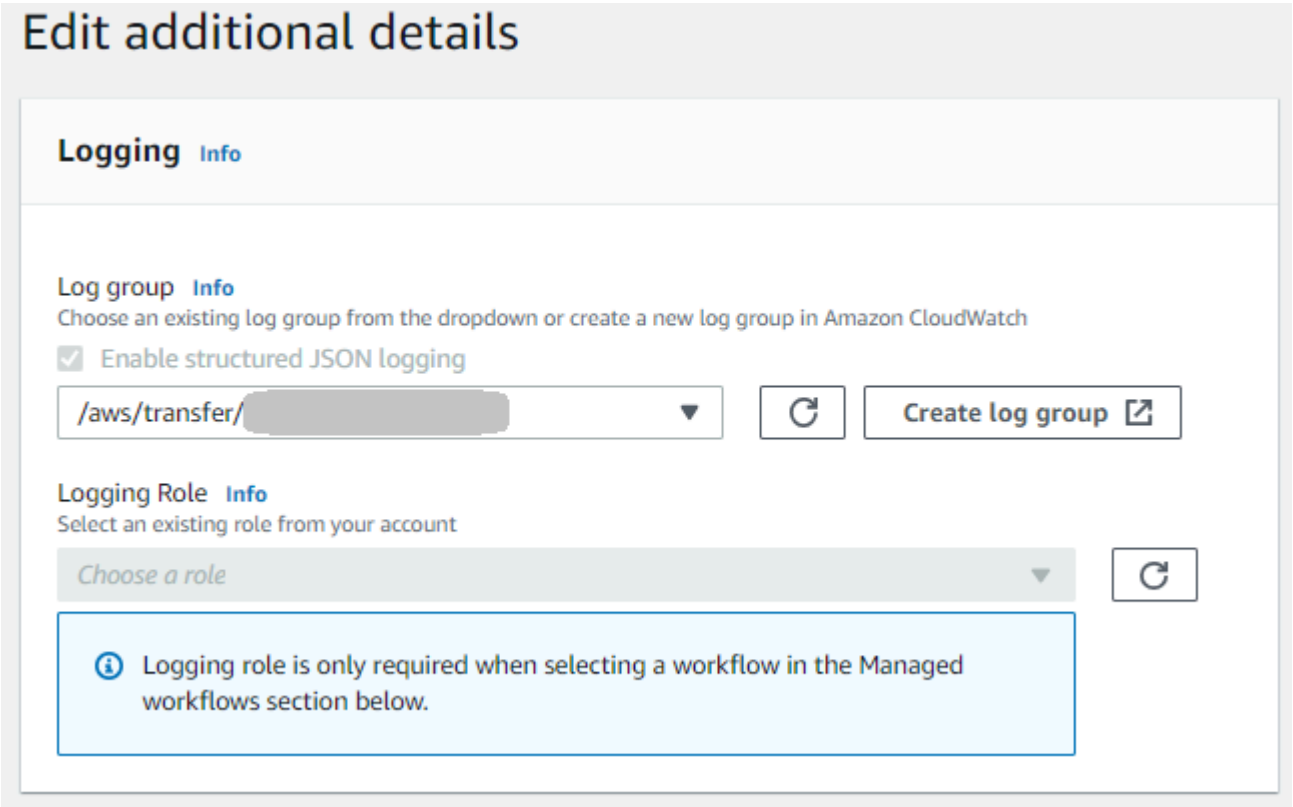
- 服务器已启用日志记录角色，但未启用结构化 JSON 日志记录。



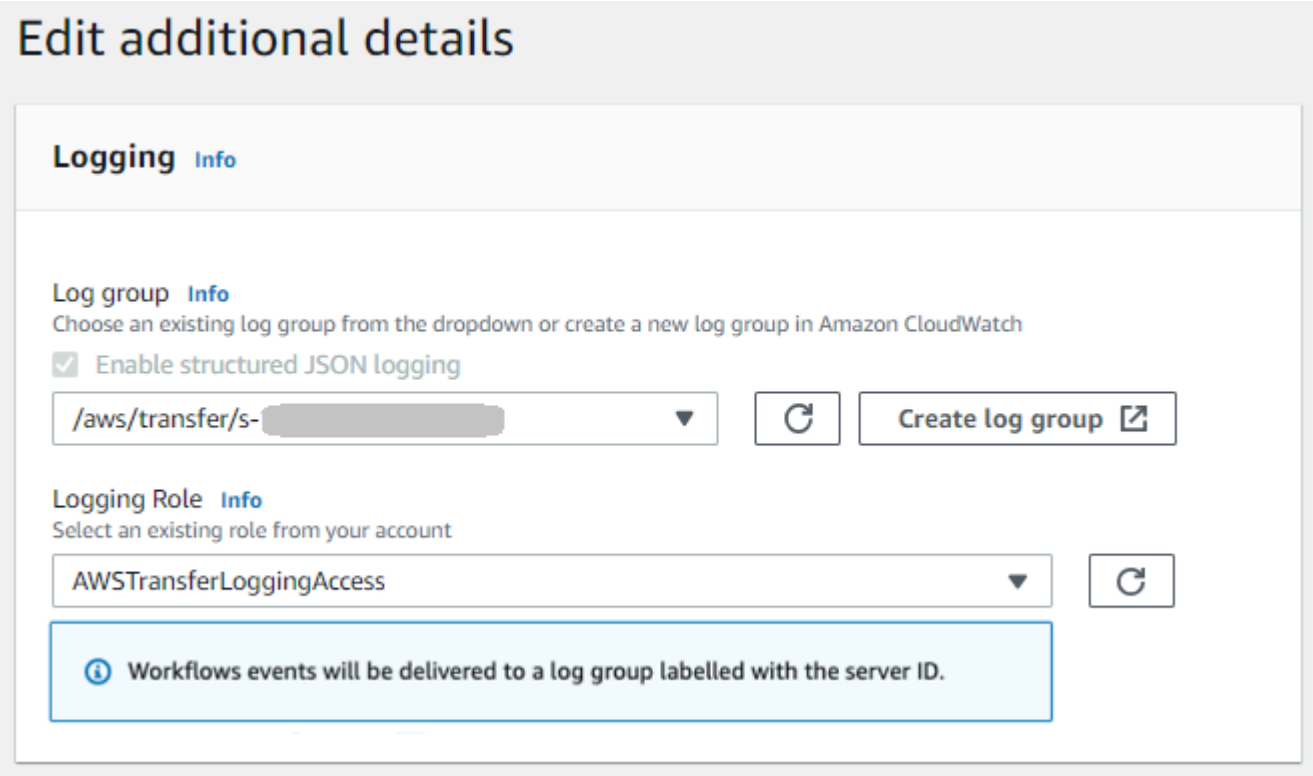
- 服务器未启用任何日志记录。



- 服务器已启用结构化 JSON 日志记录，但未指定日志记录角色。



- 服务器已启用结构化 JSON 日志记录，且已指定日志记录角色。

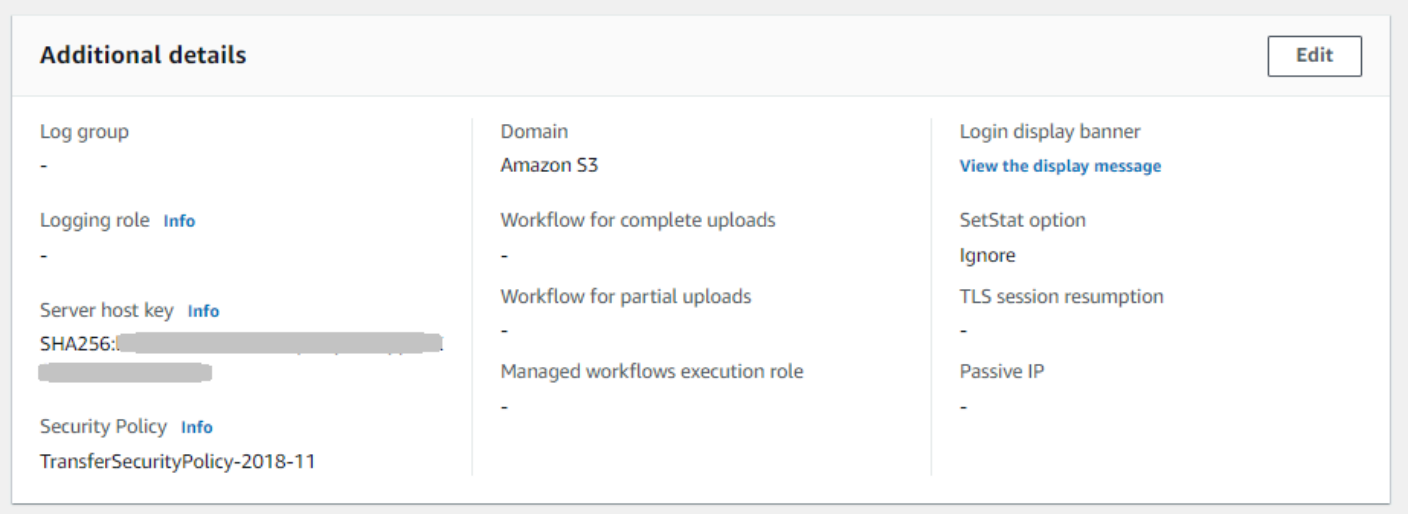


查看服务器配置

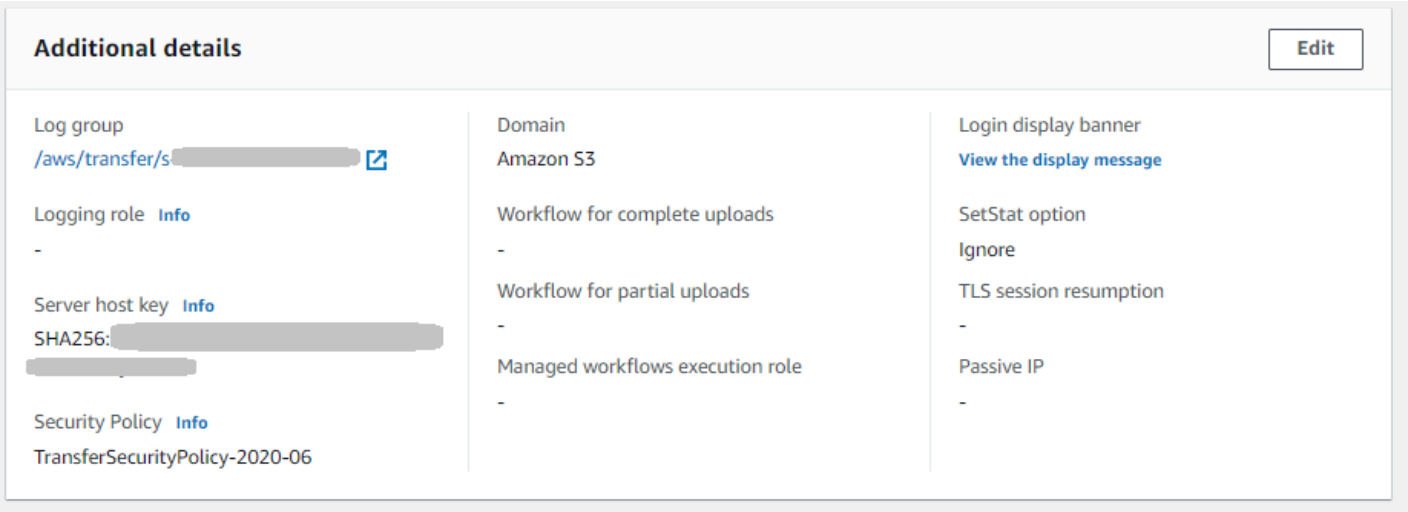
服务器配置页面的详细信息取决于您的场景：

根据您的场景，服务器配置页面可能类似于以下示例之一：

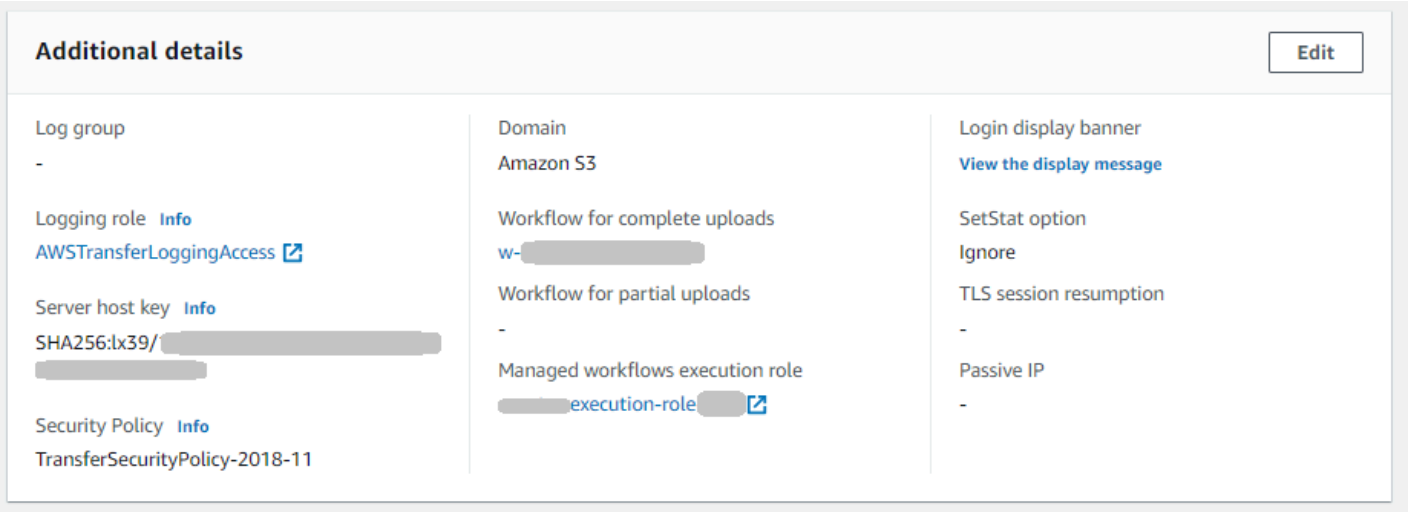
- 无日志记录已启用。



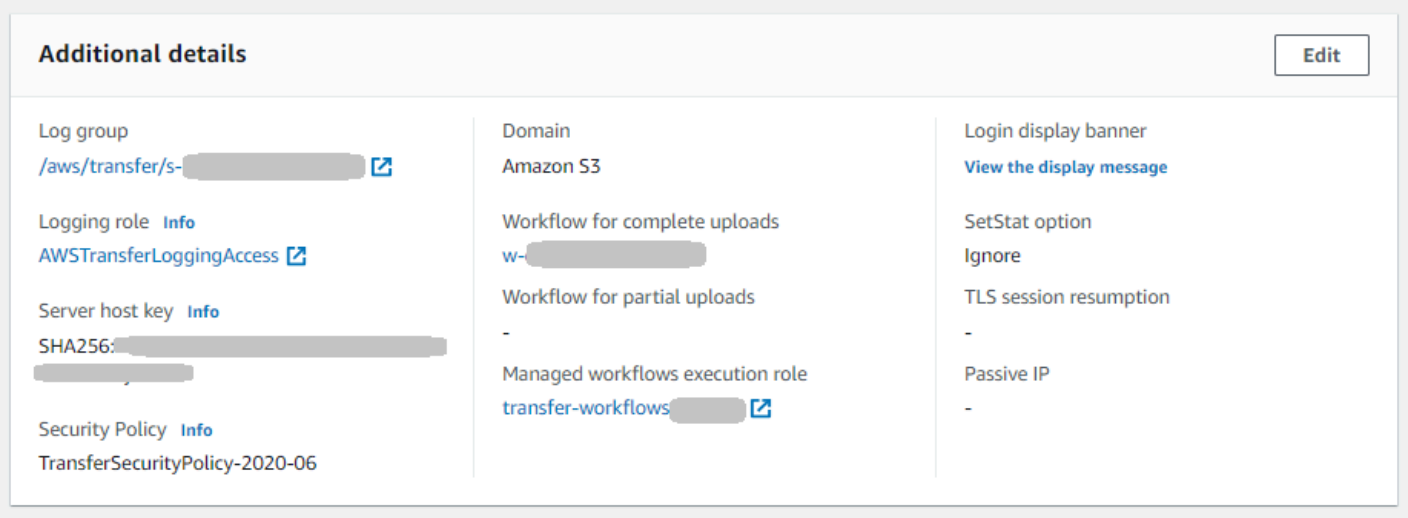
- 已启用结构化 JSON 日志记录。



- 日志记录角色已启用，但未启用结构化 JSON 日志记录。



- 两种类型的日志记录（日志记录角色和结构化 JSON 日志记录）均已启用。



管理工作流程的日志记录

CloudWatch 为工作流程进度和结果提供统一的审计和日志记录。此外，还为工作流程 AWS Transfer Family 提供了多个指标。您可以查看前一分钟有多少工作流程执行启动、成功完成和失败的指标。中描述了 Transfer Family 的所有 CloudWatch 指标使用 T CloudWatch ransfer Family 的指标。

查看 Amazon 工作流程 CloudWatch 日志

1. 打开亚马逊 CloudWatch 控制台，网址为 https://console.aws.amazon.com/cloudwatch/。
2. 在左侧导航窗格中选择日志，然后选择日志组。
3. 在日志组页面的导航栏上，为您的 AWS Transfer Family 服务器选择正确的区域。
4. 选择与您的服务器相对应的日志组。

   例如，如果您的服务器 ID 是 `s-1234567890abcdef0`，则您的日志组是 `/aws/transfer/s-1234567890abcdef0`。
5. 在服务器的日志组详细信息页面上，将显示最新的日志流。您正在探索的用户有两个日志流：

   - 每个 Secure Shell (SSH) 文件传输协议 (SFTP) 会话一个。
   - 一个用于正在为您的服务器执行的工作流程。工作流程的日志流格式为 *`username`*`.`*`workflowID`*`.`*`uniqueStreamSuffix`*。

   例如，如果您的用户是 `mary-major`，您具有以下日志流：

   ```
   mary-major-east.1234567890abcdef0
   ```

```
mary.w-abcdef01234567890.021345abcdef6789
```

> **Note**
>
> 此示例中列出的 16 位字母数字标识符是虚构的。您在 Amazon 上看到 CloudWatch 的值不同。

mary-major-usa-east.1234567890abcdef0 的“日志事件”页面显示每个用户会话的详细信息，mary.w-abcdef01234567890.021345abcdef6789 日志流包含工作流程的详细信息。

以下是基于包含复制步骤的工作流程 (w-abcdef01234567890) 的 mary.w-abcdef01234567890.021345abcdef6789 日志流示例。

```
{
    "type": "ExecutionStarted",
    "details": {
        "input": {
            "initialFileLocation": {
                "bucket": "DOC-EXAMPLE-BUCKET",
                "key": "mary/workflowSteps2.json",
                "versionId": "version-id",
                "etag": "etag-id"
            }
        }
    },
    "workflowId":"w-abcdef01234567890",
    "executionId":"execution-id",
    "transferDetails": {
        "serverId":"s-server-id",
        "username":"mary",
        "sessionId":"session-id"
    }
},
{
    "type":"StepStarted",
    "details": {
        "input": {
            "fileLocation": {
                "backingStore":"S3",
                "bucket":"DOC-EXAMPLE-BUCKET",
                "key":"mary/workflowSteps2.json",
```

```
                "versionId":"version-id",
                "etag":"etag-id"
            }
        },
        "stepType":"COPY",
        "stepName":"copyToShared"
    },
    "workflowId":"w-abcdef01234567890",
    "executionId":"execution-id",
    "transferDetails": {
        "serverId":"s-server-id",
        "username":"mary",
        "sessionId":"session-id"
    }
},
{
    "type":"StepCompleted",
    "details":{
        "output":{},
        "stepType":"COPY",
        "stepName":"copyToShared"
    },
    "workflowId":"w-abcdef01234567890",
    "executionId":"execution-id",
    "transferDetails":{
        "serverId":"server-id",
        "username":"mary",
        "sessionId":"session-id"
    }
},
{
    "type":"ExecutionCompleted",
    "details": {},
    "workflowId":"w-abcdef01234567890",
    "executionId":"execution-id",
    "transferDetails":{
        "serverId":"s-server-id",
        "username":"mary",
        "sessionId":"session-id"
    }
}
```

配置 CloudWatch 日志记录角色

要设置访问权限，您需要创建一个基于资源的 策略和一个提供该访问信息的 角色。

要启用 Amazon CloudWatch 日志记录，首先要创建启用 CloudWatch日志记录的 IAM 策略。然后，您需要创建一个 角色并将策略附加到该角色。您可在创建 SFTP 服务器或编辑现有 SFTP 服务器时执行此操作。有关的更多信息 CloudWatch，请参阅 Amazon 是什么 CloudWatch？ 以及什么是 Amazon CloudWatch 日志？ 在《亚马逊 CloudWatch 用户指南》中。

使用以下 IAM 策略示例来允许 CloudWatch 日志记录。

Use a logging role

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:DescribeLogStreams",
                "logs:CreateLogGroup",
                "logs:PutLogEvents"
            ],
            "Resource": "arn:aws:logs:*:*:log-group:/aws/transfer/*"
        }
    ]
}
```

Use structured logging

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
            "logs:CreateLogDelivery",
                "logs:GetLogDelivery",
                "logs:UpdateLogDelivery",
```

```
                "logs:DeleteLogDelivery",
                "logs:ListLogDeliveries",
                "logs:PutResourcePolicy",
                "logs:DescribeResourcePolicies",
                "logs:DescribeLogGroups"
            ],
            "Resource": "arn:aws:logs:region-id:AWS ##:log-group:/aws/transfer/*"
        }
    ]
}
```

在前述示例策略中，对于**Resource**，将*region-id* 和 *AWS ##*更换为您的值。例如，**"Resource": "arn:aws::logs:us-east-1:111122223333:log-group:/aws/transfer/*"**

然后，您可以创建一个角色并附加您创建的 CloudWatch 日志策略。

创建 IAM 角色并附加策略

1. 在导航窗格中，选择 Roles（角色），然后选择 Create role（创建角色）。

   在创建角色页面上，确保已选择AWS 服务。

2. 从服务列表中选择转移，然后选择下一步：权限。这将在和 IAM 角色 AWS Transfer Family 之间建立信任关系。此外，建议您在策略中使用 `aws:SourceAccount` 和 `aws:SourceArn` 条件键来防止出现混淆代理人问题。有关详细信息，请参阅文档。

   - 与以下机构建立信任关系的程序 AWS Transfer Family：建立信任关系
   - 代理困惑问题描述：代理困惑问题

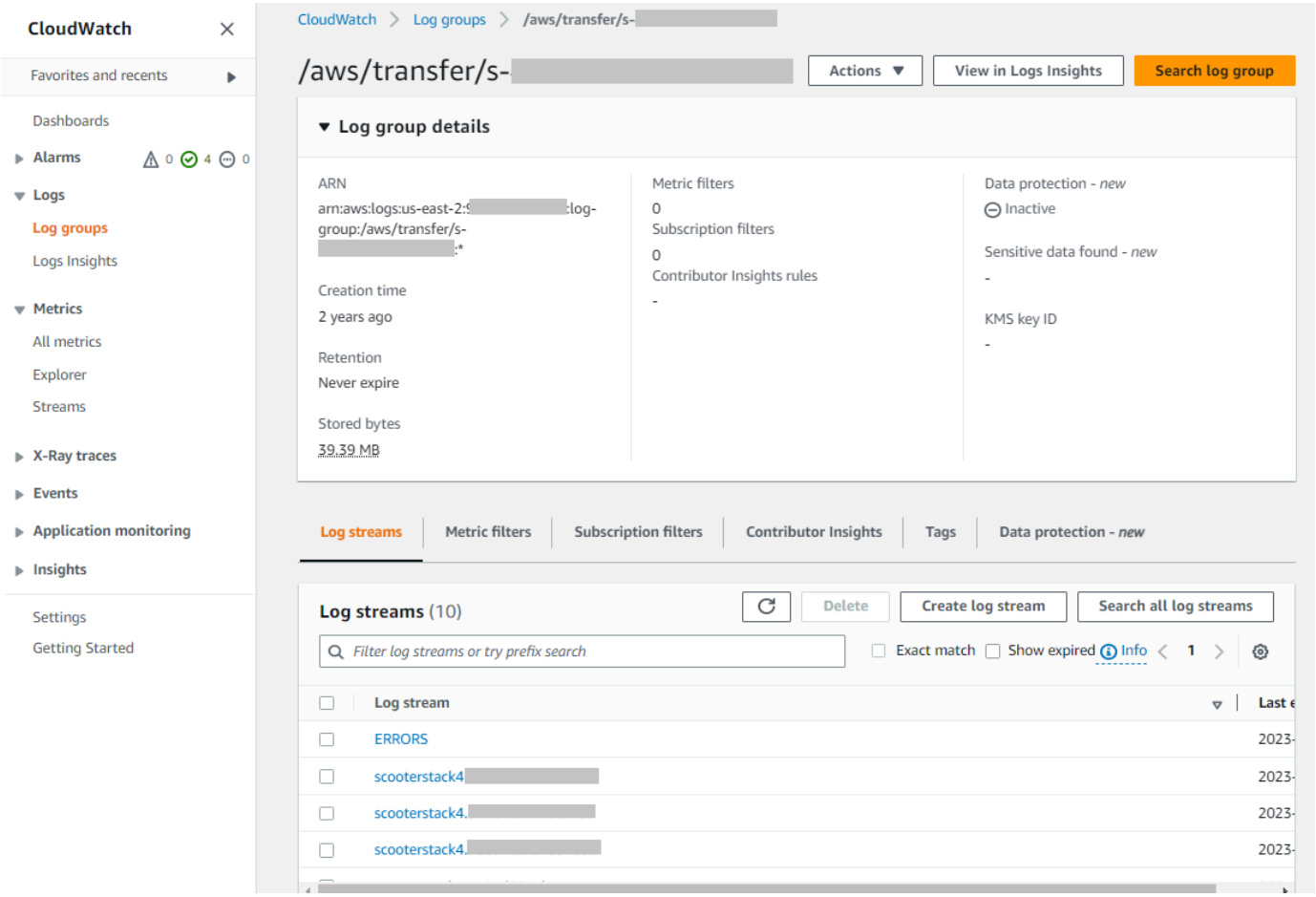
3. 在附加权限策略部分，找到并选择您刚刚创建的 CloudWatch 日志策略，然后选择下一步：标签。
4. （可选）输入标签的键和值，然后选择下一步：审核。
5. 在审核页面上，输入新角色的名称和描述，然后选择创建角色。
6. 要查看日志，请选择 Server ID (服务器 ID) 以打开服务器配置页面，然后选择 View logs (查看日志)。您将被重定向到 CloudWatch 控制台，您可以在其中查看日志流。

在服务器 CloudWatch 页面上，您可以看到用户身份验证（成功和失败）、数据上传（PUT操作）和数据下载（GET操作）的记录。

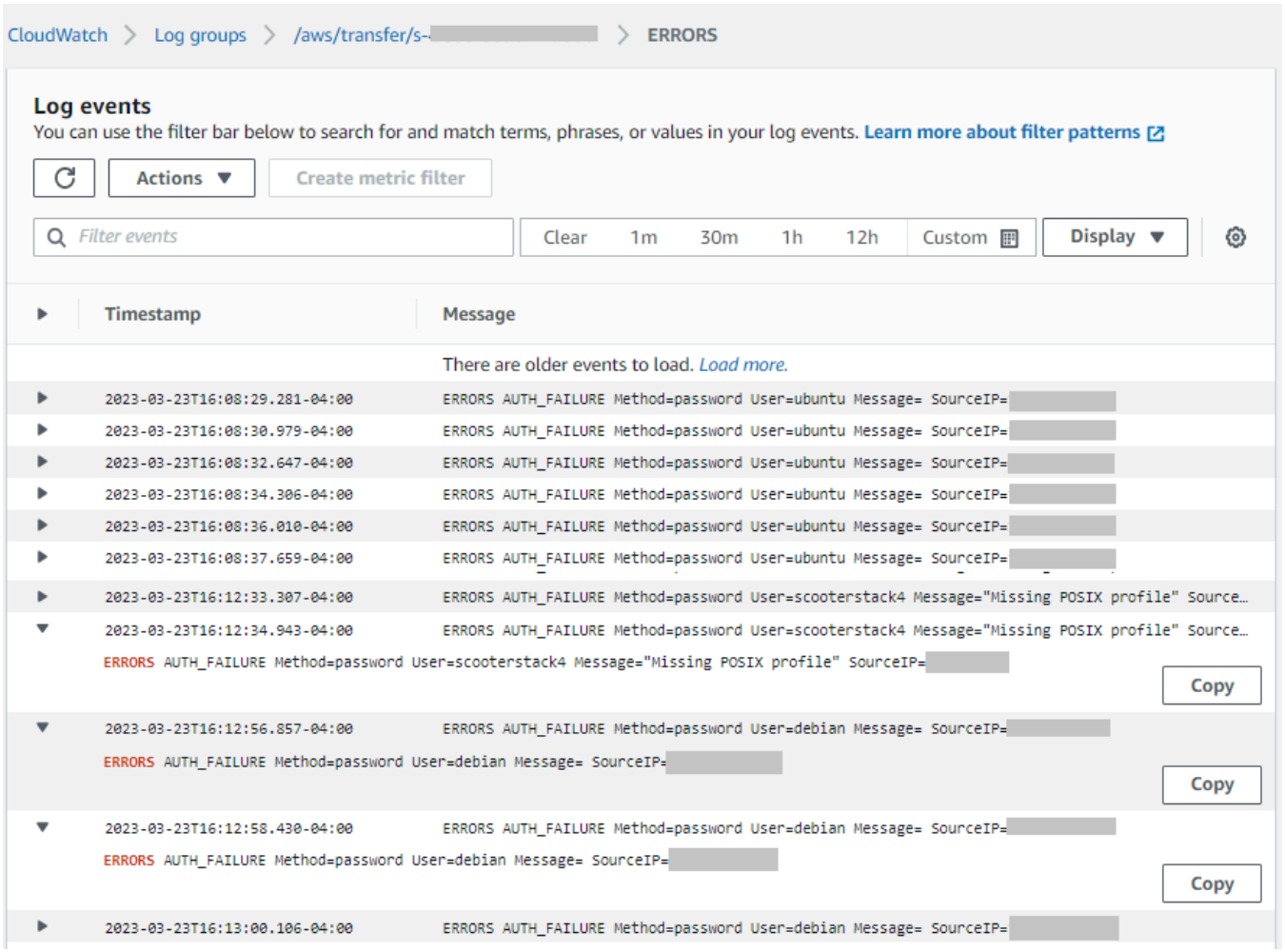
查看 Transfer Family 日志流

若要查看您的 Transfer Family 服务器日志

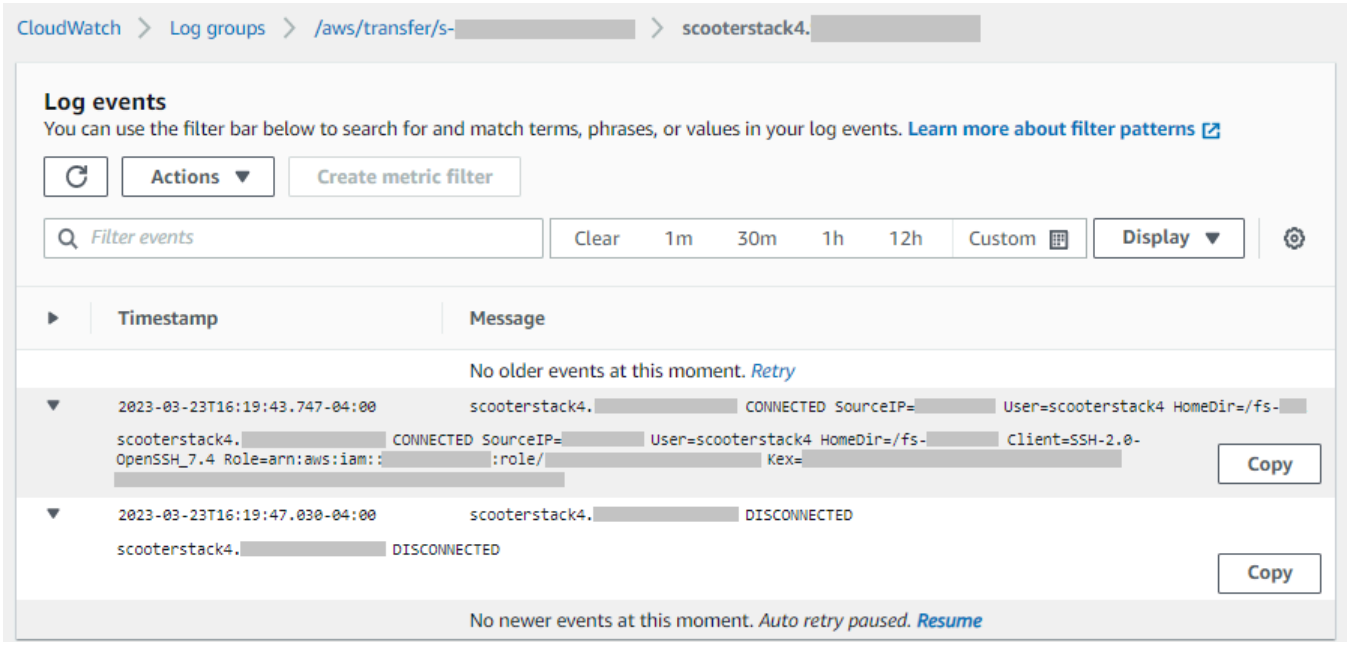
1. 导航到您的服务器详细信息页面。
2. 选择在 中查看日志。这将打开 Amazon CloudWatch。
3. 将显示选定服务器的日志组。



4. 您可以选择一项日志流，显示该流数据的详细信息和单个条目。
   - 如果列出了错误，则可以选择它，以查看服务器最新错误的详细信息。

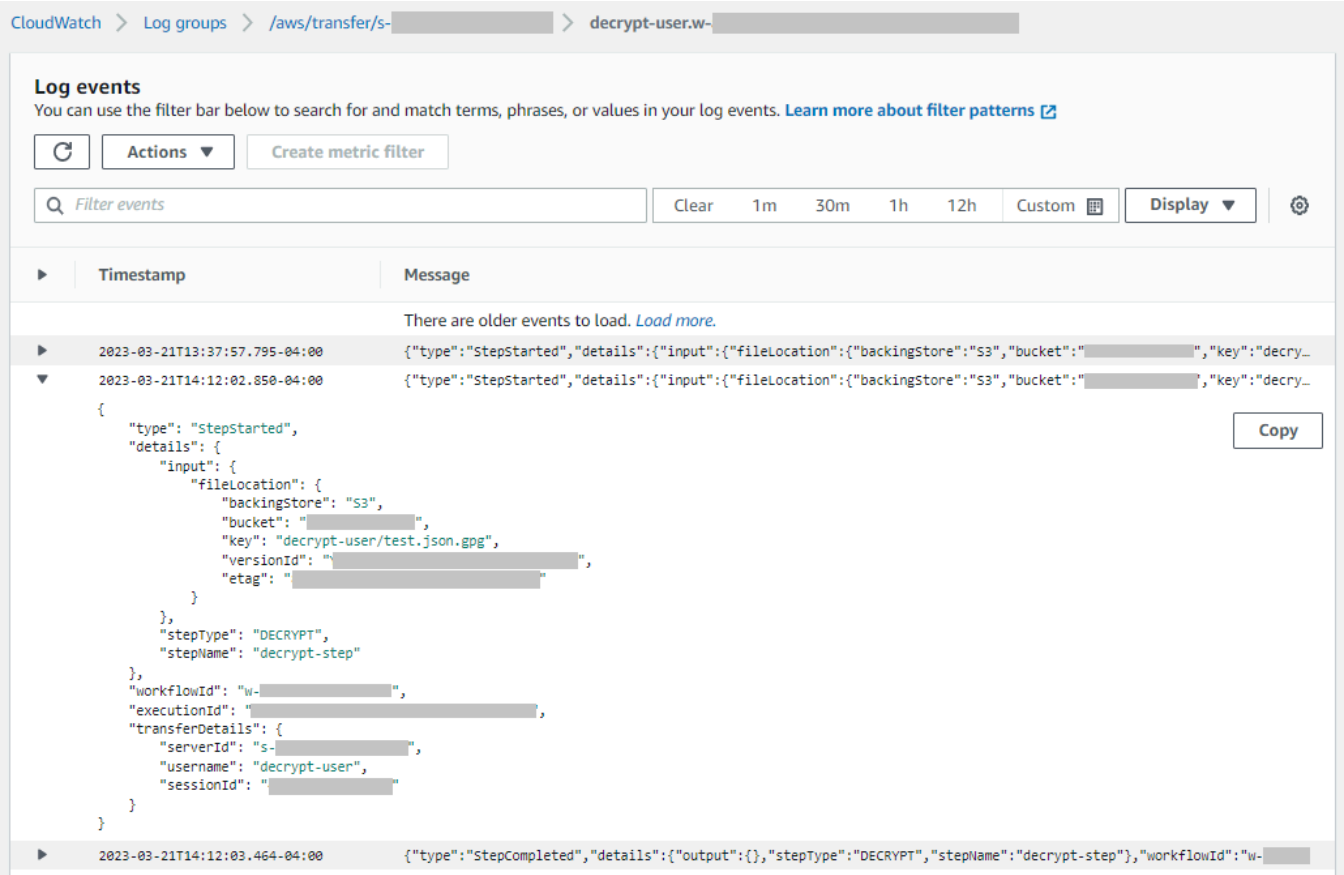


- 选择任何其他条目，以查看日志流示例。



- 如果您的服务器配备与之关联的托管工作流程，则可以查看工作流程运行日志。

> **Note**
>
> 工作流程的日志流格式为 *username*.*workflowId*.*uniqueStreamSuffix*。例如：对于日志流名称 decrypt-user.w-a1111222233334444.aaaa1111bbbb2222，其用户为 **decrypt-user**，工作流程为 **w-a1111222233334444**。



> **Note**
>
> 对于任何展开的日志条目，您可以通过选择复制将该条目复制到剪贴板。有关 CloudWatch 日志的更多详细信息，请参阅查看日志数据。

创建亚马逊 CloudWatch 警报

以下示例展示了如何使用 AWS Transfer Family 指标创建 Amazon CloudWatch 警报`FilesIn`。

CDK

```
new cloudwatch.Metric({
  namespace: "AWS/Transfer",
  metricName: "FilesIn",
  dimensionsMap: { ServerId: "s-00000000000000000" },
  statistic: "Average",
  period: cdk.Duration.minutes(1),
}).createAlarm(this, "AWS/Transfer FilesIn", {
  threshold: 1000,
  evaluationPeriods: 10,
  datapointsToAlarm: 5,
  comparisonOperator:
 cloudwatch.ComparisonOperator.GREATER_THAN_OR_EQUAL_TO_THRESHOLD,
});
```

AWS CloudFormation

```
Type: AWS::CloudWatch::Alarm
Properties:
  Namespace: AWS/Transfer
  MetricName: FilesIn
  Dimensions:
    - Name: ServerId
      Value: s-00000000000000000
  Statistic: Average
  Period: 60
  Threshold: 1000
  EvaluationPeriods: 10
  DatapointsToAlarm: 5
  ComparisonOperator: GreaterThanOrEqualToThreshold
```

Amazon S3 API 调用 S3 访问日志的记录

如果您使用 Amazon S3 访问日志识别代表文件传输用户提出的 S3 请求，则使用`RoleSessionName`显示被假定为提供文件传输提供服务的 IAM 角色。它还显示其他信息，例如用于传输的用户名、会话 ID 以及服务器 ID。格式为 `[AWS:Role Unique Identifier]/`

username.sessionid@server-id，且包含在 “请求者” 字段中。例如，以下是来自 S3 访问日志的、用于复制到 S3 存储桶中的“请求者”字段示例内容。

```
arn:aws:sts::AWS-Account-ID:assumed-role/IamRoleName/
username.sessionid@server-id
```

在上述中“请求者”字段中，它显示了名为IamRoleName的 IAM 角色。有关唯一标识符的更多信息，请参阅《IAM 用户指南》AWS Identity and Access Management 中的 IAM 标识符。

混淆代理问题限制示例

混淆代理问题是一个安全性问题，即不具有操作执行权限的实体可能会迫使具有更高权限的实体执行该操作。在中 AWS，跨服务模仿可能会导致混乱的副手问题。有关更多详细信息，请参阅防止跨服务混淆代理。

> **Note**
>
> 在以下示例中，将每个 *user input placeholder* 替换为您自己的信息。
> 在这些示例中，如果您的服务器未附加任何工作流程，则可以删除该工作流程的 ARN 详细信息。

以下示例日志/调用策略允许账户中的任何服务器（和工作流程）代入该角色。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAllServersWithWorkflowAttached",
            "Effect": "Allow",
            "Principal": {
                "Service": "transfer.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "account-id"
                },
                "ArnLike": {
                   "aws:SourceArn": [
                     "arn:aws:transfer:region:account-id:server/*",
```

```
                    "arn:aws:transfer:region:account-id:workflow/*"
                ]
            }
        }
    }
]
}
```

以下示例日志/调用策略允许特定的服务器（和工作流程）担任该角色。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSpecificServerWithWorkflowAttached",
            "Effect": "Allow",
            "Principal": {
                "Service": "transfer.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "account-id"
                },
                "ArnEquals": {
                   "aws:SourceArn": [
                       "arn:aws:transfer:region:account-id:server/server-id",
                       "arn:aws:transfer:region:account-id:workflow/workflow-id"
                   ]
                }
            }
        }
    ]
}
```

CloudWatch Transfer Family 的日志结构

本主题介绍了 Transfer Family 日志中填充的字段：包括 JSON 结构化日志条目和旧日志条目。

主题

- Transfer Family 的 JSON 结构化
- Transfer Family 的旧日志

Transfer Family 的 JSON 结构化

下表详细介绍了采用新的 JSON 结构化日志格式的 Transfer Family SFTP/FTP/FTPS 操作的日志条目字段。

字段	描述	示例条目
activity-type	The action by the user	打开 | 关闭 | 部分关闭 | 已断开连接 | 已连接
bytes-in	Number of bytes uploaded by the user	29238420042
bytes-out	Number of bytes downloaded by the user	23094032490328
ciphers	Specifies the SSH cipher negotiated for the connection (available ciphers are listed in 加密算法)	aes256-gcm@openssh.com
client	The user's client software	SSH-2.0-OpenSSH_7.4
home-dir	The directory that the end user lands on when they connect to the endpoint if their home directory type is ##: if they have a logical home directory, this value is always /	/user-home-bucket/test
kex	Specifies the negotiated SSH key exchange (KEX) for the connection (available KEX are listed in 加密算法)	diffie-hellman-group14-sha256
message	Provides more information related to the error	*<string>*
method	The authentication method	publickey

字段	描述	示例条目
mode	Specifies how a client opens a file	CREATE | TRUNCATE | WRITE
operation	The client operation on a file	OPEN | CLOSE
path	Actual file path affected	/user-test-bucket/test-file-1.pdf
resource-arn	A system-assigned, unique identifier for a specific resource (for example, a server)	arn: aws: transfer: ap-northeast-1:12346789012: server/s-1234567890akeu2js2
role	The IAM role of the user	arn: aws: iam:: 0293883675: 角色/测试用户角色
session-id	A system-assigned, unique identifier for a single session	9ca9a0e1cec6ad9d
source-ip	Client IP address	18.323.0.129
user	The end user's username	myname192
user-policy	The permissions specified for the end user: this field is populated if the user's policy is a session policy.	The JSON code for the session policy that is being used

Transfer Family 的旧日志

下表包含各种 Transfer Family 操作的日志条目的详细信息。

> **Note**
>
> 这些条目不是采用新的 JSON 结构化日志格式。

下表以新的 JSON 结构化日志格式包含各种 Transfer Family 操作的日志条目的详细信息。

操作	Amazon 日志中的相应 CloudWatch 日志
身份验证失败次数	ERRORS AUTH_FAILURE Method=publickey User=lhr Message="RSA SHA256:Lfz3R2nmLY4raK+b7Rb1rSvUIbAE+a+Hxg0c7l1JIZ0" SourceIP=3.8.172.211
复制/标记/删除/解密工作流程	{“type”:” “ , ” details” : {“input” : {StepStarted“fileLocation” : {“backingStore”: “EFS”、“Filesystemid”: “fs-12345678”、“path”:” /lhr/regex.py “}”}、“stepType”: “TAG”、“stepName”: “successful_tag_step”}、“workflowID”: “workflowID”: “workflowID”: “workflowID”: “workflowID”: “workflowID”: “workF11aaaa2222bbb3"、“executionID”: “81234abcd-1234-efgh-5678-ijklmnopqr90"、“TransferDetails”: {“serverID”: “s-1234abcdef5678efghi”、“用户名”: “lhr”、“sessionID”: “1234567890abcdef0"}}
自定义步骤工作流程	{“type”:” “ , ” details” : {“输出” : {CustomStepInvoked“token”: “mzm4mjg5ywutyt EzMy 00 YjIz LWI3OG MtYz U4OGI2 ZjQyMz E5"}、“stepType”: “CUSTOM”、“stepName”: “efs-s3_copy_2"}、“workflowID”: “w-9283e49d33297c3f7"、“executionID”: “w-9283e49d33297c3f7”: “1234abcd-1234-efgh-5678-ijklmnopqr90" , “TransferDetails” : {“serverID”: “s-zzzz11aaaa222223"、“用户名”: “lhr”、“sessionID”: “1234567890abcdef0"}}
删除	lhr.33a8fb495ffb383b DELETE Path=/bucket/user/123.jpg
Downloads	lhr.33a8fb495ffb383b OPEN Path=/bucket/user/123.jpg Mode=READ

操作	Amazon 日志中的相应 CloudWatch 日志
	llhr.33a8fb495ffb383b CLOSE path=/bucket/user/123.jpg =3618546 BytesOut
登录/登出	user.914984e553bcddb6 CONNECTED SourceIP=1.22.111.222 user=LOGICAL client=ssh-2.0-openssh_7.4 role=arn: aws:: iam:: 123456789012: role/sftp-s3-access HomeDir user.914984e553bcddb6 DISCONNECTED
重命名	lhr.33a8fb495ffb383b 重命名路径=/bucket/user/lambo.png =/bucket/user/ferrari.png =/bucket/user/ferrari. NewPath
工作流程错误日志示例	{“type”:” “ , ” details” : {“errorType” : StepErrored“BAD_REQUEST” , “ErrorMessage” : “无法标记 Efs 文件” , “stepType” : “TAG” , “stepName” : “successful_tag_step”} , “w-1234abcd5678efghi” , “executionID” : “81234abcd5678efghi” : “81234abcd5678efghi” : “81234abcd5678efghi” : “81234abcd5678efghi” : “8cd-1234-efgh-5678-ijklmnopqr90" , “TransferDetails” : {“serverID”: “s-1234abcd5678efghi”、“用户名”: “lhr”、“sessionID”: “1234567890abcdef0"}}
symlinks	lhr.eb49cf7b8651e6d5 CREATE_SYMLINK =/fs-12345678/lhr/pqr.jpg =abc.jpg =abc.jpg LinkPath TargetPath
Uploads	lhr.33a8fb495ffb383b OPEN Path=/bucket/user/123.jpg Mode=CREATE|TRUNCATE|WRITE lhr.33a8fb495ffb383b CLOSE path=/bucket/user/123.jpg =3618546 BytesIn

操作	Amazon 日志中的相应 CloudWatch 日志
工作流	{“type”:” “ , ” details” : {“input” : {Executio nStarted“backingStore”: “EFS”、“Filesystemi d”: “fs-12345678”、“path”:” /lhr/regex.py “}}}、initialFileLocation “workflowID”: “w-1111aa aa2222bbbb3”、“executionID”: “1234abcd -1234-efbbid”: “w-11aaaa2222bbbb3”、“execut ionID”: “1234abcd-1234-efbbid”: “w-11aaaa 22gh-5678-ijklmnopqr90" , “TransferDetails” : {“serverID”: “s-zzzz1111aaaa222223"、“用户名 ”: “lhr”、“sessionID”: “1234567890abcdef0"}} {“type”:” “ , ” details” : {“input” : {StepStar ted“fileLocation” : {“backingStore”: “EFS”、“Fi lesystemid”: “fs-12345678”、“path”:” /lhr/rege x.py “}}、“stepType”: “CUSTOM”、“stepName ”: “efs-s3_copy_2"}、“workflowID”: “workflow ID”: “workflowID”: “workflowID”: “9283e49d 33297c3f7"、“executionID”: “1234abcd-1234- efgh-5678-ijklmnopqr90"、“TransferDetails”: {“serverID”: “s-18ca49dce5d842e0b”、“用户 名”: “lhr”、“sessionID”: “1234567890abb”、“用 户名”: “lhr”、“sessionID”: “1234567890aba” cdef0"}}

CloudWatch 日志条目示例

本主题介绍示例日志条目。

主题

- 传输会话日志条目示例
- SFTP 连接器的日志条目示例
- 密钥交换算法失败的日志条目示例

传输会话日志条目示例

在此示例中，SFTP 用户连接到 Transfer Family 服务器，上传文件，然后断开与会话的连接。

以下日志条目反映了连接到 Transfer Family 服务器的 SFTP 用户。

```
{
   "role": "arn:aws:iam::500655546075:role/scooter-transfer-s3",
   "activity-type": "CONNECTED",
   "ciphers": "chacha20-poly1305@openssh.com,chacha20-poly1305@openssh.com",
   "client": "SSH-2.0-OpenSSH_7.4",
   "source-ip": "52.94.133.133",
   "resource-arn": "arn:aws:transfer:us-east-1:500655546075:server/
s-3fe215d89f074ed2a",
   "home-dir": "/scooter-test/log-me",
   "user": "log-me",
   "kex": "ecdh-sha2-nistp256",
   "session-id": "9ca9a0e1cec6ad9d"
}
```

以下日志条目反映了 SFTP 用户将文件上传到其 Amazon S3 存储桶的情况。

```
{
   "mode": "CREATE|TRUNCATE|WRITE",
   "path": "/scooter-test/log-me/config-file",
   "activity-type": "OPEN",
   "resource-arn": "arn:aws:transfer:us-east-1:500655546075:server/
s-3fe215d89f074ed2a",
   "session-id": "9ca9a0e1cec6ad9d"
}
```

以下日志条目反映了 SFTP 用户与 SFTP 会话断开连接的情况。首先，客户端关闭与存储桶的连接，然后断开 SFTP 会话。

```
{
   "path": "/scooter-test/log-me/config-file",
   "activity-type": "CLOSE",
   "resource-arn": "arn:aws:transfer:us-east-1:500655546075:server/
s-3fe215d89f074ed2a",
   "bytes-in": "121",
   "session-id": "9ca9a0e1cec6ad9d"
}
```

```
{
   "activity-type": "DISCONNECTED",
   "resource-arn": "arn:aws:transfer:us-east-1:500655546075:server/
s-3fe215d89f074ed2a",
   "session-id": "9ca9a0e1cec6ad9d"
}
```

SFTP 连接器的日志条目示例

本节包含成功和不成功传输的示例日志。日志生成到名为的日志组/aws/transfer/*connector-id*，其中 c*onnector-id ### SFTP ####*标识符。

> **Note**
>
> 只有在执行StartFileTransfer命令时才会生成 SFTP 连接器的日志条目。

此日志条目适用于成功完成的传输。

```
{
    "operation": "RETRIEVE",
    "timestamp": "2023-10-25T16:33:27.373720Z",
    "connector-id": "connector-id",
    "transfer-id": "transfer-id",
    "file-transfer-id": "transfer-id/file-transfer-id",
    "url": "sftp://192.0.2.0",
    "file-path": "/remotebucket/remotefilepath",
    "status-code": "COMPLETED",
    "start-time": "2023-10-25T16:33:26.945481Z",
    "end-time": "2023-10-25T16:33:27.159823Z",
    "account-id": "480351544584",
    "connector-arn": "arn:aws:transfer:us-east-1:480351544584:connector/connector-id",
    "local-directory-path": "/connectors-localbucket"
}
```

此日志条目适用于超时但未成功完成的传输。

```
{
    "operation": "RETRIEVE",
    "timestamp": "2023-10-25T22:33:47.625703Z",
    "connector-id": "connector-id",
```

```
    "transfer-id": "transfer-id",
    "file-transfer-id": "transfer-id/file-transfer-id",
    "url": "sftp://192.0.2.0",
    "file-path": "/remotebucket/remotefilepath",
    "status-code": "FAILED",
    "failure-code": "TIMEOUT_ERROR",
    "failure-message": "Transfer request timeout.",
    "account-id": "480351544584",
    "connector-arn": "arn:aws:transfer:us-east-1:480351544584:connector/connector-id",
    "local-directory-path": "/connectors-localbucket"
}
```

前面日志示例中一些关键字段的描述。

- timestamp表示何时将日志添加到 CloudWatch。 start-time并end-time对应于连接器实际开始和完成传输的时间。
- transfer-id是为每个start-file-transfer请求分配的唯一标识符。如果用户在单个 start-file-transfer API 调用中传递多个文件路径，则所有文件共享相同的路径transfer-id。
- file-transfer-id是为每个传输的文件生成的唯一值。请注意，的初始file-transfer-id部分与相同transfer-id。

密钥交换算法失败的日志条目示例

本节包含密钥交换算法 (KEX) 失败的示例日志。这些是结构化日志的 ER RO RS 日志流中的示例。

此日志条目是存在主机密钥类型错误的示例。

```
{
    "activity-type": "KEX_FAILURE",
    "source-ip": "999.999.999.999",
    "resource-arn": "arn:aws:transfer:us-east-1:999999999999:server/
s-999999999999999999",
    "message": "no matching host key type found",
    "kex": "ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ecdsa-sha2-
nistp256-cert-v01@openssh.com,ecdsa-sha2-nistp384-cert-v01@openssh.com,ecdsa-sha2-
nistp521-cert-v01@openssh.com,ssh-ed25519,ssh-rsa,ssh-dss"
}
```

此日志条目是 KEX 不匹配的示例。

```
{
```

```
    "activity-type": "KEX_FAILURE",
    "source-ip": "999.999.999.999",
    "resource-arn": "arn:aws:transfer:us-east-1:999999999999:server/
s-99999999999999999",
    "message": "no matching key exchange method found",
    "kex": "diffie-hellman-group1-sha1,diffie-hellman-group14-sha1,diffie-hellman-
group14-sha256"
}
```

使用 T CloudWatch ransfer Family 的指标

> **Note**
>
> 你还可以从 Transfer Family 控制台获取 Transfer Family 指标。有关详细信息，请参阅 在控制台中监控使用情况

您可以使用 CloudWatch 指标获取有关服务器的信息。指标表示发布到的一组按时间顺序排列的数据点。 CloudWatch使用指标时，必须指定 Transfer Family 命名空间、指标名和维度。有关指标的更多信息，请参阅 Amazon CloudWatch 用户指南中的指标。

下表描述了 Transfer Family 的 CloudWatch 指标。

命名空间	指标	描述
AWS/Transfer	BytesIn	传输至服务器的字节总数。 单位：计数 时间：5 分钟
	BytesOut	从服务器传出来的字节总数。 单位：计数 时间：5 分钟
	FilesIn	传输至服务器的字节总数。 对于使用 AS2 协议的服务器，此指标表示所收到的消息数量。

命名空间	指标	描述
		单位：计数 时间：5 分钟
	FilesOut	从服务器传出来的字节总数。 单位：计数 时间：5 分钟
	InboundMessage	成功从交易伙伴处收到的 AS2 消息总数。 单位：计数 时间：5 分钟
	InboundFailedMessage	未成功从交易伙伴处收到的 AS2 消息总数。也就是说，交易伙伴发送了一条消息，但是 Transfer Family 服务器无法成功处理该消息。 单位：计数 时间：5 分钟
	OnPartialUploadExecutionsStarted	服务器上启动的工作 on-partial-upload 流程执行总数。 单位：计数 时间：1 分钟
	OnPartialUploadExecutionsSuccess	服务器上成功执行 on-partial-upload 的工作流程总数。 单位：计数 时间：1 分钟
	OnPartialUploadExecutionsFailed	服务器上执行失败 on-partial-upload 的工作流程总数。 单位：计数 时间：1 分钟

命名空间	指标	描述
	OnUploadExecutionsStarted	服务器上启动的工作流程执行总数。 单位：计数 时间：1 分钟
	OnUploadExecutionsSuccess	服务器上执行成功的工作流程总数。 单位：计数 时间：1 分钟
	OnUploadExecutionsFailed	服务器上执行失败的工作流程总数。 单位：计数 时间 = 1 分钟

Transfer Family 维度

维度是一个名称/值对，它是指标标识的一部分。有关尺寸的更多信息，请参阅 Amazon CloudWatch 用户指南中的尺寸。

下表描述了 Transfer Family 的 CloudWatch 维度。

维度	描述
ServerId	用户的唯一 ID。

AWS 用户通知服务 与一起使用 AWS Transfer Family

要获得有关 AWS Transfer Family 事件的通知，您可以使用AWS 用户通知服务设置各种交付渠道。当事件与您指定的规则匹配时，您会收到通知。

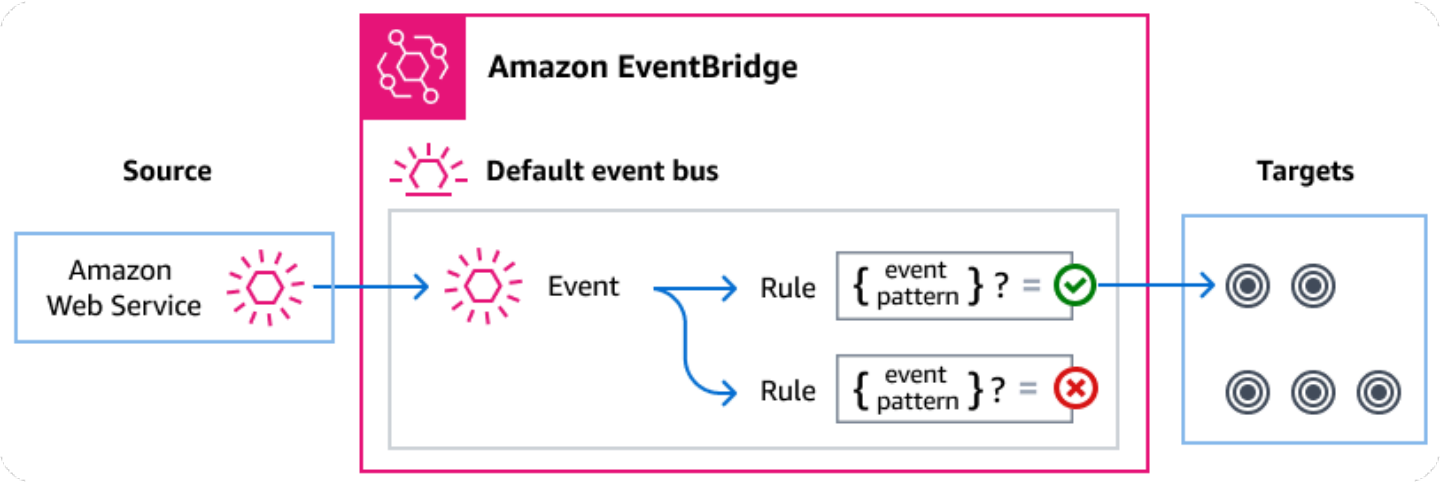
可以通过多个渠道接收事件通知，包括电子邮件、AWS Chatbot聊天通知或AWS Console Mobile Application推送通知。您还可以在控制台通知中心查看通知。 用户通知服务 支持聚合，这可以减少您在特定事件期间收到的通知数量。

有关更多信息，请参阅使用 AWS Transfer Family 托管工作流程自定义文件传送通知博客文章和什么是 AWS 用户通知服务？ 在《AWS 用户通知服务 用户指南》中。

使用管理 Transfer Family 事件 Amazon EventBridge

Amazon EventBridge 是一项无服务器服务，它使用事件将应用程序组件连接在一起，这使您可以更轻松地构建可扩展的事件驱动应用程序。事件驱动架构是一种构建松散耦合的软件系统的风格，这些系统通过发射和响应事件来协同工作。事件代表资源或环境中的变化。

与许多 AWS 服务一样， Transfer Family 生成事件并将其发送到 EventBridge 默认事件总线。请注意，默认事件总线会在每个 AWS 账户中自动配置。事件总线是接收事件并将其传送到零个或多个目的地或目标的路由器。您可以为事件总线指定规则，该总线在事件到达时对其进行评估。每条规则都会检查事件是否与规则的事件模式相匹配。如果事件匹配，则事件总线会将事件发送到一个或多个指定的目标。



主题

- Transfer Family 事件
- 使用 EventBridge 规则发送 Transfer Family 事件
- Amazon EventBridge 权限
- 其他 EventBridge 资源
- Transfer Family 事件详情参考

Transfer Family 事件

Transfer Family 自动将事件发送到默认 EventBridge 事件总线。您可以在事件总线上创建规则，其中每条规则都包含一个事件模式和一个或多个目标。与规则的事件模式相匹配的事件会尽最大努力传送到指定的目标，但是，有些事件可能会乱序传送。

以下事件由生成 Transfer Family。有关更多信息，请参阅《Amazon EventBridge 用户指南》中的EventBridge 事件。

SFTP、FTPS 和 FTP 服务器事件

活动详情类型	描述
FTP 文件服务器下载已完成	已成功下载适用于 FTP 协议的文件。
FTP 文件服务器下载失败	尝试下载 FTP 协议的文件失败。
FTP 文件服务器上传已完成	已成功上传适用于 FTP 协议的文件。
FTP 文件服务器上传失败	尝试上传 FTP 协议的文件失败。
FTPS 文件服务器下载已完成	已成功下载适用于 FTPS 协议的文件。
FTPS 文件服务器下载失败	尝试下载 FTPS 协议的文件失败。
FTPS 文件服务器上传已完成	已成功上传适用于 FTPS 协议的文件。
FTPS 文件服务器上传失败	尝试上传 FTPS 协议的文件失败。
SFTP 服务器文件下载已完成	已成功下载适用于 SFTP 协议的文件。
SFTP 服务器文件下载失败	尝试下载 SFTP 协议的文件失败。
SFTP 服务器文件上传已完成	已成功上传适用于 SFTP 协议的文件。
SFTP 服务器文件上传失败	尝试上传 SFTP 协议的文件失败。

SFTP 连接器事件

活动详情类型	描述
SFTP 连接器文件发送已完成	从连接器到远程 SFTP 服务器的文件传输已成功完成。
SFTP 连接器文件发送失败	从连接器向远程 SFTP 服务器传输文件失败。
SFTP 连接器文件检索已完成	已成功完成从远程 SFTP 服务器到连接器的文件传输。

活动详情类型	描述
检索 SFTP 连接器文件失败	从远程 SFTP 服务器向连接器传输文件失败。

A2S 赛事

活动详情类型	描述
AS2 有效载荷接收已完成	AS2 消息的有效载荷已收到。
AS2 有效载荷接收失败	尚未收到 AS2 消息的有效负载。
AS2 有效载荷发送已完成	AS2 消息的有效载荷已成功发送。
AS2 有效负载发送失败	AS2 消息的有效负载发送失败。
AS2 MDN 接收已完成	已收到 AS2 邮件的邮件处置通知。
AS2 MDN 接收失败	尚未收到 AS2 邮件的邮件处理通知。
AS2 MDN 发送已完成	AS2 邮件的邮件处置通知已成功发送。
AS2 MDN 发送失败	AS2 邮件的邮件处置通知发送失败。

使用 EventBridge 规则发送 Transfer Family 事件

如果要 EventBridge 使用默认事件总线向目标发送 Transfer Family 事件，则必须创建一个规则，其中包含与所需事件中的数据匹配 Transfer Family 的事件模式。

您可以按照以下常规步骤创建规则：

1. 为规则创建事件模式，指定以下内容：
   - Transfer Family 是规则正在评估的事件的来源。
   - （可选）要与之匹配的任何其他事件数据。

   有关更多信息，请参阅 ???。
2. （可选）创建输入转换器，在将信息 EventBridge 发送到规则目标之前，对事件中的数据进行自定义。

有关更多信息，请参阅《EventBridge 用户指南》中的输入转换。

3. 指定要 EventBridge 向其发送与事件模式匹配的事件的目标。

   目标可以是其他 AWS 服务、软件即服务 (SaaS) 应用程序、API 目标或其他自定义端点。有关更多信息，请参阅《EventBridge 用户指南》中的目标。

有关创建事件总线规则的全面说明，请参阅《EventBridge 用户指南》中的创建对事件作出反应的规则。

为事件创建 Transfer Family 事件模式

将事件 Transfer Family 传送到默认事件总线时， EventBridge 使用为每条规则定义的事件模式来确定是否应将事件传送到规则的目标。事件模式与所需 Transfer Family 事件中的数据相匹配。每个事件模式都是一个 JSON 对象，其中包含以下内容：

- 标识发送事件的服务的 `source` 属性。对于 Transfer Family 事件，来源是`aws.transfer`。
- （可选）包含要匹配的事件类型数组的`detail-type`属性。
- （可选）包含要匹配的任何其他事件数据的`detail`属性。

例如，以下事件模式与来自的所有事件匹配 Transfer Family：

```
{
  "source": ["aws.transfer"]
}
```

以下事件模式示例匹配所有 SFTP 连接器事件：

```
{
  "source": ["aws.transfer"],
  "detail-type": ["SFTP Connector File Send Completed", "SFTP Connector File Retrieve
 Completed",
                  "SFTP Connector File Retrieve Failed", "SFTP Connector File Send
 Failed"]
}
```

以下事件模式示例匹配所有 Transfer Family 失败事件：

```
{
```

```
  "source": ["aws.transfer"],
  "detail-type": [{"wildcard", "*Failed"}]
}
```

以下事件模式示例匹配成功下载的 SFTP 用户#：

```
{
  "source": ["aws.transfer"],
  "detail-type": ["SFTP Server File Download Completed"],
  "detail": {
    "username": [username]
  }
}
```

有关写入事件模式的更多信息，请参阅《EventBridge 用户指南》中的事件模式。

测试事件模式中的 Transfer Family 事件 EventBridge

您可以使用 EventBridge 沙盒快速定义和测试事件模式，而不必完成创建或编辑规则的更广泛过程。使用沙盒，您可以定义事件模式并使用示例事件来确认该模式是否与所需事件匹配。 EventBridge 允许您选择通过直接从沙箱中使用该事件模式来创建新规则。

有关更多信息，请参阅EventBridge 用户指南中的使用 EventBridge 沙盒测试事件模式。

Amazon EventBridge 权限

Transfer Family 不需要任何其他权限即可向其发送事件 Amazon EventBridge。

您指定的目标可能需要特定的权限或配置。有关为目标使用特定服务的更多详细信息，请参阅《Amazon EventBridge 用户指南》中的 Amazon EventBridge 目标。

其他 EventBridge 资源

有关如何使用 EventBridge 处理和管理事件的更多信息，请参阅《Amazon EventBridge 用户指南》中的以下主题。

- 有关事件总线工作原理的详细信息，请参阅 Amazon EventBridge 事件总线。
- 有关事件结构的信息，请参阅事件。

- 有关构造事件模式 EventBridge 以便在将事件与规则进行匹配时使用的信息，请参阅事件模式。
- 有关创建规则以指定 EventBridge 所处理事件的信息，请参阅规则。
- 有关如何指定向哪些服务或其他目的地 EventBridge 发送匹配事件的信息，请参阅目标。

Transfer Family 事件详情参考

来自 AWS 服务的所有事件都有一组公共字段，其中包含有关该事件的元数据。这些元数据可以包括作为事件来源的 AWS 服务、事件的生成时间、事件发生的账户和区域等。有关这些常规字段的定义，请参阅《Amazon EventBridge 用户指南》中的事件结构参考。

此外，每个事件都有一个 `detail` 字段，其中包该特定事件专有的数据。以下参考定义了各种 Transfer Family 事件的详细信息字段。

使用 EventBridge 选择和管理 Transfer Family 事件时，请考虑以下几点：

- 来自的所有事件的`source`字段均设置 Transfer Family 为`aws.transfer`。
- `detail-type` 字段指定事件类型。

  例如，`FTP File Server Download Completed`。
- `detail` 字段包含该特定事件专有的数据。

有关如何构造使规则能够匹配 Transfer Family 事件的事件模式的信息，请参阅《Amazon EventBridge 用户指南》中的事件模式。

有关事件及其 EventBridge 处理方式的更多信息，请参阅《Amazon EventBridge 用户指南》中的Amazon EventBridge 事件。

主题

- SFTP、FTPS 和 FTP 服务器事件
- SFTP 连接器事件
- AS2 赛事

SFTP、FTPS 和 FTP 服务器事件

以下是 SFTP、FTPS 和 FTP 服务器事件的详细信息字段：

- FTP 文件服务器下载已完成
- FTP 文件服务器下载失败
- FTP 文件服务器上传已完成
- FTP 文件服务器上传失败
- FTPS 文件服务器下载已完成
- FTPS 文件服务器下载失败
- FTPS 文件服务器上传已完成
- FTPS 文件服务器上传失败
- SFTP 服务器文件下载已完成
- SFTP 服务器文件下载失败
- SFTP 服务器文件上传已完成
- SFTP 服务器文件上传失败

下面包含`source`和`detail-type`字段，因为它们包含 Transfer Family 事件的特定值。有关所有事件中包含的其他元数据字段的定义，请参阅Amazon EventBridge 用户指南中的事件结构参考。

```
{
  . . .,
  "detail-type": "string",
  "source": "aws.transfer",
  . . .,
  "detail": {
    "failure-code" : "string",
    "status-code" : "string",
    "protocol" : "string",
    "bytes" : "number",
    "client-ip" : "string",
    "failure-message" : "string",
    "end-timestamp" : "string",
    "etag" : "string",
    "file-path" : "string",
    "server-id" : "string",
    "username" : "string",
    "session-id" : "string",
    "start-timestamp" : "string"
  }
}
```

`detail-type`

标识事件的类型。

对于此事件，该值是先前列出的 SFTP、FTPS 或 FTP 服务器事件名称之一。

`source`

标识生成事件的服务。对于 Transfer Family 事件，此值为`aws.transfer`。

`detail`

包含关于事件信息的 JSON 对象。生成事件的服务决定该字段的内容。

对于此事件，数据包括以下内容：

`failure-code`

传输失败原因的类别。值：`PARTIAL_UPLOAD | PARTIAL_DOWNLOAD | UNKNOWN_ERROR`

`status-code`

传输是否成功。价值观：`COMPLETED | FAILED`。

`protocol`

用于传输的协议。值：`SFTP | FTPS | FTP`

`bytes`

传输的字节数。

`client-ip`

参与传输的客户端 IP 地址

`failure-message`

对于失败的传输，提供传输失败原因的详细信息。

`end-timestamp`

对于成功传输，指文件处理完毕的时间戳。

`etag`

实体标签（仅用于 Amazon S3 文件）。

`file-path`

正在传输的文件的路径。

`server-id`

Transfer Family 服务器的唯一 ID。

`username`

正在执行转移的用户。

`session-id`

传输会话的唯一标识符。

`start-timestamp`

对于成功传输，指文件处理开始时间的时间戳。

Example SFTP 服务器文件下载失败示例事件

以下示例显示了在 SFTP 服务器上下载失败的事件（Amazon EFS 是否正在使用存储空间）。


```
{
    "version": "0",
    "id": "event-ID",
    "detail-type": "SFTP Server File Download Failed",
    "source": "aws.transfer",
    "account": "958412138249",
    "time": "2024-01-29T17:20:27Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:transfer:us-east-1:958412138249:server/s-1234abcd5678efghi"
    ],
    "detail": {
        "failure-code": "PARTIAL_DOWNLOAD",
        "status-code": "FAILED",
        "protocol": "SFTP",
        "bytes": 4100,
        "client-ip": "IP-address",
        "failure-message": "File was partially downloaded.",
        "end-timestamp": "2024-01-29T17:20:27.749749117Z",
        "file-path": "/fs-1234abcd5678efghi/user0/test-file",
        "server-id": "s-1234abcd5678efghi",
        "username": "test",
        "session-id": "session-ID",
        "start-timestamp": "2024-01-29T17:20:16.706282454Z"
```

```
    }
}
```

Example FTP 文件服务器上传已完成示例事件

以下示例显示了在 FTP 服务器上成功完成上传的事件（Amazon S3 是否正在使用存储空间）。

```
{
    "version": "0",
    "id": "event-ID",
    "detail-type": "FTP Server File Upload Completed",
    "source": "aws.transfer",
    "account": "958412138249",
    "time": "2024-01-29T16:31:43Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:transfer:us-east-1:958412138249:server/s-1111aaaa2222bbbb3"
    ],
    "detail": {
        "status-code": "COMPLETED",
        "protocol": "FTP",
        "bytes": 1048576,
        "client-ip": "10.0.0.141",
        "end-timestamp": "2024-01-29T16:31:43.311866408Z",
        "etag": "b6d81b360a5672d80c27430f39153e2c",
        "file-path": "/DOC-EXAMPLE-BUCKET/test/1mb_file",
        "server-id": "s-1111aaaa2222bbbb3",
        "username": "test",
        "session-id": "event-ID",
        "start-timestamp": "2024-01-29T16:31:42.462088327Z"
    }
}
```

SFTP 连接器事件

以下是 SFTP 连接器事件的详细信息字段：

- SFTP 连接器文件发送已完成
- SFTP 连接器文件发送失败
- SFTP 连接器文件检索已完成
- 检索 SFTP 连接器文件失败

下面包含source和detail-type字段，因为它们包含 Transfer Family 事件的特定值。有关所有事件中包含的其他元数据字段的定义，请参阅Amazon EventBridge 用户指南中的事件结构参考。

```
{
  . . .,
  "detail-type": "string",
  "source": "aws.transfer",
  . . .,
  "detail": {
    "operation" : "string",
    "connector-id" : "string",
    "transfer-id" : "string",
    "file-transfer-id" : "string",
    "url" : "string",
    "file-path" : "string",
    "status-code" : "string",
    "failure-code" : "string",
    "failure-message" : "string",
    "start-timestamp" : "string",
    "end-timestamp" : "string",
    "local-directory-path" : "string",
    "remote-directory-path" : "string"
    "bytes" : "number",
    "local-file-location" : {
      "domain" : "string",
      "bucket" : "string",
      "key" : "string"
    },
  }
}
```

`detail-type`

标识事件的类型。

对于此事件，该值是先前列出的 SFTP 连接器事件名称之一。

`source`

标识生成事件的服务。对于 Transfer Family 事件，此值为`aws.transfer`。

`detail`

包含关于事件信息的 JSON 对象。生成事件的服务决定了该字段的内容。

对于此事件，数据包括以下内容：

operation

StartFileTransfer请求是发送文件还是检索文件。价值观：SEND|RETRIEVE。

connector-id

正在使用的 SFTP 连接器的唯一标识符。

transfer-id

传输事件（StartFileTransfer请求）的唯一标识符。

file-transfer-id

正在传输的文件的唯一标识符。

url

合作伙伴的 AS2 或 SFTP 端点的 URL。

file-path

正在发送或检索的位置和文件。

status-code

传输是否成功。价值观：FAILED | COMPLETED。

failure-code

对于失败的传输，则为转移失败的原因代码。

failure-message

对于失败的传输，提供传输失败原因的详细信息。

start-timestamp

对于成功传输，指文件处理开始时间的时间戳。

end-timestamp

对于成功传输，指文件处理完成的时间戳。

local-directory-path

对于RETRIEVE请求，指存放检索到文件的位置。

`remote-directory-path`

对于SEND请求，指将文件放在合作伙伴的 SFTP 服务器上的文件目录。这是用户传递给StartFileTransfer请求RemoteDirectoryPath的的值。您可以在合作伙伴的 SFTP 服务器上指定默认目录。如果是，则此字段为空。

`bytes`

正在传输的字节数。对于失败的传输，该值为 0。

`local-file-location`

此参数包含 AWS 存储文件位置的详细信息。

`domain`

正在使用的存储空间。目前，唯一的值是S3。

`bucket`

Amazon S3 中对象的容器。

`key`

在 Amazon S3 中为对象分配的名称。

Example SFTP 连接器文件发送失败示例事件

以下示例显示了尝试向远程 SFTP 服务器发送文件时 SFTP 连接器出现故障的事件。

```
{
    "version": "0",
    "id": "event-ID",
    "detail-type": "SFTP Connector File Send Failed",
    "source": "aws.transfer",
    "account": "123456789012",
    "time": "2024-01-24T19:30:45Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:transfer:us-east-1:123456789012:connector/c-f1111aaaa2222bbbb3"
    ],
    "detail": {
        "operation": "SEND",
        "connector-id": "c-f1111aaaa2222bbbb3",
        "transfer-id": "transfer-ID",
```

```
        "file-transfer-id": "file-transfer-ID",
        "url": "sftp://s-21a23456789012a.server.transfer.us-east-1.amazonaws.com",
        "file-path": "/DOC-EXAMPLE-BUCKET/testfile.txt",
        "status-code": "FAILED",
        "failure-code": "CONNECTION_ERROR",
        "failure-message": "Unknown Host",
        "remote-directory-path": "",
        "bytes": 0,
        "start-timestamp": "2024-01-24T18:29:33.658729Z",
        "end-timestamp": "2024-01-24T18:29:33.993196Z",
        "local-file-location": {
            "domain": "S3",
            "bucket": "DOC-EXAMPLE-BUCKET",
            "key": "testfile.txt"
        }
    }
}
```

Example SFTP 连接器文件检索已完成示例事件

以下示例显示了一个事件，其中 SFTP 连接器成功检索了从远程 SFTP 服务器发送的文件。

```
{
    "version": "0",
    "id": "event-ID",
    "detail-type": "SFTP Connector File Retrieve Completed",
    "source": "aws.transfer",
    "account": "123456789012",
    "time": "2024-01-24T18:28:08Z",
    "region": "us-east-1",
    "resources": [
        "arn:aws:transfer:us-east-1:123456789012:connector/c-f1111aaaa2222bbbb3"
    ],
    "detail": {
        "operation": "RETRIEVE",
        "connector-id": "c-fc68000012345aa18",
        "transfer-id": "file-transfer-ID",
        "file-transfer-id": "file-transfer-ID",
        "url": "sftp://s-21a23456789012a.server.transfer.us-east-1.amazonaws.com",
        "file-path": "testfile.txt",
        "status-code": "COMPLETED",
        "local-directory-path": "/DOC-EXAMPLE-BUCKET",
        "bytes": 63533,
        "start-timestamp": "2024-01-24T18:28:07.632388Z",
```

```
        "end-timestamp": "2024-01-24T18:28:07.774898Z",
        "local-file-location": {
            "domain": "S3",
            "bucket": "DOC-EXAMPLE-BUCKET",
            "key": "testfile.txt"
        }
    }
}
```

AS2 赛事

以下是 AS2 事件的详细信息字段：

- AS2 有效载荷接收已完成
- AS2 有效载荷接收失败
- AS2 有效载荷发送已完成
- AS2 有效负载发送失败
- AS2 MDN 接收已完成
- AS2 MDN 接收失败
- AS2 MDN 发送已完成
- AS2 MDN 发送失败

下面包含`source`和`detail-type`字段，因为它们包含 Transfer Family 事件的特定值。有关所有事件中包含的其他元数据字段的定义，请参阅Amazon EventBridge 用户指南中的事件结构参考。

```
{
   . . .,
  "detail-type": "string",
  "source": "aws.transfer",
  . . .,
  "detail": {
    "s3-attributes" : {
      "file-bucket" : "string",
      "file-key" : "string",
      "json-bucket" : "string",
      "json-key" : "string",
      "mdn-bucket" : "string",
      "mdn-key" : "string"
```

```
      }
    "mdn-subject" : "string",
    "mdn-message-id" : "string",
    "disposition" : "string",
    "bytes" : "number",
    "as2-from" : "string",
    "as2-message-id" : "string",
    "as2-to" : "string",
    "connector-id" : "string",
    "client-ip" : "string",
    "agreement-id" : "string",
    "server-id" : "string",
    "requester-file-name" : "string",
    "message-subject" : "string",
    "start-timestamp" : "string",
    "end-timestamp" : "string",
    "status-code" : "string",
    "failure-code" : "string",
    "failure-message" : "string",
    "transfer-id" : "string"
  }
}
```

`detail-type`

标识事件的类型。

对于此事件，该值是先前列出的 AS2 事件之一。

`source`

标识生成事件的服务。对于 Transfer Family 事件，此值为`aws.transfer`。

`detail`

包含关于事件信息的 JSON 对象。生成事件的服务决定该字段的内容。

`s3-attributes`

识别正在传输的文件的 Amazon S3 存储桶和密钥。对于 MDN 事件，它还会识别 MDN 文件的存储桶和密钥。

`file-bucket`

Amazon S3 中对象的容器。

`file-key`

在 Amazon S3 中为对象分配的名称。

`json-bucket`

对于已完成或失败的传输，为 JSON 文件的容器。

`json-key`

对于已完成或失败的传输，指在 Amazon S3 中分配给 JSON 文件的名称。

`mdn-bucket`

对于 MDN 事件，是 MDN 文件的容器。

`mdn-key`

对于 MDN 事件，指在 Amazon S3 中分配给 MDN 文件的名称。

`mdn-subject`

对于 MDN 事件，是消息处置的文本描述。

`mdn-message-id`

对于 MDN 事件，这是 MDN 消息的唯一 ID。

`disposition`

对于 MDN 事件，指处置类别。

`bytes`

消息中的字节数。

`as2-from`

发送消息的 AS2 贸易伙伴。

`as2-message-id`

正在传输的 AS2 消息的唯一标识符。

`as2-to`

正在接收消息的 AS2 贸易伙伴。

`connector-id`

对于从 Transfer Family 服务器发送给贸易伙伴的 AS2 消息，使用的是 AS2 连接器的唯一标识符。

`client-ip`

对于服务器事件（从交易伙伴向 Transfer Family 服务器转账），是指参与转移的客户的 IP 地址。

`agreement-id`

对于服务器事件，AS2 协议的唯一标识符。

`server-id`

对于服务器事件，仅适用于 Transfer Family 服务器的唯一 ID。

`requester-file-name`

对于负载事件，指传输期间收到的文件的原始名称。

`message-subject`

消息主题的文字描述。

`start-timestamp`

对于成功传输，指文件处理开始时间的时间戳。

`end-timestamp`

对于成功传输，指文件处理完成的时间戳。

`status-code`

与 AS2 邮件传输过程状态相对应的代码。有效值：`COMPLETED | FAILED | PROCESSING`。

`failure-code`

对于失败的传输，指传输失败原因的类别。

`failure-message`

对于失败的传输，提供传输失败原因的详细信息。

`transfer-id`

转账事件的唯一标识符。

Example AS2 Payload 接收已完成示例事件

```
{
    "version": "0",
     "id": "event-ID",
    "detail-type": "AS2 Payload Receive Completed",
    "source": "aws.transfer",
    "account": "076722215406",
    "time": "2024-02-07T06:47:05Z",
    "region": "us-east-1",
    "resources": ["arn:aws:transfer:us-east-1:076722215406:connector/
c-1111aaaa2222bbbb3"],
    "detail": {
        "s3-attributes": {
            "file-key": "/inbound/processed/testAs2Message.dat",
            "file-bucket": "DOC-EXAMPLE-BUCKET"
        },
        "client-ip": "client-IP-address",
        "requester-file-name": "testAs2MessageVerifyFile.dat",
        "end-timestamp": "2024-02-07T06:47:06.040031Z",
        "as2-from": "as2-from-ID",
        "as2-message-id": "as2-message-ID",
        "message-subject": "Message from AS2 tests",
        "start-timestamp": "2024-02-07T06:47:05.410Z",
        "status-code": "PROCESSING",
        "bytes": 63,
        "as2-to": "as2-to-ID",
        "agreement-id": "a-1111aaaa2222bbbb3",
        "server-id": "s-1234abcd5678efghi"
    }
}
```

Example AS2 MDN 接收失败示例事件

```
{
  "version": "0",
  "id": "event-ID",
  "detail-type": "AS2 MDN Receive Failed",
  "source": "aws.transfer",
  "account": "889901007463",
  "time": "2024-02-06T22:05:09Z",
  "region": "us-east-1",
  "resources": ["arn:aws:transfer:us-east-1:076722215406:server/s-1111aaaa2222bbbb3"],
```

```
  "detail": {
      "mdn-subject": "Your Requested MDN Response re: Test run from Id 123456789abcde
 to partner ijklmnop987654",
      "s3-attributes": {
          "json-bucket": "DOC-EXAMPLE-BUCKET1",
          "file-key": "/as2Integ/TestOutboundWrongCert.dat",
          "file-bucket": "DOC-EXAMPLE-BUCKET2",
          "json-key": "/as2Integ/failed/TestOutboundWrongCert.dat.json"
      },
      "mdn-message-id": "MDN-message-ID",
      "end-timestamp": "2024-02-06T22:05:09.479878Z",
      "as2-from": "PartnerA",
      "as2-message-id": "as2-message-ID",
      "connector-id": "c-1234abcd5678efghj",
      "message-subject": "Test run from Id 123456789abcde to partner ijklmnop987654",
      "start-timestamp": "2024-02-06T22:05:03Z",
      "failure-code": "VERIFICATION_FAILED_NO_MATCHING_KEY_FOUND",
      "status-code": "FAILED",
      "as2-to": "MyCompany",
      "failure-message": "No public certificate matching message signature could be
 found in profile: p-1234abcd5678efghj",
      "transfer-id": "transfer-ID"
  }
}
```

安全性 AWS Transfer Family

云安全 AWS 是重中之重。作为 AWS 客户，您可以受益于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。

安全是双方共同承担 AWS 的责任。责任共担模式将其描述为云的 安全性和云中 的安全性：

要了解是否属于特定合规计划的范围，请参阅AWS 服务 “按合规计划划分的范围” ”，然后选择您感兴趣的合规计划。 AWS 服务 有关一般信息，请参阅AWS 合规计划AWS。

您可以使用下载第三方审计报告 AWS Artifact。有关更多信息，请参阅中的 “下载报告” 中的 “ AWS Artifact。

您在使用 AWS 服务 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。 AWS 提供了以下资源来帮助实现合规性：

- 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，并提供了部署以安全性和合规性为重点 AWS 的基准环境的步骤。
- 在 A@@ mazon Web Services 上构建 HIPAA 安全与合规架构 — 本白皮书描述了各公司如何使用 AWS 来创建符合 HIPAA 资格的应用程序。

> **Note**
>
> 并非所有 AWS 服务 人都符合 HIPAA 资格。有关更多信息，请参阅符合 HIPAA 要求的服务参考。

- AWS 合规资源AWS — 此工作簿和指南集可能适用于您所在的行业和所在地区。
- AWS 客户合规指南 — 从合规角度了解责任共担模式。这些指南总结了保护的最佳实践， AWS 服务并将指南映射到跨多个框架（包括美国国家标准与技术研究院 (NIST)、支付卡行业安全标准委员会 (PCI) 和国际标准化组织 (ISO)）的安全控制。
- 使用AWS Config 开发人员指南中的规则评估资源 — 该 AWS Config 服务评估您的资源配置在多大程度上符合内部实践、行业准则和法规。
- AWS Security Hub— 这 AWS 服务 提供了您内部安全状态的全面视图 AWS。Security Hub 通过安全控件评估您的 AWS 资源并检查其是否符合安全行业标准和最佳实践。有关受支持服务及控件的列表，请参阅 Security Hub 控件参考。
- AWS Audit Manager— 这 AWS 服务 可以帮助您持续审计 AWS 使用情况，从而简化风险管理以及对法规和行业标准的合规性。

本文档可帮助您了解在使用时如何应用分担责任模型 AWS Transfer Family。以下主题向您介绍如何进行配置 AWS Transfer Family 以满足您的安全和合规性目标。您还将学习如何使用其他 AWS 服务来帮助您监控和保护您的 AWS Transfer Family 资源。

我们提供一个研讨会，提供规范性指导和动手实验，介绍如何在无需修改现有应用程序或管理服务器基础架构 AWS 的情况下构建可扩展且安全的文件传输架构。您可以在此处查看本次研讨会的详细信息。

主题

AWS Transfer Family 服务器的安全策略

中的服务器安全策略 AWS Transfer Family 允许您限制与服务器关联的一组加密算法（消息身份验证码 (MAC)、密钥交换 (KEX) 和密码套件）。有关支持的密钥算法的列表，请参阅 加密算法。有关支持的服务器主机秘钥和服务托管用户秘钥算法列表，请参见所支持的用户和服务器密钥算法。

> **Note**
>
> 我们强烈建议将您的服务器更新为我们的最新安全政策。我们最新的安全策略是默认的。任何使用默认安全策略创建 Transfer Family 服务器 CloudFormation并接受默认安全策略的客户都将自动获得最新策略。如果您担心客户端兼容性，请明确说明在创建或更新服务器时您希望使用哪种安全策略，而不是使用默认策略，默认策略可能会发生变化。
> 要更改服务器的安全策略，请参阅编辑安全策略。

有关 Transfer Family 安全性的更多信息，请参阅博客文章《Transfer F amily 如何帮助您构建安全、合规的托管文件传输解决方案》。

主题

> **Note**
>
> TransferSecurityPolicy-2024-01是使用控制台、API 或 CLI 创建服务器时附加到服务器的默认安全策略。

加密算法

对于主机密钥，我们支持以下算法：

- rsa-sha2-256
- rsa-sha2-512
- ecdsa-sha2-nistp256
- ecdsa-sha2-nistp384
- ecdsa-sha2-nistp521
- ssh-ed25519

此外，2018 年和 2020 年的安全政策还允许ssh-rsa。

> **Note**
>
> 了解 RSA 密钥类型（始终是）和 RSA 主机密钥算法（可以是任何支持的算法`ssh-rsa`）之间的区别非常重要。

以下是各种安全策略支持的加密算法列表。

> **Note**
>
> 在下表和策略中，请注意算法类型的以下用法。
>
> - SFTP 服务器仅使用SshCiphersSshKexs、和SshMacs部分中的算法。
> - FTPS 服务器仅使用该TlsCiphers部分中的算法。
> - 由于FTP服务器不使用加密，因此不使用任何这些算法。

安全策略	2024-01	2023-05	2022-03	2020-06	FIPS-2024-01	FIPS-2023-05	FIPS-2020-06	2018-11
SshCiphers								
aes128-ctr	♦			♦	♦		♦	♦
aes128-gcm@openssh.com	♦	♦	♦	♦	♦	♦	♦	♦
aes192-ctr	♦	♦	♦	♦	♦	♦	♦	♦
aes256-ctr	♦	♦	♦	♦	♦	♦	♦	♦
aes256-gc	♦	♦	♦	♦	♦	♦	♦	♦

安全策略	2024-01	2023-05	2022-03	2020-06	FIPS-2024-01	FIPS-2023-05	FIPS-2020-06	2018-11
m@openssh.com								
chacha20-poly1305@openssh.com				♦				♦
SshKexs								
curve25519-sha256	♦	♦	♦					♦
curve25519-sha256@libssh.org	♦	♦	♦					♦
diffie-hellman-group14-sha1								♦
diffie-hellman-group14-sha256				♦			♦	♦

安全策略	2024-01	2023-05	2022-03	2020-06	FIPS-2024-01	FIPS-2023-05	FIPS-2020-06	2018-11
diffie-hellman-group16-sha512	♦	♦	♦	♦	♦	♦	♦	♦
diffie-hellman-group18-sha512	♦	♦	♦	♦	♦	♦	♦	♦
diffie-hellman-group-exchange-sha256		♦	♦	♦		♦	♦	♦
ecdh-nist p256-kyber-512r3-sha256-d00 @openquantumsafe.org	♦				♦			

安全策略	2024-01	2023-05	2022-03	2020-06	FIPS-2024-01	FIPS-2023-05	FIPS-2020-06	2018-11
ecdh-nistp384-kyber-768r3-sha384-d00@openquantumsafe.org	♦				♦			
ecdh-nistp521-kyber-1024r3-sha512-d00@openquantumsafe.org	♦				♦			
ecdh-sha2-nistp256	♦			♦	♦		♦	♦
ecdh-sha2-nistp384	♦			♦	♦		♦	♦
ecdh-sha2-nistp521	♦			♦	♦		♦	♦

安全策略	2024-01	2023-05	2022-03	2020-06	FIPS-2024-01	FIPS-2023-05	FIPS-2020-06	2018-11
x25519-kyber-512r3-sha256-d00@amazon.com	♦							
SshMacs								
hmac-sha1								♦
hmac-sha1-etm@openssh.com								♦
hmac-sha2-256			♦	♦			♦	♦
hmac-sha2-256-etm@openssh.com	♦	♦	♦	♦	♦	♦	♦	♦
hmac-sha2-512			♦	♦			♦	♦

安全策略	2024-01	2023-05	2022-03	2020-06	FIPS-2024-01	FIPS-2023-05	FIPS-2020-06	2018-11
hmac-sha2-512-etm@openssh.com	♦	♦	♦	♦	♦	♦	♦	♦
umac-128-etm@openssh.com				♦				♦
umac-128@openssh.com				♦				♦
umac-64-etm@openssh.com								♦
umac-64@openssh.com								♦
TlsCiphers								
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	♦	♦	♦	♦	♦	♦	♦	♦

安全策略	2024-01	2023-05	2022-03	2020-06	FIPS-2024-01	FIPS-2023-05	FIPS-2020-06	2018-11
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	♦	♦	♦	♦	♦	♦	♦	
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	♦	♦	♦	♦	♦	♦	♦	
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	♦	♦	♦	♦	♦	♦	♦	
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	♦	♦	♦	♦	♦	♦	♦	
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	♦	♦	♦	♦	♦	♦	♦	
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	♦	♦	♦	♦	♦	♦	♦	

安全策略	2024-01	2023-05	2022-03	2020-06	FIPS-2024-01	FIPS-2023-05	FIPS-2020-06	2018-11
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	♦	♦	♦	♦	♦	♦	♦	♦
TLS_RSA_WITH_AES_128_CBC_SHA256								♦
TLS_RSA_WITH_AES_256_CBC_SHA256								♦

TransferSecurityPolicy-2024-01

以下显示了 TransferSecurityPolicy -2024-01 安全策略。

```
{
    "SecurityPolicy": {
        "Fips": false,
        "SecurityPolicyName": "TransferSecurityPolicy-2024-01",
        "SshCiphers": [
            "aes128-gcm@openssh.com",
            "aes256-gcm@openssh.com",
            "aes128-ctr",
            "aes256-ctr",
            "aes192-ctr"
        ],
        "SshKexs": [
            "ecdh-nistp384-kyber-768r3-sha384-d00@openquantumsafe.org",
            "x25519-kyber-512r3-sha256-d00@amazon.com",
            "ecdh-nistp256-kyber-512r3-sha256-d00@openquantumsafe.org",
            "ecdh-nistp521-kyber-1024r3-sha512-d00@openquantumsafe.org",
            "ecdh-sha2-nistp256",
```

```
            "ecdh-sha2-nistp384",
            "ecdh-sha2-nistp521",
            "curve25519-sha256",
            "curve25519-sha256@libssh.org",
            "diffie-hellman-group18-sha512",
            "diffie-hellman-group16-sha512",
            "diffie-hellman-group-exchange-sha256"
        ],
        "SshMacs": [
            "hmac-sha2-256-etm@openssh.com",
            "hmac-sha2-512-etm@openssh.com"
        ],
        "TlsCiphers": [
            "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384"
        ]
    }
}
```

TransferSecurityPolicy-2023-05

以下显示了 TransferSecurityPolicy -2023-05 安全策略。

```
{
    "SecurityPolicy": {
        "Fips": false,
        "SecurityPolicyName": "TransferSecurityPolicy-2023-05",
        "SshCiphers": [
            "aes256-gcm@openssh.com",
            "aes128-gcm@openssh.com",
            "aes256-ctr",
            "aes192-ctr"
        ],
        "SshKexs": [
            "curve25519-sha256",
            "curve25519-sha256@libssh.org",
            "diffie-hellman-group16-sha512",
```

```
            "diffie-hellman-group18-sha512",
            "diffie-hellman-group-exchange-sha256"
        ],
        "SshMacs": [
            "hmac-sha2-512-etm@openssh.com",
            "hmac-sha2-256-etm@openssh.com"
        ],
        "TlsCiphers": [
            "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384"
        ]
    }
}
```

TransferSecurityPolicy-2022-03

以下显示了 TransferSecurityPolicy -2022-03 安全策略。

```
{
  "SecurityPolicy": {
    "Fips": false,
    "SecurityPolicyName": "TransferSecurityPolicy-2022-03",
    "SshCiphers": [
      "aes256-gcm@openssh.com",
      "aes128-gcm@openssh.com",
      "aes256-ctr",
      "aes192-ctr"
    ],
    "SshKexs": [
      "curve25519-sha256",
      "curve25519-sha256@libssh.org",
      "diffie-hellman-group16-sha512",
      "diffie-hellman-group18-sha512",
      "diffie-hellman-group-exchange-sha256"
    ],
    "SshMacs": [
      "hmac-sha2-512-etm@openssh.com",
```

```
      "hmac-sha2-256-etm@openssh.com",
      "hmac-sha2-512",
      "hmac-sha2-256"
    ],
    "TlsCiphers": [
      "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
      "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
      "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256",
      "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
      "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
      "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
      "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384",
      "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384"
    ]
  }
}
```

TransferSecurityPolicy-2020-06

以下显示了 TransferSecurityPolicy -2020-06 安全策略。

```
{
  "SecurityPolicy": {
    "Fips": false,
    "SecurityPolicyName": "TransferSecurityPolicy-2020-06",
    "SshCiphers": [
      "chacha20-poly1305@openssh.com",
      "aes128-ctr",
      "aes192-ctr",
      "aes256-ctr",
      "aes128-gcm@openssh.com",
      "aes256-gcm@openssh.com"
    ],
    "SshKexs": [
      "ecdh-sha2-nistp256",
      "ecdh-sha2-nistp384",
      "ecdh-sha2-nistp521",
      "diffie-hellman-group-exchange-sha256",
      "diffie-hellman-group16-sha512",
      "diffie-hellman-group18-sha512",
      "diffie-hellman-group14-sha256"
    ],
    "SshMacs": [
```

```
      "umac-128-etm@openssh.com",
      "hmac-sha2-256-etm@openssh.com",
      "hmac-sha2-512-etm@openssh.com",
      "umac-128@openssh.com",
      "hmac-sha2-256",
      "hmac-sha2-512"
    ],
    "TlsCiphers": [
      "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
      "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
      "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256",
      "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
      "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
      "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
      "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384",
      "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384"
    ]
  }
}
```

TransferSecurityPolicy-2018-11

以下显示了 TransferSecurityPolicy -2018-11 的安全策略。

```
{
  "SecurityPolicy": {
    "Fips": false,
    "SecurityPolicyName": "TransferSecurityPolicy-2018-11",
    "SshCiphers": [
      "chacha20-poly1305@openssh.com",
      "aes128-ctr",
      "aes192-ctr",
      "aes256-ctr",
      "aes128-gcm@openssh.com",
      "aes256-gcm@openssh.com"
    ],
    "SshKexs": [
      "curve25519-sha256",
      "curve25519-sha256@libssh.org",
      "ecdh-sha2-nistp256",
      "ecdh-sha2-nistp384",
      "ecdh-sha2-nistp521",
      "diffie-hellman-group-exchange-sha256",
```

```
      "diffie-hellman-group16-sha512",
      "diffie-hellman-group18-sha512",
      "diffie-hellman-group14-sha256",
      "diffie-hellman-group14-sha1"
    ],
    "SshMacs": [
      "umac-64-etm@openssh.com",
      "umac-128-etm@openssh.com",
      "hmac-sha2-256-etm@openssh.com",
      "hmac-sha2-512-etm@openssh.com",
      "hmac-sha1-etm@openssh.com",
      "umac-64@openssh.com",
      "umac-128@openssh.com",
      "hmac-sha2-256",
      "hmac-sha2-512",
      "hmac-sha1"
    ],
    "TlsCiphers": [
      "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
      "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
      "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256",
      "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
      "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
      "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
      "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384",
      "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384",
      "TLS_RSA_WITH_AES_128_CBC_SHA256",
      "TLS_RSA_WITH_AES_256_CBC_SHA256"
    ]
  }
}
```

TransferSecurityPolicy-FIPS-2024-01

以下显示了 TransferSecurityPolicy-FIPS-2024-01 安全策略。

> **Note**
>
> FIPS 服务终端节点和 TransferSecurityPolicy-FIPS-2024-01 安全策略仅在某些地区可用。AWS 有关更多信息，请参阅《AWS 一般参考》中的 AWS Transfer Family 端点和配额。

```
{
    "SecurityPolicy": {
        "Fips": true,
        "SecurityPolicyName": "TransferSecurityPolicy-FIPS-2024-01",
        "SshCiphers": [
            "aes128-gcm@openssh.com",
            "aes256-gcm@openssh.com",
            "aes128-ctr",
            "aes256-ctr",
            "aes192-ctr"
        ],
        "SshKexs": [
            "ecdh-nistp384-kyber-768r3-sha384-d00@openquantumsafe.org",
            "ecdh-nistp256-kyber-512r3-sha256-d00@openquantumsafe.org",
            "ecdh-nistp521-kyber-1024r3-sha512-d00@openquantumsafe.org",
            "ecdh-sha2-nistp256",
            "ecdh-sha2-nistp384",
            "ecdh-sha2-nistp521",
            "diffie-hellman-group18-sha512",
            "diffie-hellman-group16-sha512",
            "diffie-hellman-group-exchange-sha256"
        ],
        "SshMacs": [
            "hmac-sha2-256-etm@openssh.com",
            "hmac-sha2-512-etm@openssh.com"
        ],
        "TlsCiphers": [
            "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384"
        ]
    }
}
```

TransferSecurityPolicy-FIPS-2023-05

FIPS 认证详情 AWS Transfer Family 可在以下网址找到 https://csrc.nist.gov/projects/cryptographic-module-validation-program/validated-modules/search/all

以下显示了 TransferSecurityPolicy-FIPS-2023-05 安全策略。

> **Note**
>
> FIPS 服务终端节点和 TransferSecurityPolicy-FIPS-2023-05 安全策略仅在某些地区可用。AWS 有关更多信息，请参阅《AWS 一般参考》中的 AWS Transfer Family 端点和配额。

```
{
    "SecurityPolicy": {
        "Fips": true,
        "SecurityPolicyName": "TransferSecurityPolicy-FIPS-2023-05",
        "SshCiphers": [
            "aes256-gcm@openssh.com",
            "aes128-gcm@openssh.com",
            "aes256-ctr",
            "aes192-ctr"
        ],
        "SshKexs": [
            "diffie-hellman-group16-sha512",
            "diffie-hellman-group18-sha512",
            "diffie-hellman-group-exchange-sha256"
        ],
        "SshMacs": [
            "hmac-sha2-256-etm@openssh.com",
            "hmac-sha2-512-etm@openssh.com"
        ],
        "TlsCiphers": [
            "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384"
        ]
```

```
    }
}
```

TransferSecurityPolicy-FIPS-2020-06

FIPS 认证详情 AWS Transfer Family 可在以下网址找到 https://csrc.nist.gov/projects/cryptographic-module-validation-program/validated-modules/search/all

以下显示了 TransferSecurityPolicy-FIPS-2020-06 安全策略。

> **Note**
>
> FIPS 服务终端节点和 TransferSecurityPolicy-FIPS-2020-06 安全策略仅在某些地区可用。AWS 有关更多信息，请参阅 中的 终端节点和配额。

```
{
  "SecurityPolicy": {
    "Fips": true,
    "SecurityPolicyName": "TransferSecurityPolicy-FIPS-2020-06",
    "SshCiphers": [
      "aes128-ctr",
      "aes192-ctr",
      "aes256-ctr",
      "aes128-gcm@openssh.com",
      "aes256-gcm@openssh.com"
    ],
    "SshKexs": [
      "ecdh-sha2-nistp256",
      "ecdh-sha2-nistp384",
      "ecdh-sha2-nistp521",
      "diffie-hellman-group-exchange-sha256",
      "diffie-hellman-group16-sha512",
      "diffie-hellman-group18-sha512",
      "diffie-hellman-group14-sha256"
    ],
    "SshMacs": [
      "hmac-sha2-256-etm@openssh.com",
      "hmac-sha2-512-etm@openssh.com",
      "hmac-sha2-256",
      "hmac-sha2-512"
```

```
    ],
    "TlsCiphers": [
      "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
      "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
      "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256",
      "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
      "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
      "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
      "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384",
      "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384"
    ]
  }
}
```

后量子安全策略

下表列出了 Transfer Family 后量子安全策略算法。有关此策略的详细描述，请参见使用混合后量子密钥交换 AWS Transfer Family。

在策略列表中，请执行以下操作：

安全策略	TransferSecurityPolicy-pq-ssh-Experimental-2023-04	TransferSecurityPolicy-pq-ssh-fips-Experimental-2023-04
SSH ciphers		
aes128-ctr		♦
aes128-gcm@openssh.com	♦	♦
aes192-ctr	♦	♦
aes256-ctr	♦	♦
aes256-gcm@openssh.com	♦	♦
KEXs		
ecdh-nistp256-kyber-512r3-sha256-d00@openquantumsafe.org	♦	♦

安全策略	TransferSecurityPolicy-pq-ssh-Experimental-2023-04	TransferSecurityPolicy-pq-ssh-fips-Experimental-2023-04
ecdh-nistp384-kyber-768r3-sha384-d00@openquantumsafe.org	♦	♦
ecdh-nistp521-kyber-1024r3-sha512-d00@openquantumsafe.org	♦	♦
x25519-kyber-512r3-sha256-d00@amazon.com	♦	
diffie-hellman-group14-sha256		♦
diffie-hellman-group16-sha512	♦	♦
diffie-hellman-group18-sha512	♦	♦
ecdh-sha2-nistp384		♦
ecdh-sha2-nistp521		♦
diffie-hellman-group-exchange-sha256	♦	♦
ecdh-sha2-nistp256		♦
curve25519-sha256@libssh.org	♦	
curve25519-sha256	♦	
MACs		
hmac-sha2-256-etm@openssh.com	♦	♦
hmac-sha2-256	♦	♦

安全策略	TransferSecurityPolicy-pq-ssh-Experimental-2023-04	TransferSecurityPolicy-pq-ssh-fips-Experimental-2023-04
hmac-sha2-512-etm@openssh.com	♦	♦
hmac-sha2-512	♦	♦
TLS ciphers		
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	♦	♦
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	♦	♦
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	♦	♦
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	♦	♦
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	♦	♦
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	♦	♦
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	♦	♦
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	♦	♦

TransferSecurityPolicy-pq-ssh-Experimental-2023-04

下图显示了 TransferSecurityPolicy-pq-ssh-experimental-2023-04 安全策略。

```
{
    "SecurityPolicy": {
```

```
        "Fips": false,
        "SecurityPolicyName": "TransferSecurityPolicy-PQ-SSH-Experimental-2023-04",
        "SshCiphers": [
            "aes256-gcm@openssh.com",
            "aes128-gcm@openssh.com",
            "aes256-ctr",
            "aes192-ctr"
        ],
        "SshKexs": [
            "ecdh-nistp384-kyber-768r3-sha384-d00@openquantumsafe.org",
            "x25519-kyber-512r3-sha256-d00@amazon.com",
            "ecdh-nistp256-kyber-512r3-sha256-d00@openquantumsafe.org",
            "ecdh-nistp521-kyber-1024r3-sha512-d00@openquantumsafe.org",
            "curve25519-sha256",
            "curve25519-sha256@libssh.org",
            "diffie-hellman-group16-sha512",
            "diffie-hellman-group18-sha512",
            "diffie-hellman-group-exchange-sha256"
        ],
        "SshMacs": [
            "hmac-sha2-512-etm@openssh.com",
            "hmac-sha2-256-etm@openssh.com",
            "hmac-sha2-512",
            "hmac-sha2-256"
        ],
        "TlsCiphers": [
            "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384"
        ]
    }
}
```

TransferSecurityPolicy-pq-ssh-fips-Experimental-2023-04

以下显示了 TransferSecurityPolicy-pq-ssh-fips-experimental-2023-04 安全策略。

```
{
```

```
    "SecurityPolicy": {
        "Fips": true,
        "SecurityPolicyName": "TransferSecurityPolicy-PQ-SSH-FIPS-
Experimental-2023-04",
        "SshCiphers": [
            "aes256-gcm@openssh.com",
            "aes128-gcm@openssh.com",
            "aes256-ctr",
            "aes192-ctr",
            "aes128-ctr"
        ],
        "SshKexs": [
            "ecdh-nistp384-kyber-768r3-sha384-d00@openquantumsafe.org",
            "ecdh-nistp256-kyber-512r3-sha256-d00@openquantumsafe.org",
            "ecdh-nistp521-kyber-1024r3-sha512-d00@openquantumsafe.org",
            "ecdh-sha2-nistp256",
            "ecdh-sha2-nistp384",
            "ecdh-sha2-nistp521",
            "diffie-hellman-group-exchange-sha256",
            "diffie-hellman-group16-sha512",
            "diffie-hellman-group18-sha512",
            "diffie-hellman-group14-sha256"
        ],
        "SshMacs": [
            "hmac-sha2-512-etm@openssh.com",
            "hmac-sha2-256-etm@openssh.com",
            "hmac-sha2-512",
            "hmac-sha2-256"
        ],
        "TlsCiphers": [
            "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384"
        ]
    }
}
```

AWS Transfer Family SFTP 连接器的安全策略

中的 SFTP 连接器安全策略 AWS Transfer Family 允许您限制与 SFTP 连接器关联的一组加密算法（消息身份验证码 (MAC)、密钥交换 (KEX) 和密码套件）。以下是每个 SFTP 连接器安全策略支持的加密算法列表。

> **Note**
>
> `TransferSFTPConnectorSecurityPolicy-2024-03`是应用于 SFTP 连接器的默认安全策略。

安全策略	TransfersFTP -2024-03 ConnectorSecurityPolicy	TransfersFTP -2023-07 ConnectorSecurityPolicy
Ciphers		
aes128-ctr		♦
aes128-gcm@openssh.com	♦	♦
aes192-ctr	♦	♦
aes256-ctr	♦	♦
aes256-gcm@openssh.com	♦	♦
Kexs		
curve25519-sha256	♦	♦
curve25519-sha256@libssh.org	♦	♦
diffie-hellman-group14-sha1		♦
diffie-hellman-group16-sha512	♦	♦
diffie-hellman-group18-sha512	♦	♦

安全策略	TransfersFTP -2024-03 ConnectorSecurityPolicy	TransfersFTP -2023-07 ConnectorSecurityPolicy
diffie-hellman-group-exchange-sha256	♦	♦
Macs		
hmac-sha2-512-etm@openssh.com	♦	♦
hmac-sha2-256-etm@openssh.com	♦	♦
hmac-sha2-512	♦	♦
hmac-sha2-256	♦	♦
hmac-sha1		♦
hmac-sha1-96		♦
Host Key Algorithms		
rsa-sha2-256	♦	♦
rsa-sha2-512	♦	♦
ecdsa-sha2-nistp256	♦	♦
ecdsa-sha2-nistp384	♦	♦
ecdsa-sha2-nistp521	♦	♦
ssh-rsa		♦

使用混合后量子密钥交换 AWS Transfer Family

AWS Transfer Family 支持安全外壳 (SSH) 协议的混合后量子密钥建立选项。之所以需要建立后量子密钥，是因为已经有可能记录网络流量并将其保存以备将来由量子计算机解密，这被称为攻击。store-now-harvest-later

当您连接至 Transfer Family，您可使用此选项，将在 Amazon Simple Storage Service (Amazon S3) 存储或 Amazon Elastic File System (Amazon EFS) 内外安全传输文件。SSH 中的后量子混合密钥创建引入了后量子密钥建立机制，该机制与经典的密钥交换算法结合使用。通过传统密码套件创建的 SSH 密钥可以免受当前技术的暴力攻击。但是，在未来大规模量子计算出现之后，预计传统加密依然无法保证安全。

如果您的组织需要使 Transfer Family 连接传输的数据保持长期机密性，在目前没有大规模后量子计算机的情况下，可考虑改用后量子密码技术。

为了保护当今加密的数据免受未来潜在的攻击， AWS 正在与密码学界一起开发抗量子算法或后量子算法。我们在 Transfer Family 中实施了混合后量子密钥交换密码套件，通过将传统加密算法与后量子算法相结合。

这些混合密码套件可以在大多数 区域中用于您的生产工作负载。不过，由于混合密码套件的性能特征及带宽要求与传统密钥交换机制的性能特征及带宽要求有所不同，我们建议您针对 API 调用开展测试。

在后量子密码学 安全博客文章中了解后量子密码的更多信息。

目录

关于 TLS 中的混合后量子密钥交换

Transfer Family 支持后量子混合密钥交换密码套件，后者同时使用经典的Elliptic Curve Diffie-Hellman (ECDH)密钥交换算法和 CRYSTALS Kyber。Kyber 是一种后量子公钥加密和密钥创建算法，美国国家标准与技术研究所 (NIST) 已将其指定为第一后量子密钥协议算法标准。

客户端和服务器仍进行 ECDH 密钥交换。此外，服务器将后量子共享密钥封装至客户端后量子 KEM 公钥，该公钥参见客户端的 SSH 密钥交换消息。该策略将经典密钥交换的高度保证与拟议的后量子密钥交换的安全性相结合，以帮助确保只要 ECDH 或后量子共享机密无法破解，握手就会受到保护。

后量子混合密钥创建如何在 Transfer Family 中运行

AWS 最近宣布支持在 SFTP 文件传输中进行后量子密钥交换。 AWS Transfer Family Transfer Family 使用 SFTP 和其他协议安全地将 business-to-business 文件传输扩展到 AWS 存储服务。SFTP 是 SSH 运行的文件传输协议 (FTP) 的更安全的版本。Transfer Family 的后量子密钥交换支持提高了 SFTP 传输数据的安全门槛。

Transfer Family 中对后量子混合密钥交换 SFTP 的支持包括：将后量子算法 Kyber-512、Kyber-768 和 Kyber-1024 与超过 P256、P384、P521 或 Curve25519 曲线的 ECDH 相结合。后量子混合 SSH 秘钥交换草稿中指定以下对应的 SSH 秘钥交换方法。

- `ecdh-nistp256-kyber-512r3-sha256-d00@openquantumsafe.org`
- `ecdh-nistp384-kyber-768r3-sha384-d00@openquantumsafe.org`
- `ecdh-nistp521-kyber-1024r3-sha512-d00@openquantumsafe.org`
- `x25519-kyber-512r3-sha256-d00@amazon.com`

> **Note**
>
> 随着草稿向标准化发展，或者当 NIST 批准 Kyber 算法时，这些新的密钥交换方法可能会发生变化。

为什么选择 Kyber？

AWS 致力于支持标准化、可互操作的算法。Kyber 是 NIST 后量子密码学项目选择的第一个标准化后量子加密算法。一些标准机构已经在将Kyber整合到协议中。 AWS 已在某些 AWS API 端点中支持 TLS 中的 Kyber。

作为该承诺的一部分， AWS 已向IETF提交了一份后量子密码学提案草案，该草案将Kyber与NIST批准的曲线（例如用于SSH的P256）相结合。为了帮助增强客户的安全性，在 SFTP 和 SSH 中 AWS 实施后量子密钥交换遵循了该草案。在我们的提案被 IETF 采纳并成为标准之前，我们计划支持未来更新。

随着草稿向标准化发展，或者当 NIST 批准 Kyber 算法时，这些新的密钥交换方法可能会发生变化。

> **Note**
>
> 后量子算法支持在 TLS 中用于后量子混合密钥交换 AWS KMS （参见将混合后量子 TLS 与 AWS KMS）和 AWS Secrets Manager API 端点交换。AWS Certificate Manager

后量子混合 SSH 密钥交换和加密要求 (FIPS 140)

对于需要符合 FIPS 标准的客户，Transfer Family 使用 FIPS 140 认证的开源加密库-LC 在 SSH 中提供 AWS FIPS 认可的加密。 AWS根据NIST的SP 800-56Cr2（第2节），Transfer Family中TransferSecurityPolicy-pq-ssh-Fips-Experimental-2023-04中支持的后量子混合密钥交换方法已获得FIPS的批准。德国联邦信息安全办公室 (BSI) 和法国国家信息系统安全局 (ANSSI) 也推荐了这种后量子混合密钥交换方法。

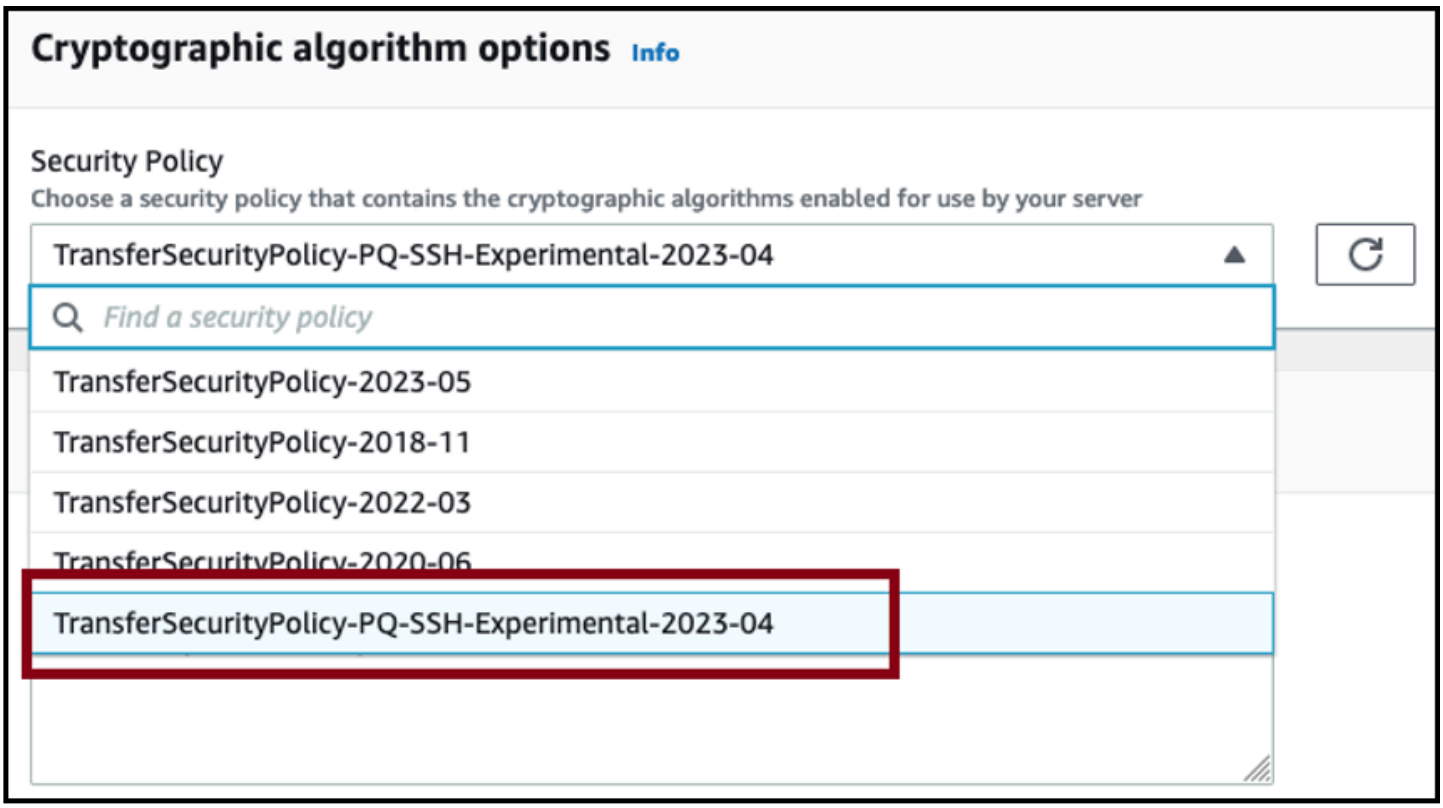
在 Transfer Family 中测试后量子混合密钥交换

本节介绍测试后量子混合密钥交换所需步骤。

1. 在 SFTP 端点启用后量子混合密钥交换.
2. 遵循上述规范草案中的指导，使用支持后量子混合密钥交换的 SFTP 客户端 (例如 设置支持后量子混合密钥交换的 SFTP 客户端)。
3. 通过 Transfer Family 服务器传输文件。
4. 确认 SFTP 中的后量子混合密钥交换.

在 SFTP 端点启用后量子混合密钥交换

当您在 Transfer Family 创建 SFTP 服务器端点时，您可选择 SSH 策略，或在现有 SFTP 端点编辑加密算法选项。以下快照显示了您更新 SSH 策略的 AWS Management Console 示例。



支持后量子密钥交换的 SSH 策略名称是-pq-ssh-experimental-2023-04 和 TransferSecurityPolicy-pq-ssh-fips-Experimental-2023-04。TransferSecurityPolicy有关 Transfer Family 政策的更多详情，请参阅 AWS Transfer Family 服务器的安全策略。

设置支持后量子混合密钥交换的 SFTP 客户端

在 SFTP Transfer Family 端点中选择正确的后量子 SSH 策略后，你可以在 Transfer Family 中尝试后量子 SFTP。遵循上述规范草案中的指导，使用支持后量子混合密钥交换的 SFTP 客户端 (例如)。

OQS OpenSSH 是 OpenSSH 的开源分支，通过使用`liboqs`在 SSH 中添加量子安全加密技术。`liboqs`是实现抗量子加密算法的开源 C 库。OQS OpenSSH 和`liboqs` 是开放量子安全 (OQS) 项目的一部分。

要通过 OQS OpenSSH 在 Transfer Family SFTP 中测试后量子混合密钥交换，你需要按照项目自述文件中的说明构建 OQS OpenSSH。构建 OQS OpenSSH 后，您可以运行示例 SFTP 客户端，按以下命令，使用后量子混合密钥交换方法连接至您的 SFTP 端点 (例如`s-1111aaaa2222bbbb3.server.transfer.us-west-2.amazonaws.com`)。

```
./sftp -S ./ssh -v -o \
   KexAlgorithms=ecdh-nistp384-kyber-768r3-sha384-d00@openquantumsafe.org \
```

```
    -i username_private_key_PEM_file \
    username@server-id.server.transfer.region-id.amazonaws.com
```

在下面的命令中，将 region 替换为您自己的信息。

- 将*username_private_key_PEM_file*更换为 SFTP 用户隐私秘钥 PEM 编码文件。
- 将 *username* 替换为该实例的用户名称。
- 将*server-id* 替换为 Transfer Family 服务器 ID
- 将 *region-id* 替换为您的 Transfer Family 服务器所在的实际区域

确认 SFTP 中的后量子混合密钥交换

要确认 SFTP 至 Transfer Family 的 SSH 连接期间是否使用了后量子混合密钥交换，请查看客户端输出。或者您可以使用数据包捕获程序。如果您使用 Open Quantum Safe OpenSSH 客户端，则应输出类似于以下内容（为简洁起见，省略不相关的信息）：

```
$./sftp -S ./ssh -v -o KexAlgorithms=ecdh-nistp384-kyber-768r3-sha384-
d00@openquantumsafe.org -
i username_private_key_PEM_file username@s-1111aaaa2222bbbb3.server.transfer.us-
west-2.amazonaws.com
OpenSSH_8.9-2022-01_p1, Open Quantum Safe 2022-08, OpenSSL 3.0.2 15 Mar 2022
debug1: Reading configuration data /home/lab/openssh/oqs-test/tmp/ssh_config
debug1: Authenticator provider $SSH_SK_PROVIDER did not resolve; disabling
debug1: Connecting to s-1111aaaa2222bbbb3.server.transfer.us-west-2.amazonaws.com
 [xx.yy.zz..12] port 22.
debug1: Connection established.
[...]
debug1: Local version string SSH-2.0-OpenSSH_8.9-2022-01_
debug1: Remote protocol version 2.0, remote software version AWS_SFTP_1.1
debug1: compat_banner: no match: AWS_SFTP_1.1
debug1: Authenticating to s-1111aaaa2222bbbb3.server.transfer.us-
west-2.amazonaws.com:22 as 'username'
debug1: load_hostkeys: fopen /home/lab/.ssh/known_hosts2: No such file or directory
[...]
debug1: SSH2_MSG_KEXINIT sent
debug1: SSH2_MSG_KEXINIT received
debug1: kex: algorithm: ecdh-nistp384-kyber-768r3-sha384-d00@openquantumsafe.org
debug1: kex: host key algorithm: ssh-ed25519
debug1: kex: server->client cipher: aes192-ctr MAC: hmac-sha2-256-etm@openssh.com
 compression: none
```

```
debug1: kex: client->server cipher: aes192-ctr MAC: hmac-sha2-256-etm@openssh.com
 compression: none
debug1: expecting SSH2_MSG_KEX_ECDH_REPLY
debug1: SSH2_MSG_KEX_ECDH_REPLY received
debug1: Server host key: ssh-ed25519 SHA256:e3b0c44298fc1c149afbf4c8996fb92427ae41e4649
[...]
debug1: rekey out after 4294967296 blocks
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug1: SSH2_MSG_NEWKEYS received
debug1: rekey in after 4294967296 blocks
[...]
Authenticated to AWS.Tranfer.PQ.SFTP.test-endpoint.aws.com ([xx.yy.zz..12]:22) using
 "publickey".s
debug1: channel 0: new [client-session]
[...]
Connected to s-1111aaaa2222bbbb3.server.transfer.us-west-2.amazonaws.com.
sftp>
```

输出显示使用后量子混合ecdh-nistp384-kyber-768r3-sha384-d00@openquantumsafe.org方法执行的、以及成功创建 SFTP 会话的客户端协商。

中的数据保护 AWS Transfer Family

AWS 分担责任模型分担责任模型适用于 AWS Transfer Family （ Transfer Family ）中的数据保护。如本模型所述 AWS ，负责保护运行所有 AWS 云的全球基础架构。您负责维护对托管在此基础设施上的内容的控制。此内容包括您使用的 AWS 服务的安全配置和管理任务。有关数据隐私的更多信息，请参阅数据隐私常见问题。有关欧洲数据保护的信息，请参阅 安全性博客 上的 责任共担模式和 GDPR 博客文章。

出于数据保护目的，我们建议您保护 AWS 账户凭证并使用 AWS Identity and Access Management (IAM) 设置个人用户账户。这仅向每个用户授予履行其工作职责所需的权限。我们还建议您通过以下方式保护数据：

- 对每个账户使用 multi-factor authentication （ MFA ）。
- 使用 SSL/TLS 与资源通信。 AWS 支持 TLS 1.2。
- 使用设置 API 和用户活动日志 AWS CloudTrail。
- 使用 AWS 加密解决方案以及 AWS 服务中的所有默认安全控制。
- 使用高级托管安全服务（例如 Amazon Macie ），它有助于发现和保护存储在 Amazon S3 中的个人数据。

- 如果在通过命令行界面或 API 访问 AWS 时需要经过 FIPS 140-2 验证的加密模块，请使用 FIPS 端点。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 140-2 版》。

我们强烈建议您切勿将敏感的可识别信息（例如您客户的账号）放入自由格式字段（例如名称字段）。这包括你使用控制台、API 或软件开发工具包使用 Tran AWS sfer Famil AWS CLI y 或其他 AWS 服务时。您输入至 Transfer Family 服务配置或其他服务配置中的数据可选择并纳入诊断日志。当您向外部服务器提供网址时，请勿在网址中包含凭证信息来验证您对该服务器的请求。

相比之下，来自 Transfer Family 服务器的上传和下载数据被视为完全私密，永远不会存在于SFTP 或 FTPS 连接等加密通道之外 。仅经过授权的人员才能访问这些数据。

主题

Amazon S3 中的数据加密

AWS Transfer Family 使用您为 Amazon S3 存储桶设置的默认加密选项来加密您的数据。如果对存储桶启用加密，则存储到该存储桶中的所有对象都会进行加密。这些对象使用服务器端加密，使用 Amazon S3 托管密钥 (SSE-S3) 或 () 托管密钥 AWS Key Management Service (SSE-KMS AWS KMS) 进行加密。有关服务器端加密的更多信息，请参阅《Amazon Simple Storage Service 用户指南》中的使用服务器端加密保护数据。

以下步骤向您展示了如何加密中的数据 AWS Transfer Family。

允许加密 AWS Transfer Family

1. 为 存储桶启用默认加密。有关更多详细信息，请参阅《Amazon Simple Storage Service 用户指南》中的适用于 S3 存储桶的 Amazon S3 默认加密。
2. 更新附加到用户的 AWS Identity and Access Management (IAM) 角色策略以授予所需的 AWS Key Management Service (AWS KMS) 权限。
3. 如果您为用户使用会话策略，则会话策略必须授予所需的 AWS KMS 权限。

以下示例显示了一个 IAM 策略，该策略授予与启用 AWS KMS 加密的 Amazon S3 存储桶 AWS Transfer Family 一起使用时所需的最低权限。可以将此示例策略包含在用户 IAM 角色策略和范围缩小策略（如果您在使用）中。

```
{
 "Sid": "Stmt1544140969635",
 "Action": [
  "kms:Decrypt",
  "kms:Encrypt",
  "kms:GenerateDataKey"
 ],
 "Effect": "Allow",
 "Resource": "arn:aws:kms:region:account-id:key/kms-key-id"
}
```

> **Note**
>
> 您在此策略中指定的 KMS 密钥 ID 必须与步骤 1 中为默认加密指定的密钥 ID 相同。AWS KMS 密钥策略中必须允许根角色或用户使用的 IAM 角色。有关 AWS KMS 密钥策略的信息，请参阅AWS Key Management Service 开发人员指南中的在 AWS KMS 中使用密钥策略。

密钥管理

在本节中，您可以找到有关 SSH 密钥的信息，包括如何生成密钥以及如何轮换的信息。有关使用 Transfer Family AWS Lambda 来管理密钥的详细信息，请参阅博客文章使用 A AWS Transfer Family 和启用用户自助服务密钥管理 AWS Lambda。

> **Note**
>
> AWS Transfer Family 接受 RSA、ECDSA 和 ED25519 密钥。

本节还介绍如何生成与管理 Pretty Good Privacy (PGP) 密钥。

主题

- 所支持的用户和服务器密钥算法
- 为服务托管用户生成 SSH 密钥
- 轮换 SSH 密钥
- 生成和管理 PGP 密钥

- 支持的 PGP 客户端

所支持的用户和服务器密钥算法

支持以下 AWS Transfer Family中的用户和服务器秘钥对算法。

> **Note**
>
> 有关在工作流程中与 PGP 解密配合使用的算法，请参阅PGP 秘钥对支持的算法。

- 对于 ED25519：`ssh-ed25519`
- 对于 RSA：
  - `rsa-sha2-256`
  - `rsa-sha2-512`
- 对于 ECDSA：
  - `ecdsa-sha2-nistp256`
  - `ecdsa-sha2-nistp384`
  - `ecdsa-sha2-nistp521`

> **Note**
>
> 我们根据旧版安全策略中的 SHA1 支持`ssh-rsa`。有关更多信息，请参阅 加密算法。

为服务托管用户生成 SSH 密钥

您可以设置 SFTP 服务器以使用服务管理的身份验证方法对用户进行身份验证，其中用户名和 SSH 密钥存储在服务中。用户的公有 SSH 密钥作为用户属性上传到 SFTP 服务器。服务器将此密钥用作密钥标准身份验证过程的一部分。每个用户均可使用单个服务器存档多个公有 SSH 密钥。有关每个用户可以存储的密钥数量限制，请参阅中的Amazon Web Services 一般参考中的AWS Transfer Family 端点和配额。

作为服务托管身份验证方法的替代方法，您可以使用自定义身份提供商对用户进行身份验证，或者 AWS Directory Service for Microsoft Active Directory。有关更多信息，请参阅使用自定义身份提供程序或使用 Di AWS rectory Service 身份提供商。

服务器只能使用一种方法（服务托管、目录服务或自定义身份提供程序）对用户进行身份验证，并且该方法在创建服务器后无法更改。

主题

在 macOS、Linux 或 Unix 系统创建 SSH 密钥

在 macOS、Linux 或 Unix 操作系统中，您可以使用`ssh-keygen`命令创建 SSH 公钥和 SSH 私钥（也称为密钥对）。

若要在 macOS、Linux 或 Unix 操作系统上创建 SSH 密钥

1. 在 macOS、Linux 或 Unix 操作系统，打开命令终端。
2. AWS Transfer Family 接受 RSA、ECDSA 和 ED25519 格式的密钥。根据您生成的密钥对类型选择相应的命令。

   > **Note**
   >
   > 在以下示例中，我们未指定密码：在这种情况下，该工具会要求您输入密码，然后重复密码进行验证。创建密码可以更好地保护您的私钥，还可以提高系统整体安全性。您无法恢复密码：如果您忘记了密码，则必须创建新的密钥。
   > 但是，如果要生成服务器主机密钥，则必须通过在命令中指定 `-N ""` 选项（或者在出现提示时按**`Enter`**两次）指定空密码，原因是 Transfer Family 服务器无法在启动时请求密码。

   - 生成 4096 位 RSA 密钥对。

     ```
     ssh-keygen -t rsa -b 4096 -f key_name
     ```

   - 若要生成 ECDSA 521 位密钥对（ECDSA 大小为 256、384 和 521），请执行以下操作：

     ```
     ssh-keygen -t ecdsa -b 521 -f key_name
     ```

   - 生成 ED25519 密钥对。

```
ssh-keygen -t ed25519 -f key_name
```

> **Note**
>
> *key_name* 是 SSH 密钥对文件名。

下面是此类输出的示例。

```
ssh-keygen -t rsa -b 4096 -f key_name
Generating public/private rsa key pair.

Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in key_name.
Your public key has been saved in key_name.pub.
The key fingerprint is:
SHA256:8tDDwPmanTFcEzjTwPGETVWOGW1nVz+gtCCE8hL7PrQ bob.amazon.com
The key's randomart image is:
+---[RSA 4096]----+
|    . ....E      |
| .   = ...       |
|. . . = ..o      |
| . o +  oo =     |
|  + =  .S.= *    |
| . o o ..B + o   |
|     .o.+.* .    |
|     =o*+*.      |
|    ..*o*+.      |
+----[SHA256]-----+
```

> **Note**
>
> 当您运行 ssh-keygen 命令时（如前所示），它将公有密钥和私有密钥创建为当前目录中的文件。

您的 SSH 密钥对现已准备就绪，可以使用。按照步骤 3 和 4 为服务托管用户存储 SSH 公钥。这些用户在 Transfer Family 服务器端点上传输文件时使用这些密钥。

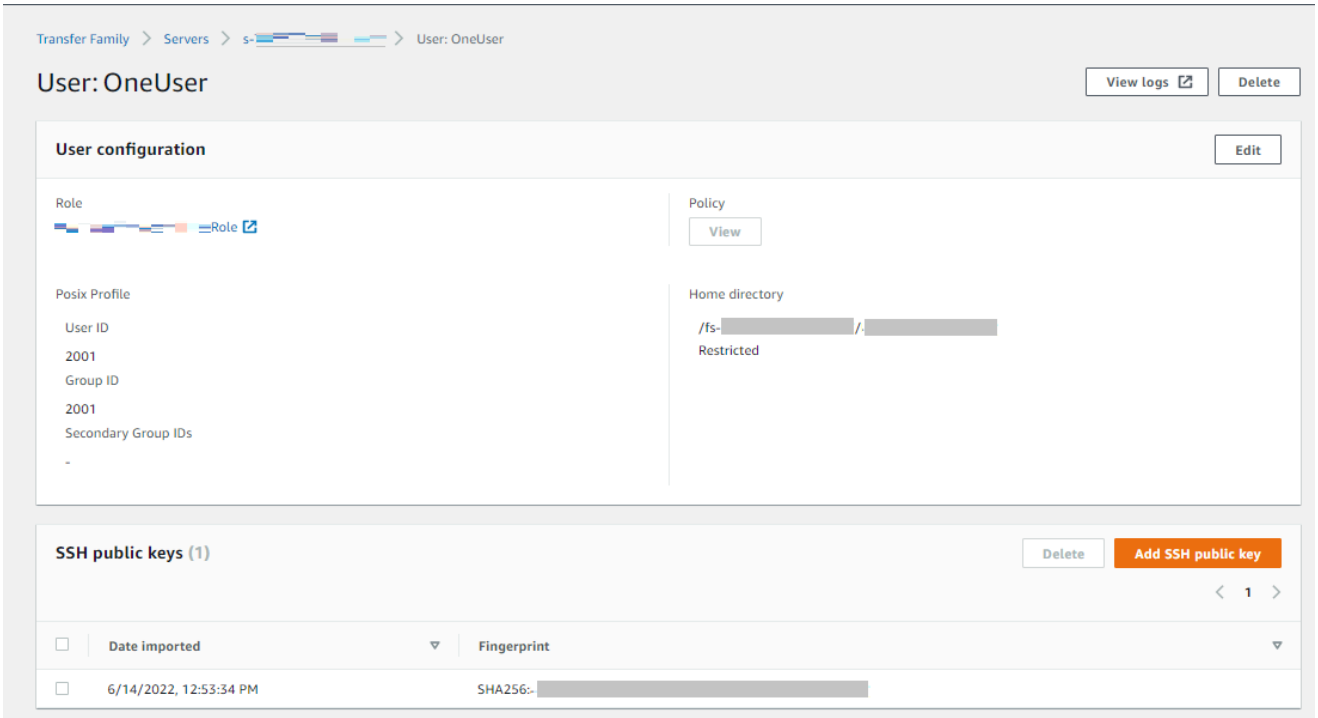
3. 导航到 *key_name*.pub，然后打开 README.md 文件。
4. 复制文本并将其粘贴至服务托管用户的SSH 公钥中。

   a. 通过 https://console.aws.amazon.com/transfer/ 打开 AWS Transfer Family 控制台，然后从导航窗格中选择 “服务器”。

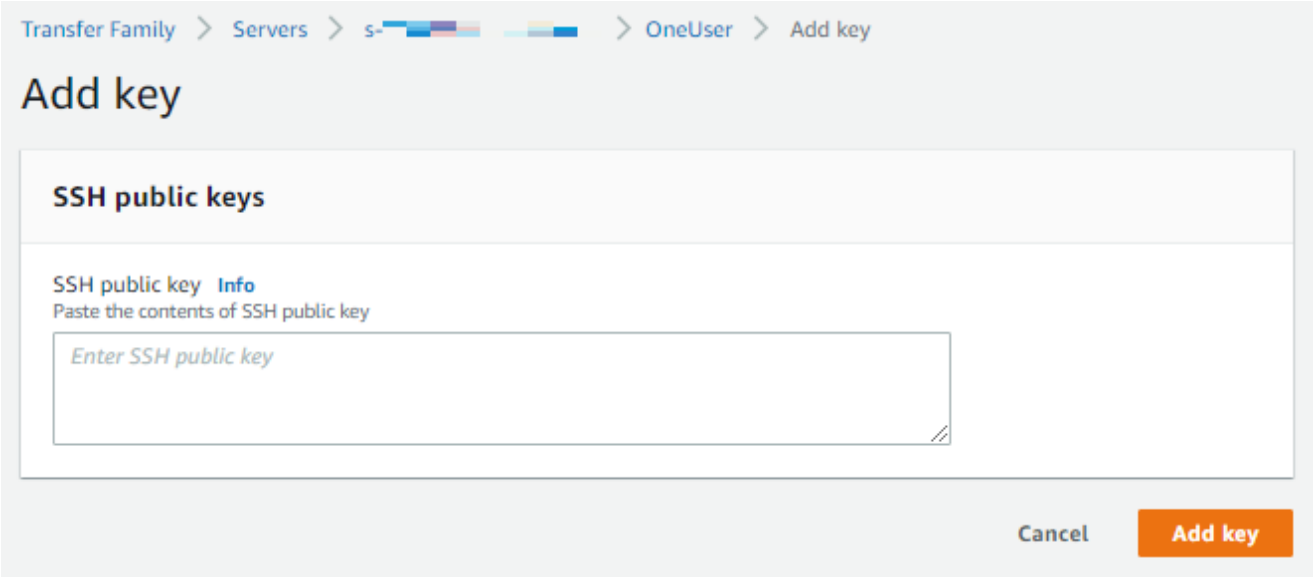
   b. 在服务器页面，选择包含要更新用户服务器的 服务器 ID。

   c. 选择要为其添加公钥的目标用户。

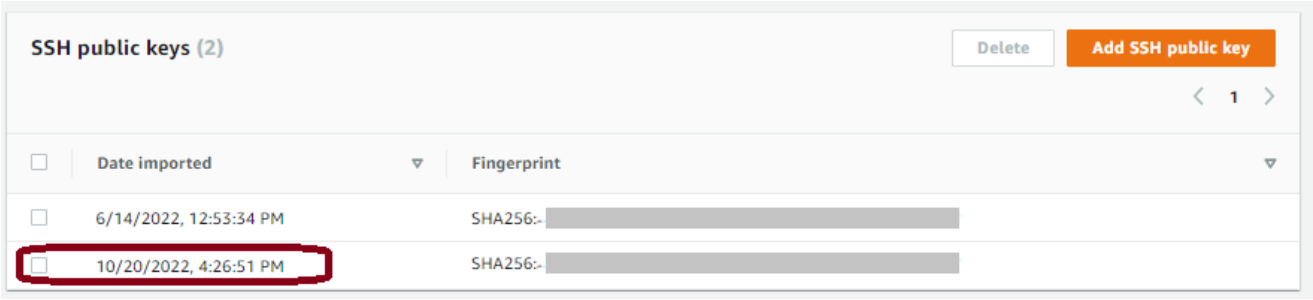
   d. 在 SSH 公钥窗格，选择添加 SSH 公钥。



   e. 将您生成的公钥文本粘贴至 SSH 公钥文本框中，然后选择添加密钥。



新密钥列于 SSH 公钥窗格。



在 Windows 上创建 SSH 密钥

Windows 使用略微不同的 SSH 密钥对格式。公有密钥必须采用 PUB 格式，私有密钥必须采用 PPK 格式。在 Windows 上，您可以使用 PuTTYgen 以适当的格式创建 SSH 密钥对。您还可以使用 PuTTYgen 将使用 `ssh-keygen` 生成的私有密钥转换为 `.ppk` 文件。

> **Note**
>
> 如果您向 WinSCP 提供的私有密钥文件不是 `.ppk` 格式，该 SFTP 客户端会为您将密钥转换为 `.ppk` 格式。

要查看有关在 Windows 上使用 PuTTYgen 创建 SSH 密钥的教程，请参阅 SSH.com 网站。

将 SSH2 公钥转换至 PEM 格式

AWS Transfer Family 仅接受 PEM 格式的公钥。如果您有 SSH2 公钥，需要对其进行转换。SSH2 公有密钥包含以下格式：

```
---- BEGIN SSH2 PUBLIC KEY ----
Comment: "rsa-key-20160402"
AAAAB3NzaC1yc2EAAAABJQAAAgEAiL0jjDdFqK/kYThqKt7THrjABTPWvXmB3URI
:
:
---- END SSH2 PUBLIC KEY ----
```

PEM 公有密钥包含以下格式：

```
ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAA...
```

运行以下命令，将 SSH2 格式的公钥转换为 PEM 格式的公钥。将*ssh2-key*替换为 SSH2 秘钥名称，将*PEM-key*替换为您的 PEM 秘钥名称。

```
ssh-keygen -i -f ssh2-key.pub > PEM-key.pub
```

轮换 SSH 密钥

我们推荐轮换 SSH 密钥的最佳安全实践。通常，此轮换被指定为安全策略的一部分，并以某种自动化的方式实现。根据安全级别，对于高度敏感的通信，SSH 密钥对可能只使用一次。这样做可以消除因存储密钥而导致的任何风险。但是，更常见的做法是将 SSH 凭证存储一段时间，并设置一个不会给 SFTP 用户带来过多负担的间隔。通常，时间间隔为 3 个月。

有两种方法用于执行 SSH 密钥轮换：

- 在控制台上，您可以上传新的 SSH 公钥和删除现有 SSH 公钥。
- 使用 API，您可以使用 AP DeleteSshPublicKeyI 删除用户的安全外壳 (SSH) 公钥，使用 ImportSshPublicKeyAPI 向用户账户添加新的安全外壳 (SSH) 公钥，从而更新现有用户。

Console

若要控制台中执行密钥轮换

1. 打开 AWS Transfer Family 控制台，网址为 https://console.aws.amazon.com/transfer/。

2. 导航至 服务器 页面。
3. 选择 Server ID (服务器 ID) 列中的标识符以查看 Server Configuration (服务器配置) 页面，如下所示。
4. 在 用户 下，选中要轮换其 SSH 公钥用户的复选框，然后选择操作，然后选择添加密钥以查看添加密钥页面。

   或者

   选择用户名以查看用户详细信息页面，然后选择添加 SSH 公钥 以查看添加密钥页面。
5. 输入新的 SSH 公钥并选择 添加密钥。

   > **Important**
   >
   > SSH 公有密钥格式取决于您生成的密钥的类型。
   >
   > - RSA 密钥的格式为 `ssh-rsa` *`string`*。
   > - ED25519 密钥的格式为 `ssh-ed25519` *`string`*。
   > - 对于 ECDSA 密钥，`ecdsa-sha2-nistp256` 字符串为`ecdsa-sha2-nistp384`、或，具体取决于您生成的密钥的大小。然后，先是*`string`*，后跟开头字符串，这与其他秘钥类型类似。

   您将返回 用户配置 屏幕，并且您刚刚上传的新 SSH 公有密钥将显示在 SSH 公有密钥部分中。
6. 选中要删除的身份旁边的复选框，然后选择删除。
7. 输入单词`delete`以确认删除操作，然后选择 删除。

API

若要使用 API 执行密钥轮换

1. 在 macOS、Linux 或 Unix 操作系统，打开命令终端。
2. 输入以下命令，以检索要删除的 SSH 密钥。若要使用此命令，请将*`serverID`*替换为您的 Transfer Family 服务器的服务器 ID，然后将*`username`*替换为您的用户名。

```
aws transfer describe-user --server-id='serverID' --user-name='username'
```

该命令返回有关此用户的详细信息。复制 "SshPublicKeyId": 字段的内容。您将需要稍后在此程序中输入此值。

```
"SshPublicKeys": [ { "SshPublicKeyBody": "public-key", "SshPublicKeyId":
 "keyID",
   "DateImported": 1621969331.072 } ],
```

3. 接下来，为您的用户导入新 SSH 密钥。在 提示符中，输入以下命令。若要使用此命令，请将*serverID*替换为您的 Transfer Family 服务器的服务器 ID，然后将替换为您的用户名。

```
aws transfer import-ssh-public-key --server-id='serverID' --user-name='username'
   --ssh-public-key-body='public-key'
```

如果命令成功，则不返回任何输出。

4. 最后通过运行以下命令删除旧密钥。若要使用此命令，将*serverID*替换为 Transfer Family 服务器的服务器 ID，将*username*替换为您的用户名，将*keyID-from-step-2*替换为您在此程序第 2 步中复制的秘钥 ID 值。

```
aws transfer delete-ssh-public-key --server-id='serverID' --user-name='username'
   --ssh-public-key-id='keyID-from-step-2'
```

5. （可选）要确认旧密钥是否存在，请重复第 2 步。

生成和管理 PGP 密钥

您可以对 Transfer Family 通过工作流程处理的文件使用 Pretty Good Privacy (PGP) 解密。要在工作流程步骤中使用解密，必须提供 PGP 密钥。

AWS 存储博客上有一篇文章描述了如何加密和解密文件，使用 PGP 加密和解密文件以及。 AWS Transfer Family

生成 PGP 密钥

生成 PGP 密钥所用的方法，取决于您的操作系统和所使用的密钥生成软件的版本。

如果您使用的是 Linux 或 Unix，请使用软件包安装程序安装gpg。根据您的 Linux 发行版，在以下选择适用于您的命令。

```
sudo yum install gnupg
```

```
sudo apt-get install gnupg
```

对于 Windows 或 macOS，你可以从https://gnupg.org/download/下载你需要的内容。

安装 PGP 密钥生成器软件后，运行 `gpg --full-gen-key` 或 `gpg --gen-key` 命令生成密钥对。

> **Note**
>
> 如果您使用的版本是GnuPG 2.3.0 或以上，则必须运行 `gpg --full-gen-key`。当提示输入要创建的密钥类型时，请选择 RSA 或 ECC。但是，如果您选择 ECC，请确保为椭圆曲线选择 NIST或BrainPool 请勿选择 。

PGP 密钥对支持的算法

PGP 密钥对支持的算法

- RSA
- Elgamal
- ECC：
  - NIST
  - BrainPool

> **Note**
>
> 不支持 Curve25519 秘钥。

有用的 **gpg** 子命令

以下是一些有用的gpg子命令：

- `gpg --help` — 此命令列出了可用选项，可能还包括一些示例。
- `gpg --list-keys` — 此命令列出了您创建的所有密钥对的详细信息。
- `gpg --fingerprint` — 此命令列出了所有密钥对的详细信息，包括每个密钥的指纹。
- `gpg --export -a` *`user-name`* — 此命令导出生成密钥时 *`user-name`* 使用密钥的公钥部分。

管理密钥对

要使用或管理 KMS 密钥，您必须使用 AWS Secrets Manager。

> **Note**
>
> 您的秘钥名称包括 Transfer Family 服务器 ID。这意味着您应在 AWS Secrets Manager中存储 PGP 秘钥信息之前识别或创建服务器。

如果您想为所有用户使用同一个密钥和密码，则可以将 PGP 密钥区块信息存储在机密名称`aws/transfer/`*`server-id`*`/@pgp-default`下，其中*`server-id`*是 Transfer Family 服务器的 ID。如果没有与执行工作流程用户*`user-name`* 匹配的密钥，则使用此默认密钥。

或者，您也可为特定用户创建密钥。在本例中，密钥名称的格式为 `aws/transfer/`*`server-id`*`/`*`user-name`*，其中 *`user-name`* 匹配正在为 Transfer Family 服务器运行工作流程的用户。

> **Note**
>
> 在每台 Transfer Family 服务器上，每位用户最多可存储 3 个 PGP 私钥。

配置用户解密的 PGP 密钥

1. 根据您使用的 GPG 版本，运行以下命令之一，生成不使用 Curve 25519 加密算法的 PGP 密钥对。

   - 如果您使用的是**GnuPG**版本为 2.3.0 或以上，请运行以下命令：

     ```
     gpg --full-gen-key
     ```

     您可选择**RSA**，或如果您选择**ECC**，您可为此椭圆曲线选择**NIST** 或 **BrainPool**。如果改为`gpg --gen-key`运行，则会创建一个使用 ECC Curve 25519 加密算法的密钥对，而我们目前不支持 PGP 密钥。

   - 对于 2.3.0 之前版本的**GnuPG**，您可以使用以下命令，原因是 RSA 是默认的加密类型。

     ```
     gpg --gen-key
     ```

> **Important**
>
> 密钥生成过程中，您必须提供密码和电子邮箱地址。请务必记下这些值。本过程后续在 AWS Secrets Manager 中输入密钥详细信息时，必须提供密码。您必须提供相同的电子邮件地址才能在下一步中导出私钥。

2. 运行以下命令以导出私钥。要使用此命令，请将*private.pgp*替换为用于保存私钥块的文件名，以及将*marymajor@example.com*替换为生成密钥对时使用的电子邮件地址。

```
gpg --output private.pgp --armor --export-secret-key marymajor@example.com
```

3. 用于存储 AWS Secrets Manager 您的 PGP 密钥。

   a. 登录 AWS Management Console 并打开 AWS Secrets Manager 控制台，网址为 https://console.aws.amazon.com/secretsmanager/。

   b. 在左侧导航窗格中，选择密钥。

   c. 在 Secrets（密钥）列表页上，选择副本密钥。

   d. 在 Store a new secret (存储新密钥) 页面上，对于 Select secret type (选择密钥类型)，选择 Other type of secrets (其他类型的密钥)。

   e. 在 秘钥/值对 部分，选择 秘钥/值选项卡。

   - 密钥 - 输入 **PGPPrivateKey**。

     > **Note**
     >
     > 必须准确输入 **PGPPrivateKey** 字符串：切勿在字符前面或字符之间添加任何空格。

   - value — 将您的私钥文本粘贴至值字段。您可以在文件中找到私钥文本（例如`private.pgp`），该文件是在您之前导出密钥时指定的文件。密钥开头为`-----BEGIN PGP PRIVATE KEY BLOCK-----`，结尾为`-----END PGP PRIVATE KEY BLOCK-----`。

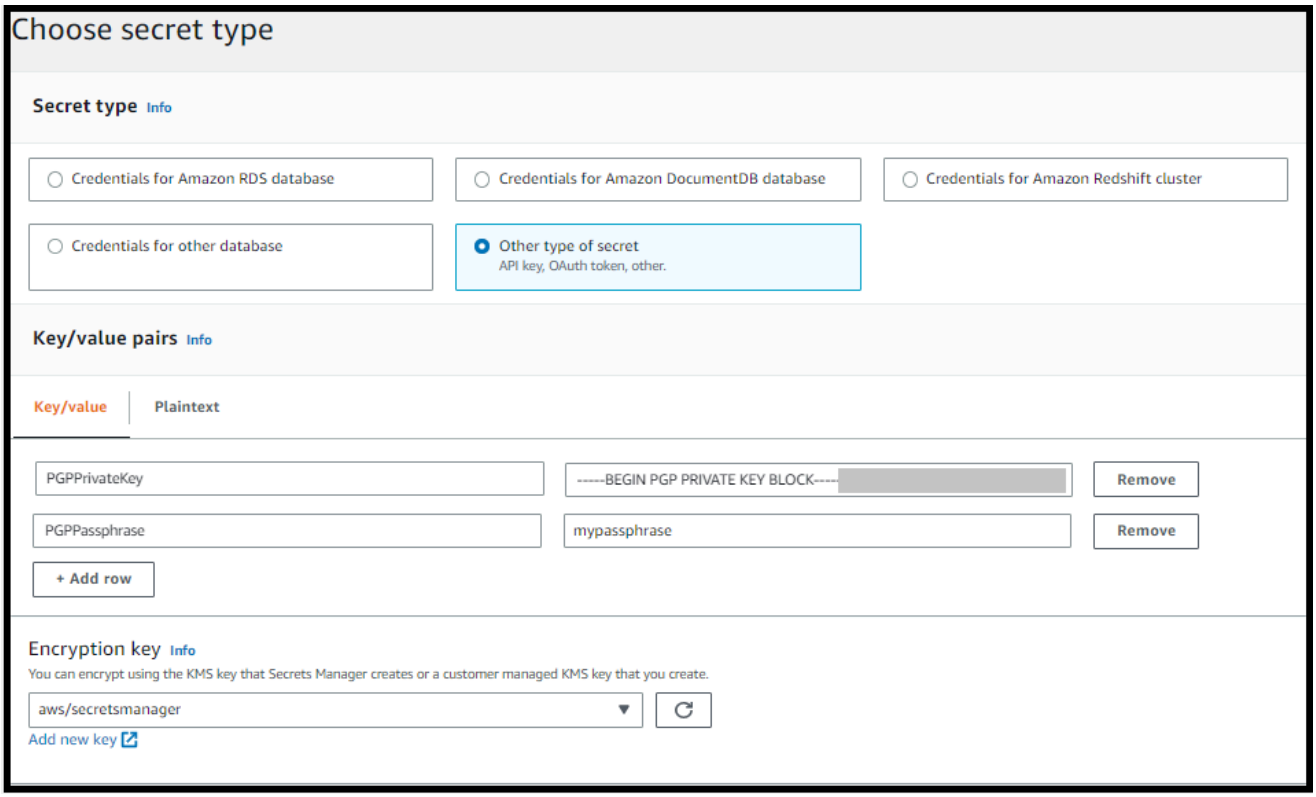
     > **Note**
     >
     > 确保文本块仅包含私钥，且不包含公钥。

f. 选择 添加行，然后在 秘钥/值对 部分选择秘钥/值选项卡。

- 密钥 - 输入 **PGPPassphrase**。

> **Note**
>
> 必须准确输入 **PGPPassphrase** 字符串：切勿在字符前面或字符之间添加任何空格。

- 值 – 输入您在生成 PGP 密钥对时使用的密码。



> **Note**
>
> 您最多可添加 3 组密钥和密码。若要添加第二组，请添加两行新行，为秘钥输入**PGPPrivateKey2** 和 **PGPPassphrase2**，并粘贴至其他私钥和密码。若要添加第三组，秘钥值必须为 **PGPPrivateKey3** 和 **PGPPassphrase3**。

g. 选择下一步。

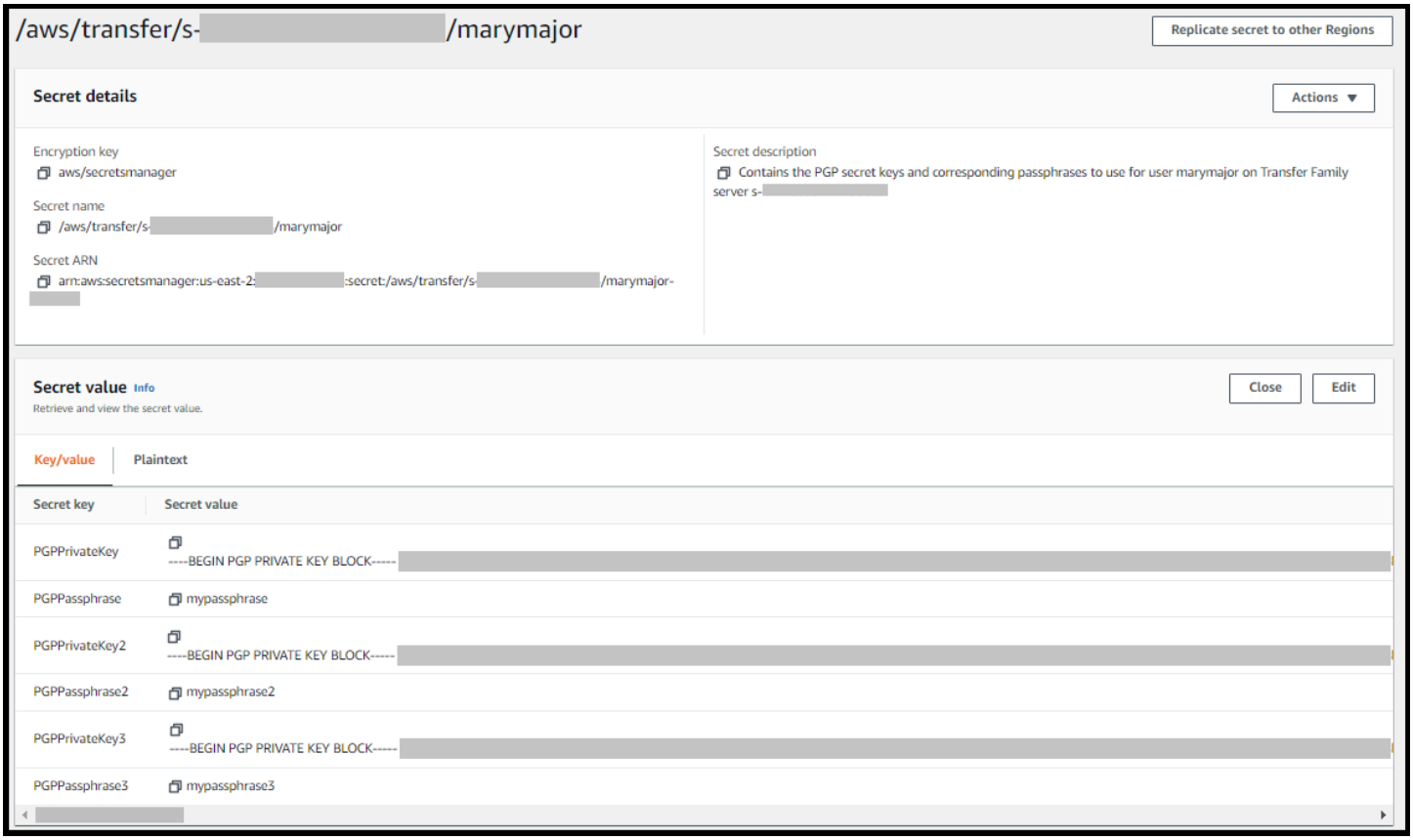
h. 在 配置密钥页面，输入密钥的名称和描述。

- 如果您要创建默认密钥，即可供任何 Transfer Family 用户使用的密钥，请输入 **aws/transfer/*server-id*/@pgp-default**。将*server-id*替换为包含解密工作流程服务器的 ID。
- 如果您正在创建供特定 Transfer Family 用户使用的密钥，请输入 **aws/transfer/*server-id*/*user-name***。将*server-id*替换为包含解密工作流程服务器的 ID，将*user-name*更换为运行工作流程的用户名称。*user-name*存储在 Transfer Family 服务器正在使用的身份提供程序。

i. 选择 下一步，接受 配置轮换 页面的默认设置。然后选择下一步。

j. 在 审核 页面，选择 存储 以创建和存储密钥。

以下屏幕截图显示了指定 Transfer Family 服务器用户 **marymajor**的详细信息。此示例显示三个密钥及其对应的密码。

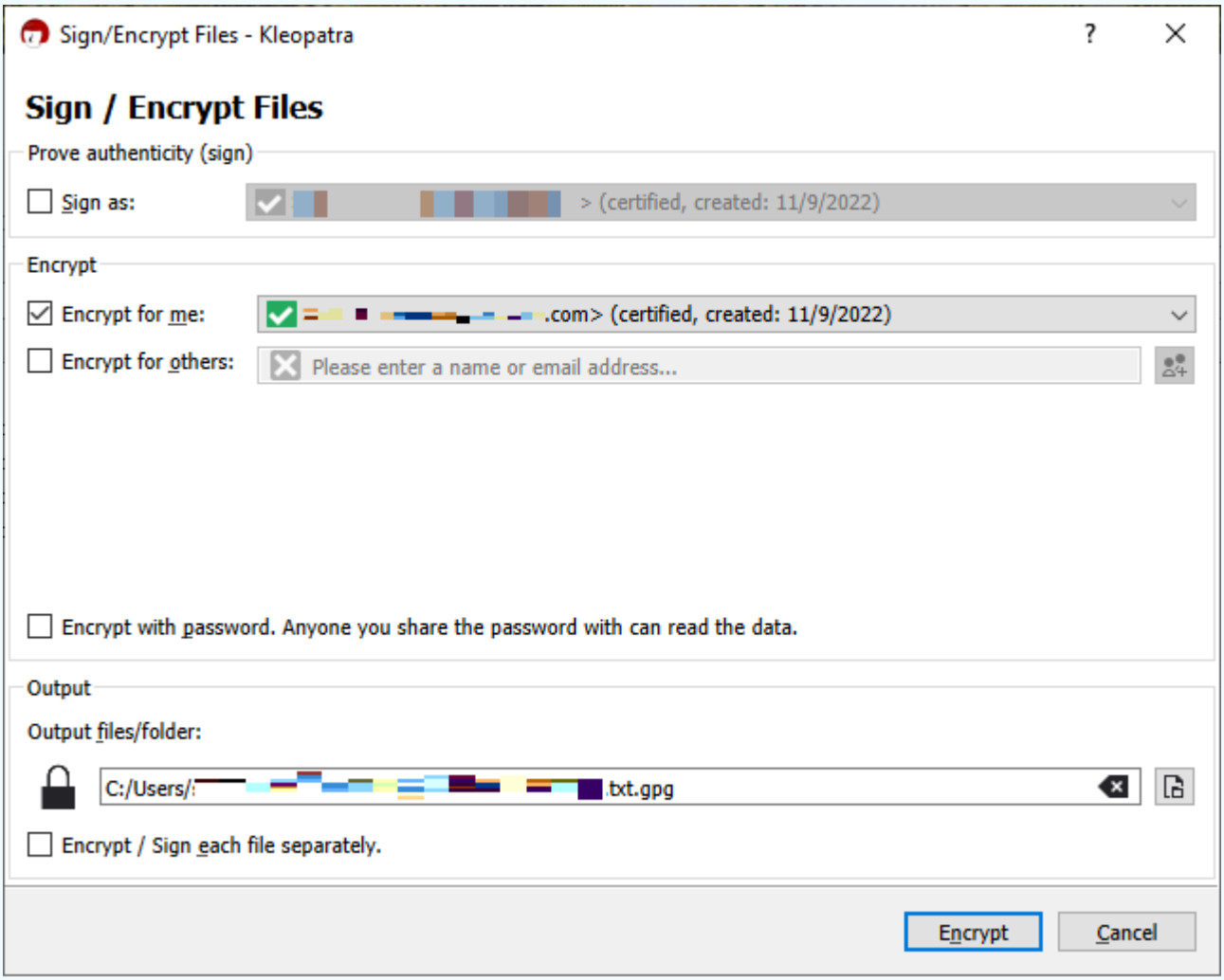


支持的 PGP 客户端

以下客户端已通过 Transfer Family 进行测试，可用于生成 PGP 密钥，以及加密您打算通过工作流程解密的文件。

- Gpg4win + Kleopatra.

> **Note**
>
> 当您选择 签名/加密文件 时，请务必取消选择 签名身份：我们目前不支持对加密文件进行签名。



如果您对加密文件进行签名并尝试使用解密工作流程将其上传到 Transfer Family 服务器，则会收到以下错误：

```
Encrypted file with signed message unsupported
```

- GnuPG 主要版本：2.4、2.3、2.2、2.0 和 1.4。

请注意，其他 PGP 客户端也可运行，但只有此处提到的客户端通过 Transfer Family 进行了测试。

的身份和访问管理 AWS Transfer Family

AWS Identity and Access Management (IAM) AWS 服务 可帮助管理员安全地控制对 AWS 资源的访问权限。IAM 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 AWS Transfer Family 资源。您可以使用 IAM AWS 服务 ，无需支付额外费用。

主题

受众

您的使用方式 AWS Identity and Access Management (IAM) 会有所不同，具体取决于您所做的工作 AWS Transfer Family。

服务用户-如果您使用该 AWS Transfer Family 服务完成工作，则您的管理员会为您提供所需的凭证和权限。当你使用更多 AWS Transfer Family 功能来完成工作时，你可能需要额外的权限。了解如何管理访问权限有助于您向管理员请求适合的权限。如果您无法访问 AWS Transfer Family中的特征，请参阅 对 AWS Transfer Family 身份和访问进行故障排除。

服务管理员-如果您负责公司的 AWS Transfer Family 资源，则可能拥有完全访问权限 AWS Transfer Family。您的工作是确定您的服务用户应访问哪些 AWS Transfer Family 功能和资源。然后，您必须向 IAM 管理员提交请求以更改服务用户的权限。请查看该页面上的信息以了解 IAM 的基本概念。要详细了解您的公司如何将 IAM 与配合使用 AWS Transfer Family，请参阅如何 AWS Transfer Family 与 IAM 配合使用。

IAM 管理员：如果您是 IAM 管理员，您可能希望了解如何编写策略以管理对 AWS Transfer Family的访问权限的详细信息。要查看您可以在 IAM 中使用的 AWS Transfer Family 基于身份的策略示例，请参阅。AWS Transfer Family 基于身份的策略示例

使用身份进行身份验证

身份验证是您 AWS 使用身份凭证登录的方式。您必须以 IAM 用户身份或通过担 AWS 账户根用户任 IAM 角色进行身份验证（登录 AWS）。

您可以使用通过身份源提供的凭据以 AWS 联合身份登录。 AWS IAM Identity Center （IAM Identity Center）用户、贵公司的单点登录身份验证以及您的 Google 或 Facebook 凭据就是联合身份的示例。当您以联合身份登录时，您的管理员以前使用 IAM 角色设置了身份联合验证。当你使用联合访问 AWS 时，你就是在间接扮演一个角色。

根据您的用户类型，您可以登录 AWS Management Console 或 AWS 访问门户。有关登录的更多信息 AWS，请参阅《AWS 登录 用户指南》中的如何登录到您 AWS 账户的。

如果您 AWS 以编程方式访问，则会 AWS 提供软件开发套件 (SDK) 和命令行接口 (CLI)，以便使用您的凭据对请求进行加密签名。如果您不使用 AWS 工具，则必须自己签署请求。有关使用推荐的方法自行签署请求的更多信息，请参阅 IAM 用户指南中的签署 AWS API 请求。

无论使用何种身份验证方法，您可能需要提供其他安全信息。例如， AWS 建议您使用多重身份验证 (MFA) 来提高账户的安全性。要了解更多信息，请参阅《AWS IAM Identity Center 用户指南》中的 多重身份验证 和《IAM 用户指南》中的 在 AWS中使用多重身份验证（MFA）。

AWS 账户根用户

创建时 AWS 账户，首先要有一个登录身份，该身份可以完全访问账户中的所有资源 AWS 服务 和资源。此身份被称为 AWS 账户 root 用户，使用您创建账户时使用的电子邮件地址和密码登录即可访问该身份。强烈建议您不要使用根用户执行日常任务。保护好根用户凭证，并使用这些凭证来执行仅根用户可以执行的任务。有关要求您以根用户身份登录的任务的完整列表，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（包括需要管理员访问权限的用户）使用与身份提供商的联合身份验证 AWS 服务 通过临时证书进行访问。

联合身份是指您的企业用户目录、Web 身份提供商、Identity C enter 目录中的用户，或者任何使用 AWS 服务 通过身份源提供的凭据进行访问的用户。 AWS Directory Service当联合身份访问时 AWS 账户，他们将扮演角色，角色提供临时证书。

要集中管理访问权限，建议您使用 AWS IAM Identity Center。您可以在 IAM Identity Center 中创建用户和群组，也可以连接并同步到您自己的身份源中的一组用户和群组，以便在您的所有 AWS 账户 和

应用程序中使用。有关 IAM Identity Center 的信息，请参阅《AWS IAM Identity Center 用户指南》中的什么是 IAM Identity Center？。

IAM 用户和群组

I AM 用户是您 AWS 账户 内部对个人或应用程序具有特定权限的身份。在可能的情况下，我们建议使用临时凭证，而不是创建具有长期凭证（如密码和访问密钥）的 IAM 用户。但是，如果您有一些特定的使用场景需要长期凭证以及 IAM 用户，我们建议您轮换访问密钥。有关更多信息，请参阅《IAM 用户指南》中的对于需要长期凭证的使用场景定期轮换访问密钥。

IAM 组是一个指定一组 IAM 用户的身份。您不能使用组的身份登录。您可以使用组来一次性为多个用户指定权限。如果有大量用户，使用组可以更轻松地管理用户权限。例如，您可能具有一个名为 IAMAdmins 的组，并为该组授予权限以管理 IAM 资源。

用户与角色不同。用户唯一地与某个人员或应用程序关联，而角色旨在让需要它的任何人代入。用户具有永久的长期凭证，而角色提供临时凭证。要了解更多信息，请参阅 IAM 用户指南中的何时创建 IAM 用户（而不是角色）。

IAM 角色

I AM 角色是您内部具有特定权限 AWS 账户 的身份。它类似于 IAM 用户，但与特定人员不关联。您可以 AWS Management Console 通过切换角色在中临时担任 IAM 角色。您可以通过调用 AWS CLI 或 AWS API 操作或使用自定义 URL 来代入角色。有关使用角色的方法的更多信息，请参阅《IAM 用户指南》中的 使用 IAM 角色。

具有临时凭证的 IAM 角色在以下情况下很有用：

- 联合用户访问 – 要向联合身份分配权限，请创建角色并为角色定义权限。当联合身份进行身份验证时，该身份将与角色相关联并被授予由此角色定义的权限。有关联合身份验证的角色的信息，请参阅《IAM 用户指南》中的 为第三方身份提供商创建角色。如果您使用 IAM Identity Center，则需要配置权限集。为控制您的身份在进行身份验证后可以访问的内容，IAM Identity Center 将权限集与 IAM 中的角色相关联。有关权限集的信息，请参阅《AWS IAM Identity Center 用户指南》中的权限集。
- 临时 IAM 用户权限 – IAM 用户可代入 IAM 用户或角色，以暂时获得针对特定任务的不同权限。
- 跨账户存取 – 您可以使用 IAM 角色以允许不同账户中的某个人（可信主体）访问您的账户中的资源。角色是授予跨账户访问权限的主要方式。但是，对于某些资源 AWS 服务，您可以将策略直接附加到资源（而不是使用角色作为代理）。要了解用于跨账户访问的角色和基于资源的策略之间的差别，请参阅《IAM 用户指南》中的 IAM 角色与基于资源的策略有何不同。

- 跨服务访问 — 有些 AWS 服务 使用其他 AWS 服务服务中的功能。例如，当您在某个服务中进行调用时，该服务通常会在 Amazon EC2 中运行应用程序或在 Amazon S3 中存储对象。服务可能会使用发出调用的主体的权限、使用服务角色或使用服务相关角色来执行此操作。
  - 转发访问会话 (FAS) — 当您使用 IAM 用户或角色在中执行操作时 AWS，您被视为委托人。使用某些服务时，您可能会执行一个操作，此操作然后在不同服务中启动另一个操作。FAS 使用调用委托人的权限以及 AWS 服务 向下游服务发出请求的请求。 AWS 服务只有当服务收到需要与其他 AWS 服务 或资源交互才能完成的请求时，才会发出 FAS 请求。在这种情况下，您必须具有执行这两个操作的权限。有关发出 FAS 请求时的策略详情，请参阅转发访问会话。
  - 服务角色 - 服务角色是服务代表您在您的账户中执行操作而分派的 IAM 角色。IAM 管理员可以在 IAM 中创建、修改和删除服务角色。有关更多信息，请参阅《IAM 用户指南》中的创建向 AWS 服务委派权限的角色。
  - 服务相关角色-服务相关角色是一种与服务相关联的服务角色。 AWS 服务服务可以代入代表您执行操作的角色。服务相关角色出现在您的中 AWS 账户 ，并且归服务所有。IAM 管理员可以查看但不能编辑服务相关角色的权限。
- 在 A@@ mazon EC2 上运行的应用程序 — 您可以使用 IAM 角色管理在 EC2 实例上运行并发出 AWS CLI 或 AWS API 请求的应用程序的临时证书。这优先于在 EC2 实例中存储访问密钥。要向 EC2 实例分配 AWS 角色并使其可供其所有应用程序使用，您需要创建附加到该实例的实例配置文件。实例配置文件包含角色，并使 EC2 实例上运行的程序能够获得临时凭证。有关更多信息，请参阅《IAM 用户指南》中的 使用 IAM 角色为 Amazon EC2 实例上运行的应用程序授予权限。

要了解是使用 IAM 角色还是 IAM 用户，请参阅 IAM 用户指南中的何时创建 IAM 角色（而不是用户）。

使用策略管理访问

您可以 AWS 通过创建策略并将其附加到 AWS 身份或资源来控制中的访问权限。策略是其中的一个对象 AWS ，当与身份或资源关联时，它会定义其权限。 AWS 在委托人（用户、root 用户或角色会话）发出请求时评估这些策略。策略中的权限确定是允许还是拒绝请求。大多数策略都以 JSON 文档的 AWS 形式存储在中。有关 JSON 策略文档的结构和内容的更多信息，请参阅《IAM 用户指南》中的 JSON 策略概览。

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体 可以对什么资源执行操作，以及在什么条件下执行。

默认情况下，用户和角色没有权限。要授予用户对所需资源执行操作的权限，IAM 管理员可以创建 IAM 策略。然后，管理员可以向角色添加 IAM 策略，并且用户可以代入角色。

IAM 策略定义操作的权限，无关乎您使用哪种方法执行操作。例如，假设您有一个允许 `iam:GetRole` 操作的策略。拥有该策略的用户可以从 AWS Management Console AWS CLI、或 AWS API 获取角色信息。

基于身份的策略

基于身份的策略是可附加到身份（如 IAM 用户、用户组或角色）的 JSON 权限策略文档。这些策略控制用户和角色可在何种条件下对哪些资源执行哪些操作。要了解如何创建基于身份的策略，请参阅《IAM 用户指南》中的创建 IAM 策略。

基于身份的策略可以进一步归类为内联策略或托管式策略。内联策略直接嵌入单个用户、组或角色中。托管策略是独立的策略，您可以将其附加到中的多个用户、群组和角色 AWS 账户。托管策略包括 AWS 托管策略和客户托管策略。要了解如何在托管式策略和内联策略之间进行选择，请参阅《IAM 用户指南》中的在托管式策略与内联策略之间进行选择。

基于资源的策略

基于资源的策略是附加到资源的 JSON 策略文档。基于资源的策略的示例包括 IAM 角色信任策略和 Simple Storage Service（Amazon S3）存储桶策略。在支持基于资源的策略的服务中，服务管理员可以使用它们来控制对特定资源的访问。对于在其中附加策略的资源，策略定义指定主体可以对该资源执行哪些操作以及在什么条件下执行。您必须在基于资源的策略中指定主体。委托人可以包括账户、用户、角色、联合用户或 AWS 服务。

基于资源的策略是位于该服务中的内联策略。您不能在基于资源的策略中使用 IAM 中的 AWS 托管策略。

访问控制列表 (ACL)

访问控制列表（ACL）控制哪些主体（账户成员、用户或角色）有权访问资源。ACL 与基于资源的策略类似，尽管它们不使用 JSON 策略文档格式。

Amazon S3 和 Amazon VPC 就是支持 ACL 的服务示例。 AWS WAF要了解有关 ACL 的更多信息，请参阅 Amazon Simple Storage Service 开发人员指南 中的访问控制列表（ACL）概览。

其它策略类型

AWS 支持其他不太常见的策略类型。这些策略类型可以设置更常用的策略类型向您授予的最大权限。

- 权限边界 - 权限边界是一个高级特征，用于设置基于身份的策略可以为 IAM 实体（IAM 用户或角色）授予的最大权限。您可为实体设置权限边界。这些结果权限是实体基于身份的策略及其权限边界

的交集。在 `Principal` 中指定用户或角色的基于资源的策略不受权限边界限制。任一项策略中的显式拒绝将覆盖允许。有关权限边界的更多信息，请参阅 IAM 用户指南中的 IAM 实体的权限边界。

- 服务控制策略 (SCP)-SCP 是 JSON 策略，用于指定组织或组织单位 (OU) 的最大权限。 AWS Organizations AWS Organizations 是一项用于对您的企业拥有的多 AWS 账户 项进行分组和集中管理的服务。如果在组织内启用了所有特征，则可对任意或全部账户应用服务控制策略 (SCP)。SCP 限制成员账户中的实体（包括每个 AWS 账户根用户实体）的权限。有关 Organizations 和 SCP 的更多信息，请参阅 AWS Organizations 用户指南中的 SCP 的工作原理。
- 会话策略 – 会话策略是当您以编程方式为角色或联合身份用户创建临时会话时作为参数传递的高级策略。结果会话的权限是用户或角色的基于身份的策略和会话策略的交集。权限也可以来自基于资源的策略。任一项策略中的显式拒绝将覆盖允许。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 AWS 确定是否允许请求，请参阅 IAM 用户指南中的策略评估逻辑。

如何 AWS Transfer Family 与 IAM 配合使用

在使用 AWS Identity and Access Management (IAM) 管理访问权限之前 AWS Transfer Family，您应该了解哪些可用的 IAM 功能 AWS Transfer Family。要全面了解如何 AWS Transfer Family 和其他 AWS 服务与 IAM 配合使用，请参阅 IAM 用户指南中的与 IAM 配合使用的AWS 服务。

主题

- AWS Transfer Family 基于身份的策略
- AWS Transfer Family 基于资源的策略
- 基于 AWS Transfer Family 标签的授权
- AWS Transfer Family IAM 角色

AWS Transfer Family 基于身份的策略

使用 IAM 基于身份的策略，您可以指定允许或拒绝的操作和资源，以及指定在什么条件下允许或拒绝操作。 AWS Transfer Family 支持特定操作、资源和条件键。要了解您在 JSON 策略中使用的所有元素，请参阅 AWS Identity and Access Management IAM 用户指南中的 IAM JSON 策略元素参考。

操作

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体 可以对什么资源 执行操作，以及在什么 条件 下执行。

JSON 策略的 `Action` 元素描述可用于在策略中允许或拒绝访问的操作。策略操作通常与关联的 AWS API 操作同名。有一些例外情况，例如没有匹配 API 操作的仅限权限 操作。还有一些操作需要在策略中执行多个操作。这些附加操作称为相关操作。

在策略中包含操作以授予执行关联操作的权限。

正在执行的策略操作在操作前 AWS Transfer Family 使用以下前缀:`transfer:`. 例如，要授予某人使用 Amazon EC2 `CreateServer` API 操作创建 VPC 的权限，您应将 `transfer:CreateServer` 操作纳入其策略中。策略语句必须包括 `Action` 或 `NotAction` 元素。 AWS Transfer Family 定义了自己的一组操作，这些操作描述了可使用该服务执行的任务。

要在单个语句中指定多项 操作，请使用逗号将它们隔开，如下所示。

```
"Action": [
      "transfer:action1",
      "transfer:action2"
```

您也可以使用通配符 (*) 指定多个操作。例如，要指定以单词 `Describe` 开头的所有操作，请包括以下操作。

```
"Action": "transfer:Describe*"
```

要查看 AWS Transfer Family 操作列表，请参阅《服务授权参考》 AWS Transfer Family中定义的操作。

资源

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体 可以对什么资源 执行操作，以及在什么条件 下执行。

`Resource` JSON 策略元素指定要向其应用操作的一个或多个对象。语句必须包含 `Resource` 或 `NotResource` 元素。作为最佳实践，请使用其 Amazon 资源名称（ARN）指定资源。对于支持特定资源类型（称为资源级权限）的操作，您可以执行此操作。

对于不支持资源级权限的操作（如列出操作），请使用通配符 (*) 指示语句应用于所有资源。

```
"Resource": "*"
```

跟踪资源拥有以下 ARN：

```
arn:aws:transfer:${Region}:${Account}:server/${ServerId}
```

例如，要在语句中指定 s-01234567890abcdef 事件，请使用以下 ARN。

```
"Resource": "arn:aws:transfer:us-east-1:123456789012:server/s-01234567890abcdef"
```

有关 ARN 格式的更多信息，请参见服务授权引用中的Amazon Resource Names (ARNs)，或IAM 用户指南中的IAM ARN。

要指定属于特定账户的所有实例，请使用通配符 (*)。

```
"Resource": "arn:aws:transfer:us-east-1:123456789012:server/*"
```

有些 AWS Transfer Family 操作是在多个资源上执行的，例如 IAM 策略中使用的资源。在这些情况下，您必须使用通配符 （ *)。

```
"Resource": "arn:aws:transfer:*:123456789012:server/*"
```

在某些情况下，您需要指定多种类型的资源，例如，如果您创建了允许访问 Transfer Family 服务器与用户的策略。要在单个语句中指定多个资源，请使用逗号分隔 ARN。

```
"Resource": [
      "resource1",
      "resource2"
            ]
```

要查看 AWS Transfer Family 资源列表，请参阅《服务授权参考》 AWS Transfer Family中定义的资源类型。

条件键

管理员可以使用 AWS JSON 策略来指定谁有权访问什么。也就是说，哪个主体可以对什么资源执行操作，以及在什么条件下执行。

在 Condition 元素（或 Condition 块）中，可以指定语句生效的条件。Condition 元素是可选的。您可以创建使用条件运算符（例如，等于或小于）的条件表达式，以使策略中的条件与请求中的值相匹配。

如果您在一个语句中指定多个 Condition 元素，或在单个 Condition 元素中指定多个键，则 AWS 使用逻辑 AND 运算评估它们。如果您为单个条件键指定多个值，则使用逻辑OR运算来 AWS 评估条件。在授予语句的权限之前必须满足所有的条件。

在指定条件时，您也可以使用占位符变量。例如，只有在使用 IAM 用户名标记 IAM 用户时，您才能为其授予访问资源的权限。有关更多信息，请参阅《IAM 用户指南》中的 IAM 策略元素：变量和标签。

AWS 支持全局条件密钥和特定于服务的条件密钥。要查看所有 AWS 全局条件键，请参阅 IAM 用户指南中的AWS 全局条件上下文密钥。

AWS Transfer Family 定义自己的条件键集，还支持使用一些全局条件键。要查看 AWS Transfer Family 条件密钥列表，请参阅《服务授权参考》 AWS Transfer Family中的条件密钥。

示例

要查看 AWS Transfer Family 基于身份的策略的示例，请参阅。AWS Transfer Family 基于身份的策略示例

AWS Transfer Family 基于资源的策略

基于资源的策略是 JSON 策略文档，用于指定委托人可以在哪些条件下对 AWS Transfer Family 资源执行哪些操作。Amazon ECR 支持针对 Amazon ECR 存储库的基于资源的权限策略。基于资源的策略允许您基于资源向其他账户授予使用权限。您还可以使用基于资源的策略来允许 AWS 服务访问您的 Amazon S3 存储#。

要启用跨账户访问，您可以将整个账户或其他账户中的 IAM 实体指定为基于资源的策略中的委托人。将跨账户主体添加到基于资源的策略只是建立信任关系工作的一半而已。当委托人和资源位于不同的 AWS 账户中时，您还必须向委托人实体授予访问资源的权限。通过将基于身份的策略附加到实体以授予权限。但是，如果基于资源的策略向同一个账户中的主体授予访问权限，则不需要额外的基于身份的策略。有关更多信息，请参阅《IAM 用户指南》中的 IAM 角色与基于资源的策略有何不同AWS Identity and Access Management 。

Amazon ECR 服务仅支持一种类型的基于资源的策略（称为####），这种策略附加到存储库。这个策略定义哪些委托人实体（账户、用户、角色和联合身份用户）可以在容器上执行操作。

示例

要查看 AWS Transfer Family 基于资源的策略的示例，请参阅AWS Transfer Family 基于标签的策略示例。

基于 AWS Transfer Family 标签的授权

您可以为 AWS Transfer Family 资源附加标签或在请求中传递标签 AWS Transfer Family。要基于标签控制访问，您需要使用 `transfer:ResourceTag/key-name`、`aws:RequestTag/key-name` 或 `aws:TagKeys` 条件键在策略的条件元素中提供标签信息。有关如何使用标签控制对 AWS Transfer Family 资源的访问的信息，请参阅AWS Transfer Family 基于标签的策略示例。

AWS Transfer Family IAM 角色

I AM 角色是您的 AWS 账户中具有特定权限的实体。

将临时证书与 AWS Transfer Family

可以使用临时凭证进行联合身份验证登录，分派 IAM 角色或分派跨账户角色。您可以通过调用AssumeRole或之类的 AWS STS API 操作来获取临时安全证书GetFederationToken。

AWS Transfer Family 支持使用临时证书。

AWS Transfer Family 基于身份的策略示例

默认情况下，IAM 用户和角色没有创建或修改 AWS Transfer Family 资源的权限。他们也无法使用 AWS Management Console AWS CLI、或 AWS API 执行任务。IAM 管理员必须创建 IAM 策略，以便为用户和角色授予权限以对所需的指定资源执行特定的 API 操作。然后，管理员必须将这些策略附加到需要这些权限的 IAM 用户或组。

要了解如何使用这些示例 JSON 策略文档创建 IAM 基于身份的策略，请参阅 AWS Identity and Access Management IAM 用户指南中的在 JSON 选项卡上创建策略。

主题

- 策略最佳实践
- 使用 AWS Transfer Family 控制台
- 允许用户查看他们自己的权限

策略最佳实践

基于身份的策略决定了某人是否可以在您的账户中创建、访问或删除 AWS Transfer Family 资源。这些操作可能会使 AWS 账户产生成本。创建或编辑基于身份的策略时，请遵循以下准则和建议：

- 开始使用 AWS 托管策略并转向最低权限权限 — 要开始向用户和工作负载授予权限，请使用为许多常见用例授予权限的AWS 托管策略。它们在你的版本中可用 AWS 账户。我们建议您通过定义针对您的用例的 AWS 客户托管策略来进一步减少权限。有关更多信息，请参阅《IAM 用户指南》中的AWS 托管策略或工作职能的AWS 托管策略。
- 应用最低权限 – 在使用 IAM 策略设置权限时，请仅授予执行任务所需的权限。为此，您可以定义在特定条件下可以对特定资源执行的操作，也称为最低权限许可。有关使用 IAM 应用权限的更多信息，请参阅《IAM 用户指南》中的 IAM 中的策略和权限。
- 使用 IAM 策略中的条件进一步限制访问权限 – 您可以向策略添加条件来限制对操作和资源的访问。例如，您可以编写策略条件来指定必须使用 SSL 发送所有请求。如果服务操作是通过特定的方式使用的，则也可以使用条件来授予对服务操作的访问权限 AWS 服务，例如 AWS CloudFormation。有关更多信息，请参阅《IAM 用户指南》中的 IAM JSON 策略元素：条件。
- 使用 IAM Access Analyzer 验证您的 IAM 策略，以确保权限的安全性和功能性 – IAM Access Analyzer 会验证新策略和现有策略，以确保策略符合 IAM 策略语言 (JSON) 和 IAM 最佳实践。IAM Access Analyzer 提供 100 多项策略检查和可操作的建议，以帮助您制定安全且功能性强的策略。有关更多信息，请参阅《IAM 用户指南》中的 IAM Acess Analyzer 策略验证。
- 需要多重身份验证 (MFA)-如果 AWS 账户您的场景需要 IAM 用户或根用户，请启用 MFA 以提高安全性。若要在调用 API 操作时需要 MFA，请将 MFA 条件添加到您的策略中。有关更多信息，请参阅《IAM 用户指南》中的 配置受 MFA 保护的 API 访问。

有关 IAM 中的最佳实操的更多信息，请参阅《IAM 用户指南》中的 IAM 中的安全最佳实操。

使用 AWS Transfer Family 控制台

要访问 AWS Transfer Family 控制台，您必须拥有一组最低权限。这些权限必须允许您列出和查看有关您 AWS 账户中 AWS Transfer Family 资源的详细信息。如果您创建的基于身份的策略比所需的最低权限更严格，则无法为具有该策略的实体（IAM 用户或角色）正常运行控制台。有关更多信息，请参阅 AWS Identity and Access Management IAM 用户指南中的为用户添加权限。

对于仅调用 AWS CLI 或 AWS API 的用户，您无需为其设置最低控制台权限。相反，只允许访问与您尝试执行的 API 操作相匹配的操作。

允许用户查看他们自己的权限

该示例说明了您如何创建策略，以允许 IAM 用户查看附加到其用户身份的内联和托管式策略。此策略包括在控制台上或使用 AWS CLI 或 AWS API 以编程方式完成此操作的权限。

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

```
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS Transfer Family 基于标签的策略示例

以下是如何根据标签控制 AWS Transfer Family 资源访问权限的示例。

使用标签控制对 AWS Transfer Family 资源的访问

策略中的条件是所需语法的一部分，您可以使用它们指定对 资源的权限。您可以根据这些 AWS Transfer Family 资源的标签来控制对这些资源（例如用户、服务器、角色和其他实体）的访问权限。标签是键值对。有关为资源添加标签的更多信息，请参阅中的为AWS 资源添加标签。AWS 一般参考

在中 AWS Transfer Family，资源可以有标签，有些操作可以包含标签。在创建 IAM 策略时，您可以使用标签条件键来控制以下：

- 根据 AWS Transfer Family 资源所具有的标签，哪些用户可以对资源执行操作。
- 哪些标签可以在操作的请求中传递。
- 是否特定标签键可在请求中使用。

通过使用基于标签的访问控制，您可以应用比 API 级别更精细的控制。与使用基于资源的访问控制相比，您还可以应用更多动态控制。您可以创建 IAM 策略，以允许或拒绝按请求中提供的标签（请求标签）执行操作。您还可以根据正在操作资源的标签（资源标签）创建 IAM 策略。通常，资源标签用于资源上已有的标签，请求标签用于向资源添加标签或从资源中删除标签。

有关标签条件键的完整请求和语义，请参阅《IAM 用户指南》中的使用标签控制访问。有关使用 API Gateway 指定 IAM 策略的详细信息，请参阅API Gateway 开发人员指南中的控制访问具有 IAM 权限的 API。

示例 3：基于资源标签拒绝操作

您可根据标签拒绝资源上执行的操作。如果用户或服务器资源通过秘钥 `stage` 和值 `prod`标记，则以下示例策略拒绝`TagResource`、`UntagResource`、`StartServer`、`StopServer`、`DescribeServer` 以及 `DescribeUser`操作。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "transfer:TagResource",
                "transfer:UntagResource",
                "transfer:StartServer",
                "transfer:StopServer",
                "transfer:DescribeServer",
                "transfer:DescribeUser
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/stage": "prod"
                }
            }
        }
```

```
    ]
}
```

示例 4：基于资源标签允许操作

您可根据标签拒绝资源上执行的操作。如果用户或服务器资源通过秘钥 stage 和值 prod标记，则以下示例策略拒
绝TagResource、UntagResource、StartServer、StopServer、DescribeServer 以及 DescribeUser操作。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "transfer:TagResource",
                "transfer:UntagResource",
                "transfer:StartServer",
                "transfer:StopServer",
                "transfer:DescribeServer",
                "transfer:DescribeUser
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:ResourceTag/stage": "prod"
                }
            }
        }
    ]
}
```

示例 3：拒绝根据请求标签创建用户或服务器

以下示例策略包含两个语句。如果标签的成本中心密钥无值，则第一条语句拒绝对所有资源执行 CreateServer 操作。

如果标签的成本中心密钥包含除 1、2 或 3 之外的任何其他值，则第二条语句将拒绝 CreateServer 操作。

> **Note**
>
> 此策略确实允许创建或删除包含costcenter秘钥和1、2 或 3 值的资源。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "transfer:CreateServer"
            ],
            "Resource": [
                "*"
            ],
            "Condition": {
                "Null":  {
                    "aws:RequestTag/costcenter": "true"
                }
            }
        },
        {
            "Effect": "Deny",
            "Action": "transfer:CreateServer",
            "Resource": [
                "*"
            ],
            "Condition": {
                "ForAnyValue:StringNotEquals": {
                    "aws:RequestTag/costcenter": [
                        "1",
                        "2",
                        "3"
                    ]
                }
            }
        }
    ]
}
```

对 AWS Transfer Family 身份和访问进行故障排除

使用以下信息来帮助您诊断和修复在使用 AWS Transfer Family 和 IAM 时可能遇到的常见问题。

主题

我无权在以下位置执行操作 AWS Transfer Family

如果 AWS Management Console 告诉您您无权执行某项操作，则必须联系管理员寻求帮助。管理员是向您提供登录凭证的人。

当 `mateojackson` IAM 用户尝试使用控制台查看有关 *`widget`* 的详细信息，但不具有 `transfer:`*`GetWidget`* 权限时，会发生以下示例错误。

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 transfer:GetWidget on resource: my-example-widget
```

在这种情况下，Mateo 请求他的管理员更新其策略，以允许他使用 `transfer;:`*`GetWidget`* 操作访问 *`my-example-widget`* 资源。

我无权执行 iam：PassRole

如果您收到一个错误，表明您无权执行 `iam:PassRole` 操作，则必须更新策略以允许您将角色传递给 AWS Transfer Family。

有些 AWS 服务 允许您将现有角色传递给该服务，而不是创建新的服务角色或服务相关角色。为此，您必须具有将角色传递到服务的权限。

当名为 `marymajor` 的 IAM 用户尝试使用控制台在 AWS Transfer Family中执行操作时，会发生以下示例错误。但是，服务必须具有服务角色所授予的权限才可执行此操作。Mary 不具有将角色传递到服务的权限。

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

在这种情况下，必须更新 Mary 的策略以允许她执行 `iam:PassRole` 操作。

如果您需要帮助，请联系您的 AWS 管理员。您的管理员是提供登录凭证的人。

以下示例策略为 中的 角色授予调用 的权限。

```
{
   "Version": "2012-10-17",
   "Statement": [
      { "Action": "iam:PassRole",
        "Resource": "arn:aws::iam::123456789012:role/*",
        "Effect": "Allow"
      }
   ]
}
```

我想允许 AWS 账户之外的人访问我的 AWS Transfer Family 资源

您可以创建一个角色，以便其他账户中的用户或您组织外的人员可以使用该角色来访问您的资源。您可以指定谁值得信赖，可以担任角色。对于支持基于资源的策略或访问控制列表（ACL）的服务，您可以使用这些策略向人员授予对您的资源的访问权。

要了解更多信息，请参阅以下内容：

- 要了解是否 AWS Transfer Family 支持这些功能，请参阅如何 AWS Transfer Family 与 IAM 配合使用。
- 要了解如何提供对您拥有的资源的访问权限 AWS 账户 ，请参阅 IAM 用户指南中的向您拥有 AWS 账户 的另一个 IAM 用户提供访问权限。
- 要了解如何向第三方提供对您的资源的访问权限 AWS 账户，请参阅 IAM 用户指南中的向第三方提供访问权限。 AWS 账户
- 要了解如何通过联合身份验证提供访问权限，请参阅 IAM 用户指南中的为经过外部身份验证的用户（联合身份验证）提供访问权限。
- 要了解使用角色和基于资源的策略进行跨账户存取之间的差别，请参阅《IAM 用户指南》中的 IAM 角色与基于资源的策略有何不同。

合规性验证 AWS Transfer Family

AWS Transfer Family 作为多个合规计划的一部分，第三方审计师对安全性和 AWS 合规性进行评估。其中包括 SOC、PCI、HIPAA 等。有关完整列表，请参阅合规性计划范围内AWS 服务。

有关特定合规计划范围内的 AWS 服务列表，请参阅按合规计划划分的范围内的AWS 服务。有关一般信息，请参阅 AWS 合规性计划。

您可以使用下载第三方审计报告 AWS Artifact。有关更多信息，请参阅中的下载报告 AWS Artifact。

您在使用 AWS Transfer Family 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。 AWS 提供了以下资源来帮助实现合规性：

- 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。 AWS
- HIPAA 安全与合规架构白皮书 — 本白皮书描述了各公司如何使用它来 AWS 创建符合 HIPAA 标准的应用程序。
- AWS 合规性资源 – 此业务手册和指南集合可能适用于您的行业和位置。
- AWS Config— 该 AWS 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。
- AWS Security Hub— 此 AWS 服务可全面了解您的安全状态 AWS ，帮助您检查是否符合安全行业标准和最佳实践。

韧性在 AWS Transfer Family

AWS 全球基础设施是围绕 AWS 区域和可用区构建的。 AWS 区域提供多个物理隔离和隔离的可用区，这些可用区通过低延迟、高吞吐量和高度冗余的网络相连。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故障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设施相比，可用区具有更高的可用性、容错性和可扩展性。

AWS Transfer Family 最多支持 3 个可用区，并由 auto Scaling 的冗余队列提供支持，用于处理您的连接和传输请求。

请注意以下几点：

- 对于有端点：
  - 可用区域级冗余内置于服务中
  - 每个可用区都有冗余实例集。
  - 这种冗余是自动提供的
- 支持 virtual private cloud (VPC) 终端节点

另请参阅

- 有关 AWS 区域 和可用区的更多信息，请参阅AWS 全球基础架构。
- 有关如何使用基于延迟的路由来构建更高的冗余并最大限度地减少网络延迟的示例，请参阅博客文章最大限度地减少服务器的网络延迟。 AWS Transfer Family

中的基础设施安全 AWS Transfer Family

作为一项托管服务 AWS Transfer Family ，受 AWS 全球网络安全的保护。有关 AWS 安全服务以及如何 AWS 保护基础设施的信息，请参阅AWS 云安全。要使用基础设施安全的最佳实践来设计您的 AWS 环境，请参阅 S AWS ecurity Pillar Well-Architected Fram ework 中的基础设施保护。

您可以使用 AWS 已发布的 API 调用 AWS Transfer Family 通过网络进行访问。客户端必须支持以下内容：

- 传输层安全性协议 (TLS) 我们要求使用 TLS 1.2，建议使用 TLS 1.3。
- 具有完全向前保密 (PFS) 的密码套件，例如 DHE（临时 Diffie-Hellman）或 ECDHE（临时椭圆曲线 Diffie-Hellman）。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 委托人关联的秘密访问密钥来对请求进行签名。或者，您可以使用 AWS Security Token Service（AWS STS）生成临时安全凭证来对请求进行签名。

是一个 Web 应用程序防火墙。

AWS WAF 是一种 Web 应用程序防火墙，可帮助保护 Web 应用程序和 API 免受攻击。通过它，您可以配置一组规则（称为 Web 访问控制列表，即 Web ACL），基于可自定义的 Web 安全规则以及您定义的条件，允许、阻止或统计 Web 请求。有关更多信息，请参阅使用保护 AWS WAF 您的 API。

要添加 AWS WAF

1. 打开 API Gateway 控制台，网址为：https://console.aws.amazon.com/apigateway/。
2. 在 API 导航窗格，选择您的自定义身份提供程序模板。
3. 选择 Stages (阶段)。
4. 在 Stages (阶段) 窗格中，选择该阶段的名称。
5. 在 Stage Editor (阶段编辑器) 窗格中，选择设置选项卡。
6. 请执行以下操作之一：

   - 在 Web 应用程序防火墙(WAF) 下，选择要与此阶段关联的区域 Web ACL。

- 如果您所需的 Web ACL 不存在，则通过以下方式创建：
  1. 选择 Create web ACL (创建 Web ACL)。
  2. 在 AWS WAF 服务主页上，选择创建 Web ACL。
  3. 在 Web ACL 详细信息，在 名称 中键入 Web ACL 的名称。
  4. 在规则，选择添加规则，然后选择添加自己的规则和规则组。
  5. 对于规则类型，选择 IP 集以标识特定 IP 地址列表。
  6. 对于规则名称，输入规则的名称。
  7. 对于 IP 集，请选择现有的 IP 集。要创建 IP 集，请参阅 CreateIPSet。
  8. 对于用作初始地址的 IP 地址，请选择标头中的 IP 地址
  9. 在 标头题字段名 中，输入`SourceIP`。
  10. 对于标头内位置，选择第一个 IP 地址。
  11. 对于缺少 IP 地址的回退，请根据标题中无效（或缺失）IP 地址的处理方式，选择匹配 或不匹配。
  12. 在 操作 中，选择 IP 集的操作。
  13. 对于不符合任何规则请求的默认 Web ACL 操作，请选择 允许或阻止，然后单击 下一步。
  14. 对于步骤 4 和 5，选择 下一步。
  15. 在 审核和创建 中，查看您的选择，然后选择创建 Web ACL。
7. 选择保存更改。
8. 选择资源。
9. 依次选择 Actions（操作）、Deploy API（部署 API）。

有关 AWS Web 应用程序防火墙安全 AWS Transfer Family 性的信息，请参阅 AWS 存储博客中的AWS Transfer Family 使用 AWS 应用程序防火墙和 Amazon API Gateway 进行保护。

防止跨服务混淆代理

混淆代理问题是一个安全性问题，即不具有操作执行权限的实体可能会迫使具有更高权限的实体执行该操作。在中 AWS，跨服务模仿可能会导致混乱的副手问题。一个服务（呼叫服务）调用另一项服务（所谓的服务）时，可能会发生跨服务模拟。可以操纵调用服务以使用其权限对另一个客户的资源进行操作，否则该服务不应有访问权限。为防止这种情况， AWS 提供可帮助您保护所有服务的数据的工具，而这些服务中的服务主体有权限访问账户中的资源。有关此问题的详细描述，请参阅 IAM 用户指南 中的 混淆代理问题。

我们建议在资源策略中使用aws:SourceArn和aws:SourceAccount全局条件上下文密钥来限制 Transf AWS er Family 对资源的权限。如果使用两个全局条件上下文键，在同一策略语句中使用时，aws:SourceAccount 值和 aws:SourceArn 值中的账户必须使用相同的账户 ID。

防止混淆代理问题最有效的方法是使用具有资源完整 Amazon Resource Name (ARN) 的 全局条件上下文键。如果不知道资源的完整 ARN，或者正在指定多个资源，请针对 ARN 未知部分使用带有通配符字符 (*) 的 aws:SourceArn 全局上下文条件键。例如，arn:aws:transfer::*region*::*account-id*:server/*。

AWS Transfer Family 使用以下类型的角色：

- 用户角色-允许服务管理的用户访问必要的 Transfer Family 资源。 AWS Transfer Family 在 Transfer Family 用户 ARN 的背景下担任此角色。
- 访问角色 - 仅提供对正在传输的 Amazon S3 文件的访问权限。对于入站 AS2 传输，访问角色使用协议的 Amazon 资源名称（ARN）。对于出站 AS2 传输，访问角色使用连接器的 ARN。
- 调用角色 – 用于作为服务器自定义身份提供程序的 Amazon API Gateway。Transfer Family 在 Transfer Family 服务器 ARN 的背景下扮演这个角色。
- 日志角色-用于将条目登录到 Amazon CloudWatch。Transfer Family 使用此角色记录成功和失败的详细信息以及有关文件传输的信息。Transfer Family 在 Transfer Family 服务器 ARN 的背景下扮演这个角色。对于出站 AS2 传输，日志角色使用连接器 ARN。
- 执行角色 - 允许 Transfer Family 用户调用和启动工作流程。Transfer Family 在 Transfer Family 工作流程 ARN 的背景下担任此角色。

有关更多信息，请参阅《 IAM 用户指南》中的 IAM 中的策略和权限。

> **Note**
>
> 在以下示例中，将每个 *user input placeholder* 替换为您自己的信息。

> **Note**
>
> 在我们的示例中，我们同时使用 ArnLike 和 ArnEquals。它们在功能上是相同的，因此您可以在制定策略时使用其中任何一个。Transfer Family 文档在条件包含通配符时使用ArnLike，ArnEquals用于表示完全匹配的条件。

AWS Transfer Family 用户角色跨服务混淆副手预防

以下示例日志记录/调用策略允许账户中的任何服务器担任该角色。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "transfer.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "account-id"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:transfer:region:account-id:user/*"
                }
            }
        }
    ]
}
```

以下示例策略允许任何特定服务器用户承担此角色。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "transfer.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "account-id"
                },
                "ArnEquals": {
```

```
                    "aws:SourceArn": "arn:aws:transfer:region:account-id:user/server-
id/*"
                }
            }
        }
    ]
}
```

以下示例策略允许任何特定服务器用户承担此角色。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "transfer.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:transfer:region:account-id:user/server-
id/user-name"
                }
            }
        }
    ]
}
```

AWS Transfer Family 工作流程角色跨服务混乱副手预防

以下示例日志记录/调用策略允许账户中的任何服务器担任该角色。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "transfer.amazonaws.com"
```

```
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "account-id"
                },
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:transfer:region:account-id:workflow/*"
                }
            }
        }
    ]
}
```

以下示例策略允许任何特定服务器用户承担此角色。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Principal": {
                "Service": "transfer.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "ArnLike": {
                    "aws:SourceArn": "arn:aws:transfer:region:account-
id:workflow/workflow-id"
                }
            }
        }
    ]
}
```

AWS Transfer Family 日志和调用角色跨服务混淆了副手预防

> **Note**
>
> 以下示例可用于日志记录和调用角色。

在这些示例中，如果您的服务器未附加任何工作流程，则可以删除该工作流程的 ARN 详细信息。

以下示例日志/调用策略允许账户中的任何服务器（和工作流程）代入该角色。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowAllServersWithWorkflowAttached",
            "Effect": "Allow",
            "Principal": {
                "Service": "transfer.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "account-id"
                },
                "ArnLike": {
                   "aws:SourceArn": [
                     "arn:aws:transfer:region:account-id:server/*",
                     "arn:aws:transfer:region:account-id:workflow/*"
                   ]
                }
            }
        }
    ]
}
```

以下示例日志/调用策略允许特定的服务器（和工作流程）担任该角色。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSpecificServerWithWorkflowAttached",
            "Effect": "Allow",
            "Principal": {
                "Service": "transfer.amazonaws.com"
            },
```

```
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "account-id"
                },
                "ArnEquals": {
                   "aws:SourceArn": [
                       "arn:aws:transfer:region:account-id:server/server-id",
                       "arn:aws:transfer:region:account-id:workflow/workflow-id"
                   ]
                }
            }
        }
    ]
}
```

AWS Transfer Famil AWS y 的托管政策

要向用户、群组和角色添加权限，使用 AWS 托管策略比自己编写策略要容易得多。创建 AWS Identity and Access Management (IAM) 客户托管策略仅向您的团队提供他们所需的权限需要时间和专业知识。要快速入门，您可以使用我们的 AWS 托管策略。这些策略涵盖常见使用案例，可在您的 AWS 账户中使用。有关 AWS 托管策略的更多信息，请参阅《IAM 用户指南》中的AWS 托管策略。有关所有 AWS 托管策略的详细列表，请参阅AWS 托管策略参考指南。

AWS 服务维护和更新 AWS 托管策略。您无法更改 AWS 托管策略中的权限。服务偶尔会向 AWS 托管式策略添加额外权限以支持新功能。此类更新会影响附加策略的所有身份（用户、组和角色）。当启动新功能或新操作可用时，服务最有可能会更新 AWS 托管式策略。服务不会从 AWS 托管策略中移除权限，因此策略更新不会破坏您的现有权限。

此外，还 AWS 支持跨多个服务的工作职能的托管策略。例如，`ReadOnlyAccess` AWS 托管策略提供对所有 AWS 服务和资源的只读访问权限。当服务启动一项新功能时， AWS 会为新操作和资源添加只读权限。有关工作职能策略的列表和说明，请参阅《IAM 用户指南》中的适用于工作职能的AWS 托管式策略。

AWS 托管策略： AWSTransferConsoleFullAccess

该`AWSTransferConsoleFullAccess`政策提供通过 AWS 管理控制台对 Transfer Family 的完全访问权限。

权限详细信息

该策略包含以下权限。

- 授予权限以检索证书 ARN 以及每个 ARN 的域名列表
- 授予权限以描述一个或多个弹性 IP 地址
- 授予权限以描述可供您使用的一个或多个可用区
- 授予权限以描述一个或多个网络接口
- 授予权限以描述一个或多个安全组
- 授予权限以描述一个或多个子网
- 授予权限以描述一个或多个虚拟私有网关
- 授予权限以描述一个或多个 VPC 终端节点
- 返回每种事件类型的（问题、计划的更改和账户通知）事件数。
- 授予权限以检索有关指定托管策略的版本的信息，包括策略文档
- 授予权限以列出所有托管策略
- 授予权限以列出具有指定路径前缀的 IAM 角色
- `iam:PassRole` — 授予将 IAM 角色传递至 Transfer Family 的权限。有关更多详细信息，请参阅向用户授予将角色传递给的权限 AWS 服务。
- 授予权限以获取与当前 关联的公有和私有托管区域列表
- 授予权限以列出该请求的经身份验证的发件人拥有的所有桶
- `transfer:*` — 授予对 Transfer Family 资源的访问权限。星号 (*) 授权访问所有资源。

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "transfer.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
```

```
                "acm:ListCertificates",
                "ec2:DescribeAddresses",
                "ec2:DescribeAvailabilityZones",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeSecurityGroups",
                "ec2:DescribeSubnets",
                "ec2:DescribeVpcs",
                "ec2:DescribeVpcEndpoints",
                "health:DescribeEventAggregates",
                "iam:GetPolicyVersion",
                "iam:ListPolicies",
                "iam:ListRoles",
                "route53:ListHostedZones",
                "s3:ListAllMyBuckets",
                "transfer:*"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS 托管策略： AWSTransferFullAccess

此 AWSTransferFullAccess 策略提供对 Transfer Family 服务的完全访问权限。

权限详细信息

该策略包含以下权限。

- transfer:* — 授予对 Transfer Family 资源的访问权限。星号 (*) 授权访问所有资源。
- iam:PassRole — 授予将 IAM 角色传递至 Transfer Family 的权限。有关更多详细信息，请参阅向用户授予将角色传递给的权限 AWS 服务。
- 授予权限以描述一个或多个弹性 IP 地址
- 授予权限以描述一个或多个网络接口
- 授予权限以描述一个或多个 VPC 终端节点

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

```
            "Effect": "Allow",
            "Action": "transfer:*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "iam:PassRole",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "iam:PassedToService": "transfer.amazonaws.com"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVpcEndpoints",
                "ec2:DescribeNetworkInterfaces",
                "ec2:DescribeAddresses"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS 托管策略： AWSTransferLoggingAccess

该AWSTransferLoggingAccess政策授予 T AWS ransfer Family 创建日志流和群组以及将日志事件存入您的账户的完全访问权限。

权限详细信息

此策略包括以下权限 Amazon CloudWatch Logs。

- CreateLogStream — 授权主体创建日志流。
- DescribeLogStreams— 授权主体列出日志组的日志流。
- CreateLogGroup — 授权主体创建日志流。
- 授予权限以将一批日志事件上传到指定的日志流

```
{
```

```
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:DescribeLogStreams",
                "logs:CreateLogGroup",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS 托管策略： AWSTransferReadOnlyAccess

此 AWSTransferReadOnlyAccess 策略提供对 Transfer Family 服务的完全访问权限。

权限详细信息

此策略包含以下权限。

- DescribeUser — 授权主体查看用户描述。
- DescribeServer— 授权主体查看服务器描述。
- ListUsers — 授予主体列出服务器用户。
- ListServers — 授权主体列出账户服务器。
- TestIdentityProvider— 授权主体策略配置身份提供程序是否设置得当。
- 授予列出 Amazon Forecast 资源的标签的权限

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "transfer:DescribeUser",
                "transfer:DescribeServer",
```

```
                "transfer:ListUsers",
                "transfer:ListServers",
                "transfer:TestIdentityProvider",
                "transfer:ListTagsForResource"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS 将 Family 更新转移到 AWS 托管政策

查看 Transfer Family AWS 托管政策自该服务开始跟踪这些变更以来这些更新的详细信息。 AWS 要获得有关此页面更改的自动提示，请订阅 的文档历史记录 AWS Transfer Family 页面上的 RSS 源。

更改	描述	日期
文档更新	为每个 Transfer Family 托管策略添加章节。	2022 年 1 月 27 日
AWSTransferReadOnlyAccess – 更新了现有策略	AWS Transfer Family 添加了允许读取策略的新权限 AWS Managed Microsoft AD。	2021 年 9 月 30 日
AWS Transfer Family 开始追踪变更	AWS Transfer Family 开始跟踪其 AWS 托管政策的变更。	2021 年 6 月 15 日

故障排除 AWS Transfer Family

使用以下信息来帮助您诊断和修复在处理时可能遇到的常见问题 AWS Transfer Family。

有关 Transfer Family 中的 IAM 问题，请参阅对 AWS Transfer Family 身份和访问进行故障排除。

主题

对服务托管用户进行故障排除

本部分介绍了以下问题的可能解决方案。

主题

对 Amazon EFS 服务托管用户进行故障排除

描述

运行sftp命令时，但提示符未出现，而是会看到以下消息：

```
Couldn't canonicalize: Permission denied
           Need cwd
```

原因

您的 AWS Identity and Access Management (IAM) 用户的角色无权访问亚马逊 Elastic File System (Amazon EFS)。

解决方案

增加用户角色的策略权限。您可以添加 AWS 托管策略，例如AmazonElasticFileSystemClientFullAccess。

对公有密钥正文过长进行故障排除

描述

您在尝试创建服务托管用户时收到以下错误：

```
Failed to create user (1 validation error detected:
'sshPublicKeyBody' failed to satisfy constraint: Member must have length less than or
 equal to 2048)
```

原因

您可能正在为公钥正文输入 PGP 密钥，并且 AWS Transfer Family 不支持服务托管用户的 PGP 密钥。

解决方案

如果 PGP 密钥是基于 RSA 的，则可以将其转换为 PEM 格式。例如，Ubuntu 在这里提供了一个转换工具：https://manpages.ubuntu.com/manpages/xenial/man1/openpgp2ssh.1.html

对添加 SSH 公有密钥失败进行故障排除

描述

您在尝试为服务托管用户添加公有密钥时收到以下错误：

```
Failed to add SSH public key (Unsupported or invalid SSH public key format)
```

原因

您可能正在尝试导入 SSH2 格式的公钥，但服务托管用户 AWS Transfer Family 不支持 SSH2 格式的公钥。

解决方案

您需要将密钥转换为 OpenSSH 格式。将 SSH2 公钥转换至 PEM 格式 中介绍了此过程。

对 Amazon API Gateway 问题进行故障排除

本节介绍以下 API Gateway 问题的可能解决方案。

主题

身份验证失败次数过多

描述

当您尝试使用 Secure Shell (SSH) 文件传输协议 (SFTP) 连接到服务器时，会出现以下错误：

```
Received disconnect from 3.15.127.197 port 22:2: Too many authentication failures
Authentication failed.
Couldn't read packet: Connection reset by peer
```

原因

您可能输入了错误的用户密码。请重试输入正确的密码。

如果密码正确，则问题可能是由角色 Amazon 资源名称 (ARN) 无效引起的。要确认这是问题所在，请测试服务器的身份提供商。如果您看到类似于以下内容的响应，则角色 ARN 仅为占位符，如全部为零的角色 ID 值所示：

```
{
    "Response": "{\"Role\": \"arn:aws:iam::000000000000:role/MyUserS3AccessRole\",
\"HomeDirectory\": \"/\"}",
    "StatusCode": 200,
    "Message": "",
```

```
    "Url": "https://api-gateway-ID.execute-api.us-east-1.amazonaws.com/prod/
servers/transfer-server-ID/users/myuser/config"
}
```

解决方案

将占位符角色 ARN 替换为有权访问服务器的实际角色。

更新角色

1. 打开 AWS CloudFormation 控制台，网址为 https://console.aws.amazon.com/cloudformation。
2. 在左侧导航窗格中，选择 Stacks (堆栈)。
3. 在堆栈列表中，选择您的堆栈，然后选择参数选项卡。
4. 选择更新。在更新堆栈页面上，选择使用当前模板，然后选择下一步。
5. 替换为UserRoleArn具有足够权限访问您的 Transfer Family 服务器的角色 ARN。

> **Note**
>
> 要授予必要的权限，您也可以将AmazonAPIGatewayAdministrator和AmazonS3FullAccess托管策略添加到角色中。

6. 选择下一步，然后再次选择下一步。在查看##页面上，选择我确认 AWS CloudFormation 可能会创建 IAM 资源，然后选择更新堆栈。

连接关闭

描述

当您尝试使用 Secure Shell (SSH) 文件传输协议 (SFTP) 连接到服务器时，会出现以下错误：

```
Connection closed
```

原因

造成此问题的一个可能原因是，您的亚马逊 CloudWatch 日志角色与 Transfer Family 没有信任关系。

解决方案

确保服务器的日志记录角色与 Transfer Family 具有信任关系。有关更多信息，请参阅 建立信任关系。

对加密 Amazon S3 存储桶的策略进行故障排除

描述

您有一个加密的 Amazon S3 存储桶，用作您的 Transfer Family 服务器的存储空间。如果您尝试将文件上传到服务器，则会收到错误消息`Couldn't close file: Permission denied`。

而且，如果您查看服务器日志，则会看到以下错误：

```
ERROR Message="Access denied" Operation=CLOSE Path=/bucket/user/test.txt BytesIn=13
ERROR Message="Access denied"
```

原因

您的 IAM 用户的策略没有权限访问加密存储桶。

解决方案

您必须在策略中指定其他权限才能授予所需的 AWS Key Management Service (AWS KMS) 权限。有关更多信息，请参阅 Amazon S3 中的数据加密。

对身份认证问题进行故障排除

本节介绍以下身份验证问题的可能解决方案。

主题

- 身份验证失败 — SSH/SFTP
- 托管 AD 领域不匹配问题
- 其他身份验证问题

身份验证失败 — SSH/SFTP

描述

当您尝试使用 Secure Shell (SSH) 文件传输协议 (SFTP) 连接到服务器时，会收到类似于以下内容的消息：

```
Received disconnect from 3.130.115.105 port 22:2: Too many authentication failures
```

```
Authentication failed.
```

> **Note**
>
> 如果您使用的是 API Gateway 并收到此错误，请参阅身份验证失败次数过多。

原因

您尚未为用户添加 RSA 密钥对，因此必须改用密码进行身份验证。

解决方案

运行该`sftp`命令时，请指定`-o PubkeyAuthentication=no`选项。此选项会强制系统请求您的密码。例如：

```
sftp -o PubkeyAuthentication=no sftp-user@server-id.server.transfer.region-id.amazonaws.com
```

托管 AD 领域不匹配问题

描述

用户的领域和他们的组领域必须匹配。它们必须都在默认领域中，或者它们都必须位于可信领域。

原因

如果用户及其组不匹配，则无法通过 Transfer Family 对该用户进行身份验证。如果您测试用户的身份提供商，则会收到错误找不到用户组的关联访问权限。

解决方案

引用用户领域中与组领域（默认或可信）相匹配的组。

其他身份验证问题

描述

您收到身份验证错误，但其他故障排除均无效

原因

您可能已经为包含前导或尾部斜杠 (/) 的逻辑目录指定了目标。

解决方案

更新您的逻辑目录目标，确保它以斜杠开头，并且不包含尾部斜杠。例如，/*DOC-EXAMPLE-BUCKET*/images可以接受*DOC-EXAMPLE-BUCKET*/images，但不/*DOC-EXAMPLE-BUCKET*/images/是。

对托管工作流程问题进行故障排除

本节介绍以下工作流程问题的可能解决方案。

主题

- 使用 Amazon 解决与工作流程相关的错误 CloudWatch
- 对工作流程复制错误进行故障排除

使用 Amazon 解决与工作流程相关的错误 CloudWatch

描述

如果您的工作流程出现问题，可以使用 Amazon CloudWatch 来调查原因。

原因

可能有多种原因。使用 Amazon CloudWatch 日志进行调查。

解决方案

Transfer Family 会将工作流程执行状态发送到 CloudWatch 日志中。 CloudWatch 日志中可能会出现以下类型的工作流程错误：

- `"type": "StepErrored"`
- `"type": "ExecutionErrored"`
- `"type": "ExecutionThrottled"`
- `"Service failure on starting workflow"`

您可以使用不同的筛选器和模式语法来筛选工作流程的执行日志。例如，您可以在日志中创建日志过滤器，以捕获包含该ExecutionErrored消息的工作流程执行日志。 CloudWatch 有关详细信息，请参阅 Amazon Log CloudWatch s 用户指南中的使用订阅实时处理日志数据以及筛选和模式语法。

StepErrored

```
2021-10-29T12:57:26.272-05:00
            {"type":"StepErrored","details":
{"errorType":"BAD_REQUEST","errorMessage":"Cannot
            tag Efs file","stepType":"TAG","stepName":"successful_tag_step"},
            "workflowId":"w-
abcdef01234567890","executionId":"1234abcd-56ef-78gh-90ij-1234klmno567",
            "transferDetails":
{"serverId":"s-1234567890abcdef0","username":"lhr","sessionId":"1234567890abcdef0"}
```

此处，StepErrored表示工作流程中的某个步骤产生了错误。在单个工作流程中，您可以配置多个步骤。此错误会告诉您错误发生在哪个步骤中，并提供错误消息。在此特定示例中，该步骤配置为标记文件；但是，不支持在 Amazon EFS 文件系统中标记文件，因此该步骤生成了一处错误。

ExecutionErrored

```
2021-10-29T12:57:26.618-05:00
            {"type":"ExecutionErrored","details":{},"workflowId":"w-w-
abcdef01234567890",
            "executionId":"1234abcd-56ef-78gh-90ij-1234klmno567","transferDetails":
{"serverId":"s-1234567890abcdef0",
            "username":"lhr","sessionId":"1234567890abcdef0"}}
```

当工作流程无法执行任何步骤时，它会生成ExecutionErrored消息。例如，如果您在给定工作流程中配置了单个步骤，并且该步骤无法执行，则整个工作流程将失败。

Executionthrottled

如果工作流程的触发速度超过系统所能支持的速度，则执行受到限制。此日志消息表明您必须降低工作流程的执行速度。如果您无法降低工作流程执行率，请通过 Contact 联系 AWS Support 。 AWS

启动工作流程时服务失败

无论何时从服务器上移除工作流程并用新的工作流程替换它，或者更新服务器配置（这会影响工作流程的执行角色），都必须等待大约 10 分钟才能执行新的工作流程。Transfer Family 服务器会缓存工作流程细节，服务器需要 10 分钟才能刷新其缓存。

此外，您必须注销所有活动的 SFTP 会话，然后等待 10 分钟重新登录才能看到更改。

对工作流程复制错误进行故障排除

描述

如果您正在执行的工作流程中包含复制已上传文件的步骤，则可能会遇到以下错误：

```
{
  "type": "StepErrored", "details": {
    "errorType": "BAD_REQUEST", "errorMessage": "Bad Request (Service: Amazon S3;
 Status Code: 400; Error Code: 400 Bad Request;
    Request ID: request-ID; S3 Extended Request ID: request-ID Proxy: null)",
 "stepType": "COPY", "stepName": "copy-step-name" },
  "workflowId": "workflow-ID",
  "executionId": "execution-ID",
  "transferDetails": {
     "serverId": "server-ID",
     "username": "user-name",
     "sessionId": "session-ID"
  }
}
```

原因

源文件位于与目标存储桶不同 AWS 区域 的 Amazon S3 存储桶中。

解决方案

如果您正在执行包含复制步骤的工作流程，请确保源存储桶和目标存储桶位于同一 AWS 区域存储桶中。

对工作流程解密问题进行故障排除

本节介绍加密工作流程中以下问题的可能解决方案。

主题

- 对签名加密文件出现错误进行故障排除
- 对 FIPS 算法的错误进行故障排除

对签名加密文件出现错误进行故障排除

描述

您的解密工作流程失败，并且您收到以下错误：

```
"Encrypted file with signed message unsupported"
```

原因

Transfer Family 目前不支持对加密文件进行签名。

解决方案

在您的 PGP 客户端中，如果可以选择对加密文件进行签名，请务必清除该选项，因为 Transfer Family 目前不支持对加密文件进行签名。

对 FIPS 算法的错误进行故障排除

描述

解密工作流程失败，日志消息如下所示：

```
{
   "type": "StepErrored",
   "details": {
      "errorType": "BAD_REQUEST",
      "errorMessage": "File encryption algorithm not supported with FIPS mode
 enabled.",
      "stepType": "DECRYPT",
      "stepName": "step-name"
   },
   "workflowId": "workflow-ID",
   "executionId": "execution-ID",
   "transferDetails": {
      "serverId": "server-ID",
      "username": "user-name",
      "sessionId": "session-ID"
   }
}
```

原因

Transfer Family 服务器已启用 FIPS 模式和相关的解密工作流程步骤。在上传到 Transfer Family 服务器之前对文件进行加密时，加密客户端可能会生成使用非 FIPS 批准的对称加密算法的加密文件。在这种情况下，工作流程无法解密文件。在以下示例中，GnuPG 版本 2.4.0 使用 OCB（一种非 FIPS 分组密码模式）来加密文件：这会导致工作流程失败。

解决方案

您必须编辑用于加密文件的 GPG 密钥，然后对其重新加密。以下过程描述了您必须采取的步骤。

编辑 PGP 密钥

1. 通过运行`gpg --list-keys`来确定必须编辑的密钥

   这将返回密钥列表。每个密钥的详细信息类似于以下内容：

   ```
   pub   ed25519 2022-07-07 [SC]
         wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
   uid           [ultimate] Mary Major <marymajor@example.com>
   sub   cv25519 2022-07-07 [E]
   ```

2. 标识要编辑的密钥。在上一步所示的示例中，ID 为`wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`。

3. 运行 `gpg --edit-key wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`。

   系统以有关 GnuPG 程序和指定密钥的详细信息进行响应。

4. 在`gpg>`提示符下，输入`showpref`。将返回以下详细信息：

   ```
   [ultimate] (1). Mary Major <marymajor@example.com>
      Cipher: AES256, AES192, AES, 3DES
      AEAD: OCB
      Digest: SHA512, SHA384, SHA256, SHA224, SHA1
      Compression: ZLIB, BZIP2, ZIP, Uncompressed
      Features: MDC, AEAD, Keyserver no-modify
   ```

   请注意，已列出存储在密钥上的首选算法。

5. 我们希望编辑密钥以保留除 OCB 之外的所有算法。运行`setpref`命令，指定要保留的所有算法：

   ```
   gpg> setpref AES256, AES192,AES,3DES,SHA512, SHA384, SHA256, SHA224, SHA1,ZLIB,
    BZIP2, ZIP, Uncompressed
   ```

   这将返回以下详细信息：

   ```
   Set preference list to:
        Cipher: AES256, AES192, AES, 3DES
        AEAD:
   ```

```
     Digest: SHA512, SHA384, SHA256, SHA224, SHA1
     Compression: ZLIB, BZIP2, ZIP, Uncompressed
     Features: MDC, Keyserver no-modify
Really update the preferences? (y/N)
```

6. 输入y进行更新，然后在系统提示确认更改时输入密码。
7. 保存更改。

```
gpg> save
```

在重新运行解密工作流程之前，必须使用编辑后的密钥重新加密文件。

对 Amazon EFS 问题进行故障排除

本节介绍以下 Amazon EFS 问题的可能解决方案。

主题

- 对缺失 POSIX 配置文件进行故障排除
- 使用 Amazon EFS 逻辑目录进行故障排除

对缺失 POSIX 配置文件进行故障排除

描述

如果您在服务器上使用 Amazon EFS 存储，并且使用自定义身份提供商，则必须为 AWS Lambda 函数提供 POSIX 配置文件。

原因

一个可能的原因是，我们为创建 AWS Lambda支持的 Amazon API Gateway 方法提供的模板目前不包含 POSIX 信息。

如果您确实提供了 POSIX 信息，那么 Transfer Family 可能无法正确解析您用于提供 POSIX 信息的格式。

解决方案

请务必向 Transfer Family 提供`PosixProfile`参数的 JSON 元素。

例如，如果您使用的是 Python，则可以在解析PosixProfile参数的位置添加以下行：

```
if PosixProfile:
        response_data["PosixProfile"] = json.loads(PosixProfile)
```

或者 JavaScript，可以在中添加以下行，其中*uid-value*和*gid-value*是分别代表用户 ID (UID) 和组 ID (GID) 的 0 或大于 0 的整数：

```
PosixProfile: {"Uid": uid-value, "Gid": gid-value},
```

这些代码示例将PosixProfile参数作为 JSON 对象而非字符串发送到 Transfer Family。

此外 AWS Secrets Manager，您还必须按如下方式存储PosixProfile参数。将*your-uid*和*your-gid*替换为您的 GID 和 UID 的实际值。

```
{"Uid": your-uid, "Gid": your-gid, "SecondaryGids": []}
```

使用 Amazon EFS 逻辑目录进行故障排除

描述

如果用户的主目录不存在，并且他们运行了ls命令，则系统会按如下方式做出响应：

```
sftp> ls
remote readdir ("/"): No such file or directory
```

原因

如果 Transfer Family 服务器使用 Amazon EFS，则必须先创建具有读写访问权限的用户主目录，然后用户才能在其逻辑主目录中工作。用户无法自己创建此目录，因为他们将缺乏mkdir对逻辑主目录的权限。

解决方案

对父目录具有管理访问权限的用户需要创建该用户的逻辑主目录。

对测试您的身份提供商进行故障排除

描述

如果您使用控制台或TestIdentityProvider API 调用测试身份提供商，则该Response字段为空。例如：

```
{
    "Response": "{}",
    "StatusCode": 200,
    "Message": ""
}
```

原因

最可能的原因是用户名或密码不正确导致身份验证失败。

解决方案

确保您使用的是正确的用户凭证，并在必要时更新用户名或密码。

为您的 SFTP 连接器添加可信主机密钥进行故障排除

描述

创建或编辑 SFTP 连接器并添加可信主机密钥时，您会收到以下错误：`Failed to edit connector details (Invalid host key format.)`

原因

如果您粘贴了正确的公钥，则问题可能在于您包含了密钥的comment部分。 AWS Transfer Family 目前不接受密钥的注释部分。

解决方案

将密钥的注释部分粘贴到文本字段中时，将其删除。例如，假设您的密钥可能如下所示：

```
ssh-rsa AAAA...== marymajor@dev-dsk-marymajor-1d-c1234567.us-east-1.amazon.com
```

删除==字符后面的文本，然后仅粘贴密钥中直至并包括==的部分。

```
ssh-rsa AAAA...==
```

文件上传问题进行故障排除

本节介绍以下文件上传问题的可能解决方案。

主题

对 Amazon S3 文件上传错误进行故障排除

描述

当您尝试使用 Transfer Family 将文件上传到 Amazon S3 存储空间时，您会收到如下错误消息：AWS Transfer 不支持对 S3 对象进行随机访问写入。

原因

当您使用 Amazon S3 作为服务器存储空间时，Transfer Family 不支持单次传输的多个连接。

解决方案

如果您的 Transfer Family 服务器使用 Amazon S3 进行存储，请禁用任何提及使用多个连接进行单次传输的客户端软件选项。

对无法读取的文件名称进行故障排除

描述

您会在上传的某些文件中看到文件名已损坏。用户有时会在 FTP 和 SFTP 传输中遇到问题，这些问题会破坏文件名中的某些字符，例如变音符号、重音字母或某些脚本，例如中文或阿拉伯语。

原因

尽管 FTP 和 SFTP 协议允许客户端协商文件名的字符编码，但 Amazon S3 和 Amazon EFS 却不允许。相反，它们需要 UTF-8 字符编码。因此，某些字符无法正确呈现。

解决方案

要解决此问题，请查看您的客户端应用程序中的文件名字符编码，并确保将其设置为 UTF-8。

对ResourceNotFound异常进行故障排除

描述

您会收到一条找不到资源的错误。例如，如果您运行UpdateServer，可能会收到如下错误：

```
An error occurred (ResourceNotFoundException) when calling the UpdateServer operation:
 Unknown server
```

原因

收到ResourceNotFoundException消息的原因有多种。在大多数情况下，您在 API 命令中指定的资源不存在。如果您确实指定了现有资源，那么最有可能的原因是您的默认区域与资源的区域不同。例如，如果您的默认区域为 us-east-1，而您的 Transfer Family 服务器在 us-east-2 中，则您将收到未知资源异常。

有关设置默认区域的详细信息，请参阅使用aws configure进行快速配置。

解决方案

在 API 命令中添加区域参数，以明确指定在何处查找特定资源。

```
aws transfer -describe-server --server-id server-id --region us-east-2
```

对 SFTP 连接器问题进行故障排除

本节介绍以下 SFTP 连接器问题的可能解决方案。

主题

- 密钥协商失败
- 其他 SFTP 连接器问题

密钥协商失败

描述

您会收到密钥交换协商失败的错误。例如：

```
Key exchange negotiation failed due to incompatible host key algorithms.
```

```
Client offered: [ecdsa-sha2-nistp256, ecdsa-sha2-nistp384,
ecdsa-sha2-nistp521, rsa-sha2-512, rsa-sha2-256] Server offered: [ssh-rsa]
```

原因

出现此错误是因为服务器支持的主机密钥算法与连接器支持的主机密钥算法之间没有重叠。

解决方案

确保远程服务器支持错误消息中列出的至少一种客户端主机密钥算法。有关支持的算法列表，请参阅AWS Transfer Family SFTP 连接器的安全策略。

其他 SFTP 连接器问题

描述

运行后您会收到错误`StartFileTransfer`，但不知道问题的原因，并且在 API 调用后只返回连接器 ID。

原因

此错误可能有多种原因。要排除故障，我们建议您测试连接器并搜索 CloudWatch 日志。

解决方案

- 测试您的连接器：请参阅测试 SFTP 连接器。如果测试失败，系统会根据测试失败的原因提供错误消息。该部分介绍如何通过控制台或使用 TestConnectionAPI 命令测试您的连接器。
- 查看您的连接器的 CloudWatch 日志：请参阅SFTP 连接器的日志条目示例。本主题提供了 SFTP 连接器日志条目的示例，以及命名惯例，以帮助您找到相应的日志。

对 AS2 问题进行故障排除

此处描述了启用适用性声明 2 (AS2) 的服务器的错误消息和故障排除提示：AS2 错误代码。

API 参考

以下各节记录了 AWS Transfer Family API 服务调用、数据类型、参数和错误。

主题

欢迎使用 AWS Transfer Family API

AWS Transfer Family 是一项安全的传输服务，您可以使用它通过以下协议将文件传入和传出亚马逊简单存储服务 (Amazon S3) Storage Service 存储：

- Secure Shell (SSH) 文件传输协议 (SFTP)
- 安全文件传输协议 (FTPS)
- 文件传输协议 (FTP)
- 适用性声明 2 (AS2)

文件传输协议用于不同行业的数据交换工作流程，例如金融服务、医疗保健、广告和零售等。 AWS Transfer Family 简化了将文件传输工作流迁移到 AWS。

要使用该 AWS Transfer Family 服务，您需要在您选择的 AWS 区域中实例化服务器。您可以创建服务器，列出可用服务器，并更新和删除服务器。服务器是向其请求文件操作的实体 AWS Transfer Family。服务器有许多重要的属性。服务器是由系统分配的 `ServerId` 标识符所标识的命名实例。您可以选择为服务器分配主机名，甚至分配自定义主机名。该服务针对任何实例化的服务器（即使服务器处于 `OFFLINE` 状态）以及传输的数据量进行收费。

用户必须知道请求文件操作的服务器。由用户名标识的用户被分配给服务器。
用户名用于对请求进行身份验证。一个服务器只能有一个身份验证方法，即 `AWS_DIRECTORY_SERVICE`、`SERVICE_MANAGED`、`AWS_LAMBDA`、或 `API_GATEWAY`.

您可以使用以下任何一种身份提供程序类型来对用户进行身份验证：

- 对于 SERVICE_MANAGED，SSH 公有密钥与服务器上的用户属性一起存储。用户可以拥有文件的一个或多个 SSH 公有密钥，用于 SERVICE_MANAGED 身份验证方法。当客户端请求 SERVICE_MANAGED 方法的文件操作时，客户端提供用户名和 SSH 私有密钥，该私有密钥经过身份验证并提供访问权限。
- 您可以通过选择AWS_DIRECTORY_SERVICE身份验证方法来管理用户对 Microsoft Active Directory 组的身份验证和访问权限。
- 您可以使用连接到自定义身份提供商 AWS Lambda。选择AWS_LAMBDA身份验证方法。
- 您还可以使用提供用户身份验证和访问的自定义身份验证方法对用户请求进行身份验证。此方法依赖于 Amazon API Gateway 来使用您身份提供商的 API 调用对用户请求进行身份验证。此方法在 API 调用中称为 API_GATEWAY，在控制台中称为 自定义。您可以使用此自定义方法根据目录服务、数据库名称/密码对或其他某个机制对用户进行身份验证。

为用户分配一个策略，该策略在用户与 Amazon S3 存储桶之间具有信任关系。用户可能能够访问全部或部分存储桶。为了使服务器代表用户，服务器必须从用户继承信任关系。创建一个包含信任关系的 AWS Identity and Access Management (IAM) 角色，并为该角色分配 AssumeRole 操作。然后，服务器可以像用户一样执行文件操作。

具有 home 目录属性集的用户将使该目录（或文件夹）充当文件操作的目标和源。如果未设置 home 目录，则存储桶的 root 目录将成为登录目录。

服务器、用户和角色均由其 Amazon 资源名称 (ARN) 标识。您可以为具有 ARN 的实体分配标签（键值对）。标签是可用于分组或搜索这些实体的元数据。标签有用的一个例子是用于会计目的。

在 AWS Transfer Family ID 格式中应遵守以下惯例：

- ServerId 值采用 s-01234567890abcdef 形式。
- SshPublicKeyId 值采用 key-01234567890abcdef 形式。

Amazon 资源名称 (ARN) 格式采用以下形式：

- 对于服务器，ARN 采用 arn:aws:transfer:*region*:*account-id*:server/*server-id* 形式。

  服务器 ARN 的示例是：arn:aws:transfer:us-east-1:123456789012:server/s-01234567890abcdef。

- 对于用户，ARN 采用 `arn:aws:transfer:region:account-id:user/server-id/username` 形式。

  例如，`arn:aws:transfer:us-east-1:123456789012:user/s-01234567890abcdef/user1`。

正在使用的 DNS 条目（端点）如下所示：

- API 终端节点采用 `transfer.region.amazonaws.com` 形式。
- 服务器终端节点采用 `server.transfer.region.amazonaws.com` 形式。

有关按 AWS 地区划分的 Transfer Family 终端节点列表，请参阅中的AWS Transfer Family 终端节点和配额AWS 一般参考。

此 API 接口参考 AWS Transfer Family 包含可用于管理的编程接口的文档 AWS Transfer Family。参考结构如下所示：

- 有关 API 操作的字母顺序列表，请参阅Actions。
- 有关数据类型的字母顺序列表，请参阅Data Types。
- 有关常用查询参数的列表，请参阅常用参数。
- 有关错误代码的描述，请参阅常见错误。

> **Tip**
>
> 您可以将`--generate-cli-skeleton`参数与任何 API 调用一起使用来生成和显示参数模板，而不是实际运行命令。然后，您可以使用生成的模板进行自定义，并将其用作后续命令的输入。有关详细信息，请参阅生成并使用参数骨架文件。

操作

支持以下操作：

- CreateAccess
- CreateAgreement
- CreateConnector

- CreateProfile
- CreateServer
- CreateUser
- CreateWorkflow
- DeleteAccess
- DeleteAgreement
- DeleteCertificate
- DeleteConnector
- DeleteHostKey
- DeleteProfile
- DeleteServer
- DeleteSshPublicKey
- DeleteUser
- DeleteWorkflow
- DescribeAccess
- DescribeAgreement
- DescribeCertificate
- DescribeConnector
- DescribeExecution
- DescribeHostKey
- DescribeProfile
- DescribeSecurityPolicy
- DescribeServer
- DescribeUser
- DescribeWorkflow
- ImportCertificate
- ImportHostKey
- ImportSshPublicKey
- ListAccesses
- ListAgreements

- ListCertificates
- ListConnectors
- ListExecutions
- ListHostKeys
- ListProfiles
- ListSecurityPolicies
- ListServers
- ListTagsForResource
- ListUsers
- ListWorkflows
- SendWorkflowStepState
- StartFileTransfer
- StartServer
- StopServer
- TagResource
- TestConnection
- TestIdentityProvider
- UntagResource
- UpdateAccess
- UpdateAgreement
- UpdateCertificate
- UpdateConnector
- UpdateHostKey
- UpdateProfile
- UpdateServer
- UpdateUser

CreateAccess

管理员使用它来选择目录中哪些组有权使用AWS Transfer Family通过已启用的协议上传和下载文件。例如，Microsoft Active Directory 可能包含 50,000 个用户，但只有一小部分用户可能需要能够将文件传输到服务器。管理员可以使用`CreateAccess`将访问权限限制为需要此功能的正确用户组。

请求语法

```
{
   "ExternalId": "string",
   "HomeDirectory": "string",
   "HomeDirectoryMappings": [
      {
         "Entry": "string",
         "Target": "string",
         "Type": "string"
      }
   ],
   "HomeDirectoryType": "string",
   "Policy": "string",
   "PosixProfile": {
      "Gid": number,
      "SecondaryGids": [ number ],
      "Uid": number
   },
   "Role": "string",
   "ServerId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

ExternalId

识别目录中特定组所需的唯一标识符。您关联的组中的用户可以通过使用AWS Transfer Family的启用协议访问您的 Amazon S3 或 Amazon EFS 资源。如果您知道组名，则可以使用 Windows 运行以下命令来查看 SID 值 PowerShell。

```
Get-ADGroup -Filter {samAccountName -like "YourGroupName*"} -Properties
* | Select SamAccountName,ObjectSid
```

在该命令中，替换为您的 YourGroupNameActive Directory 组的名称。

用于验证此参数的正则表达式是由不带空格的大写和小写字母数字字符组成的字符串。此外，还可以包括下划线或以下任何字符：=、.@：/-

类型：字符串

长度限制：最小长度为 1。最大长度为 256。

模式：`S-1-[\d-]+`

必需：是

HomeDirectory

用户使用客户端登录服务器时的登录目录（文件夹）。

`HomeDirectory` 示例为 `/bucket_name/home/mydirectory`。

> **Note**
>
> `HomeDirectory` 参数仅在 `HomeDirectoryType` 设置为 `PATH` 时使用。

类型：字符串

长度限制：长度下限为 0。最大长度为 1024。

模式：`(|/.*)`

必需：否

HomeDirectoryMappings

逻辑目录映射指定哪些 Amazon S3 或 Amazon EFS 路径和密钥应对您的用户可见，以及使其对用户可见的方式。您需要指定`Entry`和`Target`对，其中 `Entry` 显示如何使路径可见，`Target` 是实际的 Amazon S3 或 Amazon EFS 路径。如果您只指定一个目标，则将按原样显示。您还必须确保您的AWS Identity and Access Management（IAM）角色提供对`Target`中路径的访问权限。只有当`HomeDirectoryType`设置为LOGICAL时，才能设置此值。

以下是 Entry 和 Target 对示例。

```
[ { "Entry": "/directory1", "Target": "/bucket_name/home/
mydirectory" } ]
```

在大多数情况下，您可以使用此值而不是会话策略将您的用户锁定到指定的主目录（“chroot”）。为此，您可以将 Entry 设置为 / 并将 Target 设置为 HomeDirectory 参数值。

以下是chroot的 Entry 和 Target 对示例。

```
[ { "Entry": "/", "Target": "/bucket_name/home/mydirectory" } ]
```

类型：HomeDirectoryMapEntry 对象数组

数组成员：最少 1 个物品。最大数量为 50000 个。

必需：否

HomeDirectoryType

您希望用户在登录服务器时，用户主目录的登录目录（文件夹）的类型。如果您将其设置为PATH，则用户将在其文件传输协议客户端中原样看到 Amazon S3 存储桶或 Amazon EFS 的绝对路径。如果您将其设置为LOGICAL，则需要针对您希望如何使 Amazon S3 或 Amazon EFS 路径对用户可见，在HomeDirectoryMappings中提供映射。

> **Note**
>
> 如果HomeDirectoryType是LOGICAL，则必须使用HomeDirectoryMappings参数提供映射。另一方面，如果HomeDirectoryType是PATH，则使用HomeDirectory参数提供绝对路径。您的模板中不能同时使用HomeDirectory和HomeDirectoryMappings。

类型：字符串

有效值：PATH | LOGICAL

必需：否

Policy

适用于您的用户的会话策略，可让您跨多个用户使用相同的 AWS Identity and Access Management(IAM) 角色。此策略将用户的访问权限缩小至 Amazon S3 存储桶的一部分。可在

此策略中使用的变量包括 ${Transfer:UserName}、${Transfer:HomeDirectory} 和 ${Transfer:HomeBucket}。

> **Note**
>
> 仅当ServerId域为 Amazon S3 时，此政策才适用。Amazon EFS 不使用会话策略。
> 对于会话测量，AWS Transfer Family 将策略存储为 JSON blob，而不是策略的 Amazon 资源名称（ARN）。您将策略保存为 JSON blob 并将其传递给 Policy 参数。
> 有关会话策略的示例，请参阅示例会话策略。
> 有关更多信息，请参阅 AWS Security Token ServiceAPI 参考AssumeRole中的。

类型：字符串

长度约束：最小长度为 0。最大长度为 2048。

必需：否

PosixProfile

完整的 POSIX 身份，包括用户 ID (Uid)、组 ID (Gid) 和任何辅助组 ID (SecondaryGids)，用于控制用户对 Amazon EFS 文件系统的访问。POSIX 权限针对文件系统中的文件和目录设置，用于确定用户在将文件传入和传出 Amazon EFS 文件系统时获得的访问权限级别。

类型：PosixProfile 对象

必需：否

Role

控制用户对 Amazon S3 桶或 Amazon EFS 文件系统访问权限的 AWS Identity and Access Management（IAM）角色的 Amazon 资源名称（ARN）。附加到此角色的策略确定在将文件传入和传出 Amazon S3 桶或 Amazon EFS 文件系统时要为用户提供的访问权限级别。IAM 角色还应包含一个信任关系，从而允许服务器在为用户的传输请求提供服务时访问您的资源。

类型：字符串

长度约束：最小长度为 20。最大长度为 2048。

模式：arn:.*role/\S+

必需：是

ServerId

服务器实例的系统分配的唯一标识符。这是将您的用户添加到的特定服务器。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

必需：是

响应语法

```
{
   "ExternalId": "string",
   "ServerId": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

ExternalId

用户所在组的外部标识符可以通过使用AWS Transfer Family的启用协议访问您的 Amazon S3 或 Amazon EFS 资源。

类型：字符串

长度限制：最小长度为 1。最大长度为 256。

模式：S-1-[\d-]+

ServerId

用户附加到的服务器的标识符。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceExistsException

请求的资源不存在，或者存在于为命令指定的区域以外的区域。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK

- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版 SDK
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

CreateAgreement

创建协议。协议是 AWS Transfer Family 服务器和 AS2 流程之间的双边贸易伙伴协议或伙伴关系。该协议定义了服务器与 AS2 进程之间的文件和消息传输关系。为了定义协议，Transfer Family 结合了服务器、本地配置文件、合作伙伴配置文件、证书和其他属性。

合作伙伴以 `PartnerProfileId` 识别，AS2 进程则以 `LocalProfileId` 标识。

请求语法

```
{
   "AccessRole": "string",
   "BaseDirectory": "string",
   "Description": "string",
   "LocalProfileId": "string",
   "PartnerProfileId": "string",
   "ServerId": "string",
   "Status": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

AccessRole

连接器用于使用 AS2 或 SFTP 协议发送文件。对于访问角色，请提供要使用的 AWS Identity and Access Management 角色的 Amazon 资源名称 (ARN)。

对于 AS2 连接器

借助 AS2，您可以通过调用 `StartFileTransfer` 并在请求参数中指定文件路径 `SendFilePaths` 来发送文件。我们使用文件的父目录（例如 `--send-file-paths /bucket/dir/file.txt`，父目录是 `/bucket/dir/`）来临时存储经过处理的 AS2 消息文件，存储

MDN（当从合作伙伴那里收到时），以及写入包含传输相关元数据的最终 JSON 文件。因此，AccessRole 需要提供对 StartFileTransfer 请求中所使用文件位置父目录的读取和写入权限。此外，您还需要提供对您想要使用 StartFileTransfer 发送的文件父目录的读取和写入权限。

如果您对 AS2 连接器执行基本身份验证，则访问角色需要密钥secretsmanager:GetSecretValue权限。如果使用客户管理的密钥而不是 Secrets Manager 中的 AWS 托管密钥对密钥进行加密，则该角色还需要该密钥的kms:Decrypt权限。

对于 SFTP 连接器

因此, 确保提供对 StartFileTransfer 请求中所使用文件位置父目录的读取和写入权限。此外，请确保该角色向提供secretsmanager:GetSecretValue权限 AWS Secrets Manager。

类型：字符串

长度约束：最小长度为 20。最大长度为 2048。

模式：arn:.*role/\S+

必需：是

BaseDirectory

使用 AS2 协议传输的文件的登录目录（文件夹）。

BaseDirectory 示例为 /DOC-EXAMPLE-BUCKET/home/mydirectory。

类型：字符串

长度约束：最小长度为 0。最大长度为 1024。

模式：(|/.*)

必需：是

Description

用于识别协议的名称或简短描述。

类型：字符串

长度限制：最小长度为 1。最大长度为 200。

模式：[\p{Graph}]+

必需：否

LocalProfileId

AS2 本地配置文件的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：p-([0-9a-f]{17})

必需：是

PartnerProfileId

协议中使用的合作伙伴配置文件的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：p-([0-9a-f]{17})

必需：是

ServerId

服务器实例的系统分配的唯一标识符。此标识符表示协议使用的特定服务器。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

必需：是

Status

组件的状态。协议可以是ACTIVE 或 INACTIVE。

类型：字符串

有效值：ACTIVE | INACTIVE

必需：否

Tags

可用于分组和搜索协议的键/值对。

类型：Tag 对象数组

数组成员：最少 1 个物品。最多 50 项。

必需：否

响应语法

```
{
   "AgreementId": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

AgreementId

字段的唯一标识符。使用此 ID 删除或者更新协议，以及用于要求您指定协议 ID 的任何其他 API 调用。

类型：字符串

长度限制：固定长度为 19。

模式：a-([0-9a-f]{17})

错误

有关所有操作返回的常见错误的信息，请参阅 常见错误。

InternalServiceError

当 AWS Transfer Family 服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceExistsException

请求的资源不存在，或者存在于为命令指定的区域以外的区域。

HTTP 状态代码：400

ResourceNotFoundException

当 Transfer Family 服务找不到资源时，就会 AWS引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 Trans AWS fer Family 服务不可用。

HTTP 状态代码：500

ThrottlingException

由于请求限制而导致请求被拒绝。

HTTP 状态代码：400

示例

示例

以下示例创建了协议，并返回协议 ID。

```
aws transfer create-agreement --server-id s-021345abcdef6789 --local-profile-
id p-1234567890abcdef0 --partner-profile-id p-abcdef01234567890 --base-folder /DOC-
EXAMPLE-BUCKET/AS2-files --access-role arn:aws:iam::111122223333:role/AS2-role
```

示例响应

API 调用返回新协议的 ID。

```
{
    "AgreementId": "a-11112222333344444"
}
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- AWS 适用于 .NET 的 SDK
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS JavaScript V3 版软件开发工具包
- AWS 适用于 PHP 的 SDK V3
- AWS Python 软件开发工具包
- AWS 适用于 Ruby V3 的 SDK

CreateConnector

创建连接器，用于捕获 AS2 协议出站连接的参数。发送文件到外部托管的 AS2 服务器时需要使用连接器。对于 SFTP，向 SFTP 服务器发送文件或从 SFTP 服务器接收文件时，需要连接器。有关连接器的更多详细信息，请参阅配置 AS2 连接器和创建 SFTP 连接器。

> **Note**
>
> 您必须只指定一个配置对象：用于 AS2 (`As2Config`) 或 SFTP (`SftpConfig`)。

请求语法

```
{
   "AccessRole": "string",
   "As2Config": {
      "BasicAuthSecretId": "string",
      "Compression": "string",
      "EncryptionAlgorithm": "string",
      "LocalProfileId": "string",
      "MdnResponse": "string",
      "MdnSigningAlgorithm": "string",
      "MessageSubject": "string",
      "PartnerProfileId": "string",
      "SigningAlgorithm": "string"
   },
   "LoggingRole": "string",
   "SecurityPolicyName": "string",
   "SftpConfig": {
      "TrustedHostKeys": [ "string" ],
      "UserSecretId": "string"
   },
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ],
   "Url": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

AccessRole

连接器用于使用 AS2 或 SFTP 协议发送文件。对于访问角色，请提供要使用的 AWS Identity and Access Management 角色的 Amazon 资源名称 (ARN)。

对于 AS2 连接器

借助 AS2，您可以通过调用 `StartFileTransfer` 并在请求参数中指定文件路径 `SendFilePaths` 来发送文件。我们使用文件的父目录（例如 `--send-file-paths /bucket/dir/file.txt`，父目录是 `/bucket/dir/`）来临时存储经过处理的 AS2 消息文件，存储 MDN（当从合作伙伴那里收到时），以及写入包含传输相关元数据的最终 JSON 文件。因此，`AccessRole` 需要提供对 `StartFileTransfer` 请求中所使用文件位置父目录的读取和写入权限。此外，您还需要提供对您想要使用 `StartFileTransfer` 发送的文件父目录的读取和写入权限。

如果您对 AS2 连接器执行基本身份验证，则访问角色需要密钥`secretsmanager:GetSecretValue`权限。如果使用客户管理的密钥而不是 Secrets Manager 中的 AWS 托管密钥对密钥进行加密，则该角色还需要该密钥的`kms:Decrypt`权限。

对于 SFTP 连接器

因此, 确保提供对 `StartFileTransfer` 请求中所使用文件位置父目录的读取和写入权限。此外，请确保该角色向提供`secretsmanager:GetSecretValue`权限 AWS Secrets Manager。

类型：字符串

长度约束：最小长度为 20。最大长度为 2048。

模式：`arn:.*role/\S+`

必需：是

As2Config

包含连接器对象参数的结构。

类型：As2ConnectorConfig 对象

必需：否

LoggingRole

(IAM) 角色的亚马逊资源名称 AWS Identity and Access Management (ARN)，它允许连接器开启对 Amazon S3 CloudWatch 事件的日志记录。设置后，您可以在 CloudWatch 日志中查看连接器活动。

类型：字符串

长度约束：最小长度为 20。最大长度为 2048。

模式：`arn:.*role/\S+`

必需：否

SecurityPolicyName

为连接器指定安全策略的名称。

类型：字符串

长度限制：长度下限为 0。最大长度为 100。

模式：`TransferSFTPConnectorSecurityPolicy-[A-Za-z0-9-]+`

必需：否

SftpConfig

包含 SFTP 连接器对象参数的结构。

类型：SftpConnectorConfig 对象

必需：否

Tags

可用于分组和搜索连接器的键/值对。标签是出于任何目的附加到用户的元数据。

类型：Tag 对象数组

数组成员：最少 1 个物品。最多 50 项。

必需：否

Url

合作伙伴的 AS2 或 SFTP 端点的 URL。

类型：字符串

长度约束：最小长度为 0。最大长度为 255。

必需：是

响应语法

```
{
   "ConnectorId": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

ConnectorId

AS2 配置文件的唯一标识符，将在 API 调用成功后返回。

类型：字符串

长度限制：固定长度为 19。

模式：`c-([0-9a-f]{17})`

错误

有关所有操作的常见错误信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family 服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceExistsException

请求的资源不存在，或者存在于为命令指定的区域以外的区域。

HTTP 状态代码：400

ResourceNotFoundException

当 Transfer Family 服务找不到资源时，就会 AWS引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 Trans AWS fer Family 服务不可用。

HTTP 状态代码：500

ThrottlingException

由于请求限制而导致请求被拒绝。

HTTP 状态代码：400

示例

示例

以下示例创建 AS2 连接器。在以下命令中，替换以下项目：

- `url`：提供贸易伙伴的 AS2 服务器 URL。
- `your-IAM-role-for-bucket-access`：IAM 角色，具有您将用于存储文件的 Amazon S3 存储桶访问权限。
- 使用 ARN 作为您的日志角色，其中包括您 AWS 账户 的 ID。
- 提供包含 AS2 连接器配置参数的文件路径。AS2 连接器配置对象在 A s ConnectorConfig 2 中进行了描述。

```
// Listing for testAs2Config.json
{
  "LocalProfileId": "your-profile-id",
  "PartnerProfileId": "partner-profile-id",
  "MdnResponse": "SYNC",
  "Compression": "ZLIB",
  "EncryptionAlgorithm": "AES256_CBC",
  "SigningAlgorithm": "SHA256",
  "MdnSigningAlgorithm": "DEFAULT",
  "MessageSubject": "Your Message Subject"
}
```

```
aws transfer create-connector --url "http://partner-as2-server-url" \
          --access-role your-IAM-role-for-bucket-access \
          --logging-role arn:aws:iam::your-account-id:role/service-role/
AWSTransferLoggingAccess \
          --as2-config file://path/to/testAS2Config.json
```

示例

以下示例创建 AS2 连接器。在以下命令中，替换以下项目：

- `sftp-server-url`：提供您正在与之交换文件的 SFTP 服务器 URL。
- `your-IAM-role-for-bucket-access`：IAM 角色，具有您将用于存储文件的 Amazon S3 存储桶访问权限。
- 使用 ARN 作为您的日志角色，其中包括您 AWS 账户 的 ID。
- 提供包含 AS2 连接器配置参数的文件路径。中描述了 SFTP 连接器配置对象。SftpConnectorConfig

```
// Listing for testSFTPConfig.json
{
   "UserSecretId": "arn:aws:secretsmanager:us-east-2:123456789012:secret:aws/transfer/
example-username-key",
   "TrustedHostKeys": [
      "sftp.example.com ssh-rsa AAAAbbbb...EEEE="
   ]
}
```

```
aws transfer create-connector --url "sftp://sftp-server-url" \
--access-role your-IAM-role-for-bucket-access \
--logging-role arn:aws:iam::your-account-id:role/service-role/AWSTransferLoggingAccess
 \
--sftp-config file://path/to/testSFTPConfig.json
```

示例

API 调用返回新连接器的 ID。

示例响应

```
{
    "ConnectorId": "a-11112222333344444"
}
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- AWS 适用于 .NET 的 SDK
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS JavaScript V3 版软件开发工具包
- AWS 适用于 PHP 的 SDK V3
- AWS Python 软件开发工具包
- AWS 适用于 Ruby V3 的 SDK

CreateProfile

创建用于 AS2 传输的本地或合作伙伴配置文件。

请求语法

```
{
   "As2Id": "string",
   "CertificateIds": [ "string" ],
   "ProfileType": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

As2Id

As2Id 是 AS2-name，如 RFC 4130 中所定义。对于入站传输，这是合作伙伴发送的 AS2 消息的 AS2-From 标头。对于出站连接器，这是使用 AS2-To API 操作发送给合作伙伴的 AS2 消息的 StartFileTransfer 标头。此 ID 不能包含空格。

类型：字符串

长度限制：长度下限为 1。长度上限为 128。

模式：[\p{Print}\s]*

必需：是

CertificateIds

已导入证书的标识符数组。您可以使用此标识符来处理配置文件和合作伙伴配置文件。

类型：字符串数组

长度限制：固定长度为 22。

模式：`cert-([0-9a-f]{17})`

必需：否

ProfileType

确定要创建的配置文件类型：

- 指定 LOCAL 以创建本地配置文件。本地配置文件表示支持 AS2 的 Transfer Family 服务器组织或群体。
- 指定 PARTNER 以创建本地配置文件。合作伙伴配置文件表示 Transfer Family 外部的远程组织。

类型：字符串

有效值：`LOCAL | PARTNER`

必需：是

Tags

可用于分组和搜索配置文件的键/值对。

类型：Tag 对象数组

数组成员：最少 1 个物品。最多 50 项。

必需：否

响应语法

```
{
   "ProfileId": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

ProfileId

AS2 配置文件的唯一标识符，将在 API 调用成功后返回。

类型：字符串

长度限制：固定长度为 19。

模式：`p-([0-9a-f]{17})`

错误

有关所有操作返回的常见错误的信息，请参阅 常见错误。

InternalServiceError

当 AWS Transfer Family 服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 Transfer Family 服务找不到资源时，就会 AWS引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 Trans AWS fer Family 服务不可用。

HTTP 状态代码：500

ThrottlingException

由于请求限制而导致请求被拒绝。

HTTP 状态代码：400

示例

示例

以下示例创建配置文件，并返回配置文件 ID。

证书 ID 是在您运行import-certificate时创建的，一个用于签名证书，一个用于加密证书。

```
aws transfer create-profile --as2-id MYCORP --certificate-ids c-abcdefg123456hijk
               c-987654aaaa321bbbb
```

示例响应

API 调用返回新配置文件的 ID。

```
{
    "ProfileId": "p-11112222333344444"
}
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- AWS 适用于 .NET 的 SDK
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS JavaScript V3 版软件开发工具包
- AWS 适用于 PHP 的 SDK V3
- AWS Python 软件开发工具包
- AWS 适用于 Ruby V3 的 SDK

CreateServer

在 AWS中基于所选文件传输协议实例化自动伸缩的虚拟服务器。当您更新启用文件传输协议的服务器或与用户一起工作时，使用分配给新创建的服务器的服务生成的 ServerId 属性。

请求语法

```
{
   "Certificate": "string",
   "Domain": "string",
   "EndpointDetails": {
      "AddressAllocationIds": [ "string" ],
      "SecurityGroupIds": [ "string" ],
      "SubnetIds": [ "string" ],
      "VpcEndpointId": "string",
      "VpcId": "string"
   },
   "EndpointType": "string",
   "HostKey": "string",
   "IdentityProviderDetails": {
      "DirectoryId": "string",
      "Function": "string",
      "InvocationRole": "string",
      "SftpAuthenticationMethods": "string",
      "Url": "string"
   },
   "IdentityProviderType": "string",
   "LoggingRole": "string",
   "PostAuthenticationLoginBanner": "string",
   "PreAuthenticationLoginBanner": "string",
   "ProtocolDetails": {
      "As2Transports": [ "string" ],
      "PassiveIp": "string",
      "SetStatOption": "string",
      "TlsSessionResumptionMode": "string"
   },
   "Protocols": [ "string" ],
   "S3StorageOptions": {
      "DirectoryListingOptimization": "string"
   },
   "SecurityPolicyName": "string",
   "StructuredLogDestinations": [ "string" ],
   "Tags": [
```

```
      {
         "Key": "string",
         "Value": "string"
      }
   ],
   "WorkflowDetails": {
      "OnPartialUpload": [
         {
            "ExecutionRole": "string",
            "WorkflowId": "string"
         }
      ],
      "OnUpload": [
         {
            "ExecutionRole": "string",
            "WorkflowId": "string"
         }
      ]
   }
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

Certificate

AWS Certificate Manager （ ACM ）证书的 Amazon 资源名称（ ARN ）。在 `Protocols` 设置为 FTPS 时是必需的。

要申请新的公共证书，请参阅《 AWS Certificate Manager 用户指南》中的申请公共证书。

要将现有证书导入 ACM，请参阅 AWS Certificate Manager 用户指南中的将证书导入 ACM。

要请求私有证书通过私有 IP 地址使用 FTPS，请参阅《 AWS Certificate Manager 用户指南》中的申请私有证书。

支持具有以下加密算法和密钥大小的证书：

- 2048 位 RSA (RSA_2048)
- 4096 位 RSA (RSA_4096)

- Elliptic Prime Curve 256 位 (EC_prime256v1)
- Elliptic Prime Curve 384 位 (EC_secp384r1)
- Elliptic Prime Curve 521 位 (EC_secp521r1)

> **Note**
>
> 证书必须是指定了 FQDN 或 IP 地址且具有有关颁发者的信息的有效 SSL/TLS X.509 版本 3 证书。

类型：字符串

长度约束：最小长度为 0。长度上限为 1600。

必需：否

Domain

指定用于文件传输的存储系统的域。有两个域可用：Amazon Simple Storage Service 和 Amazon Elastic File System System (Amazon EFS)。默认值为 3。

> **Note**
>
> 在创建用户后，将无法更改用户名。

类型：字符串

有效值：`S3 | EFS`

必需：否

EndpointDetails

要为服务器配置的 Virtual Private Cloud (VPC) 端点设置。当您在 VPC 中托管端点时，您可以使端点仅可供 VPC 内的资源访问，也可以附加弹性 IP 地址并使端点可由客户端通过 Internet 访问。您 VPC 的默认安全组会自动分配到您的端点。

类型：EndpointDetails 对象

必需：否

EndpointType

您希望服务器使用的端点类型。您可以选择使服务器的端点可公开访问（PUBLIC），或将其托管在 VPC 内。如果端点在 VPC 中托管，您可以仅在 VPC 内限制对服务器和资源的访问，或者通过将弹性 IP 地址直接附加到其上，使其面向 Internet。

> **Note**
>
> 2021 年 5 月 19 日之后， AWS 账户 如果您的账户EndpointType=VPC_ENDPOINT在 2021 年 5 月 19 日之前尚未使用自己的服务器创建服务器，则您将无法使用创建服务器。如果您已EndpointType=VPC_ENDPOINT在 2021 年 5 月 19 AWS 账户 日当天或之前创建过服务器，则不会受到影响。在此日期之后，使用 EndpointType = VPC。
> 有关更多信息，请参阅 停止使用 VPC_ENDPOINT。
> 建议您使用 VPC 作为 EndpointType。使用此终端节点类型，您可以选择将最多三个弹性 IPv4 地址（包括 BYO IP）直接与服务器的终端节点相关联，并使用 VPC 安全组限制客户端的公有 IP 地址的流量。如果 EndpointType 设置为 VPC_ENDPOINT，则无法实现此操作。

类型：字符串

有效值：PUBLIC | VPC | VPC_ENDPOINT

必需：否

HostKey

用于启用 SFTP 的服务器的 RSA、ECDSA 或 ED25519 私钥。如果要轮换密钥，可以添加多个主机密钥，也可以添加一组使用不同算法的活动密钥。

使用以下命令生成不带密码的 RSA 2048 位密钥：

```
ssh-keygen -t rsa -b 2048 -N "" -m PEM -f my-new-server-key.
```

为 -b 选项使用最小值 2048。您可以使用 3072 或 4096 创建更强的密钥。

使用以下命令生成不带密码的 ECDSA 256 位密钥：

```
ssh-keygen -t ecdsa -b 256 -N "" -m PEM -f my-new-server-key.
```

ECDSA 的 -b 选项的有效值为 256、384 和 521。

使用以下命令生成不带密码的 ED25519 密钥：

```
ssh-keygen -t ed25519 -N "" -f my-new-server-key.
```

对于所有这些命令，你可以my-new-server-key用你选择的字符串替换。

> **Important**
>
> 如果您不打算将现有用户从启用 SFTP 的现有服务器迁移到新服务器，不要更新主机密钥。意外更改服务器的主机密钥会导致中断。

有关更多信息，请参阅《用户指南》中的更新启用 SFTP 的服务器的主机密钥。 AWS Transfer Family

类型：字符串

长度约束：最小长度为 0。最大长度为 4096。

必需：否

IdentityProviderDetails

在 IdentityProviderType 设置为 AWS_DIRECTORY_SERVICE 或 时是必需的。接受包含使用 AWS_DIRECTORY_SERVICE 中的目录或调用客户提供的身份验证 API（包括 API 网关 URL）所需的所有信息的数组。在 IdentityProviderType 设置为 SERVICE_MANAGED 时不是必需的。

类型：IdentityProviderDetails 对象

必需：否

IdentityProviderType

服务器的身份验证模式。默认值为SERVICE_MANAGED，允许您在 AWS Transfer Family 服务中存储和访问用户凭证。

用于提供AWS_DIRECTORY_SERVICE对本地环境中的活动目录组 AWS Directory Service for Microsoft Active Directory 或 Microsoft Active Directory 组的访问权限，或者 AWS 使用 AD Connector 提供访问权限。此选项还要求您使用 IdentityProviderDetails 参数提供 Directory ID。

使用 API_GATEWAY 值以与您选择的身份提供者集成。API_GATEWAY 设置要求您提供 Amazon API Gateway 端点 URL 以使用 IdentityProviderDetails 参数调用身份验证。

使用该AWS_LAMBDA值可直接使用 AWS Lambda 函数作为您的身份提供商。如果选择该值，则必须在 Function 数据类型的 IdentityProviderDetails 参数中指定 Lambda 函数的 ARN。

类型：字符串

有效值：SERVICE_MANAGED | API_GATEWAY | AWS_DIRECTORY_SERVICE | AWS_LAMBDA

必需：否

LoggingRole

(IAM) 角色的亚马逊资源名称 (ARN)，它允许服务器为亚马逊 S3 或 Amazon efSevents 开启亚马逊 CloudWatch 日志记录。 AWS Identity and Access Management 设置后，您可以在 CloudWatch 日志中查看用户活动。

类型：字符串

长度约束：最小长度为 0。最大长度为 2048。

模式：(|arn:.*role/\S+)

必需：否

PostAuthenticationLoginBanner

指定用户连接到服务器时要显示的字符串。此字符串在用户进行身份验证后显示。

> **Note**
>
> SFTP 协议不支持身份验证后显示横幅。

类型：字符串

长度约束：最小长度为 0。最大长度为 4096。

模式：[\x09-\x0D\x20-\x7E]*

必需：否

PreAuthenticationLoginBanner

指定用户连接到服务器时要显示的字符串。此字符串在用户进行身份验证前显示。例如，以下横幅显示有关使用系统的详细信息：

```
This system is for the use of authorized users only. Individuals using
this computer system without authority, or in excess of their authority,
are subject to having all of their activities on this system monitored
and recorded by system personnel.
```

类型：字符串

长度约束：最小长度为 0。最大长度为 4096。

模式：`[\x09-\x0D\x20-\x7E]*`

必需：否

ProtocolDetails

为服务器配置的协议设置。

- 要指示被动模式（适用于 FTP 和 FTPS 协议），请使用 `PassiveIp` 参数。输入一个点分四组的 IPv4 地址，例如防火墙、路由器或负载均衡器的外部 IP 地址。
- 要忽略当客户端尝试对您上传到 S3 桶的文件使用 `SETSTAT` 命令时生成的错误，请使用 `SetStatOption` 参数。要让 AWS Transfer Family 服务器忽略`SETSTAT`命令并上传文件而不必对 SFTP 客户端进行任何更改，请将该值设置为。`ENABLE_NO_OP`如果您将`SetStatOption`参数设置为`ENABLE_NO_OP`，Transfer Family 会生成一个到 Amazon Logs 的 CloudWatch 日志条目，以便您可以确定客户何时`SETSTAT`拨打电话。
- 要确定您的 AWS Transfer Family 服务器是否通过唯一的会话 ID 恢复最近协商的会话，请使用`TlsSessionResumptionMode`参数。
- `As2Transports` 指示 AS2 消息的传输方法。目前仅支持 HTTP。

类型：ProtocolDetails 对象

必需：否

Protocols

指定文件传输协议客户端可以用来连接到服务器端点的一个或多个文件传输协议。可用的协议包括：

- SFTP（Secure Shell (SSH) 文件传输协议）：通过 SSH 的文件传输
- FTPS（文件传输协议安全）：使用 TLS 加密的文件传输
- FTP（文件传输协议）：未加密的文件传输
- AS2（适用性声明 2）：用于传输结构化 business-to-business 数据

> **Note**
>
> - 如果选择FTPS，则必须选择存储在 AWS Certificate Manager (ACM) 中的证书，当客户端通过 FTPS 连接到服务器时，该证书用于识别您的服务器。
> - 如果 Protocol 包括 FTP 或 FTPS，则 EndpointType 必须为 VPC，且 IdentityProviderType 必须为 AWS_DIRECTORY_SERVICE、AWS_LAMBDA 或 API_GATEWAY。
> - 如果 Protocol 包含 FTP，则无法关联 AddressAllocationIds。
> - 如果仅将 Protocol 设置为 SFTP，则可以将 EndpointType 设置为 PUBLIC，并且可以将 IdentityProviderType 设置为任何支持的身份类型：SERVICE_MANAGED、AWS_DIRECTORY_SERVICE、AWS_LAMBDA 或 API_GATEWAY。
> - 如果 Protocol 包括 AS2，则 EndpointType 必须是 VPC，并且域必须是 Amazon S3。

类型：字符串数组

数组成员：最少 1 个物品。最多 4 项。

有效值：SFTP | FTP | FTPS | AS2

必需：否

S3StorageOptions

指定是否对您的 Amazon S3 目录的性能进行了优化。默认情况下，将禁用该功能。

默认情况下，主目录映射TYPE的值为。DIRECTORY如果启用此选项，则需要将显式设置为，HomeDirectoryMapEntryType以FILE使映射具有文件目标。

类型：S3StorageOptions 对象

必需：否

SecurityPolicyName

指定服务器安全策略的名称。

类型：字符串

长度限制：长度下限为 0。最大长度为 100。

模式：Transfer[A-Za-z0-9]*SecurityPolicy-[A-Za-z0-9-]+

必需：否

StructuredLogDestinations

指定将服务器日志发送到日志组。

要指定日志组，必须提供现有日志组的 ARN。在这种情况下，日志组的格式如下所示：

arn:aws:logs:region-name:amazon-account-id:log-group:log-group-name:*

例如，arn:aws:logs:us-east-1:111122223333:log-group:mytestgroup:*

如果您之前为服务器指定了日志组，则可以通过在update-server调用中为该参数提供空值来清除该日志组，从而关闭结构化日志记录。例如：

```
update-server --server-id s-1234567890abcdef0 --structured-log-destinations
```

类型：字符串数组

数组成员：最少 0 项。最多 1 项。

长度约束：最小长度为 20。长度上限为 1600。

模式：arn:\S+

必需：否

Tags

可用于分组和搜索服务器的键/值对。

类型：Tag 对象数组

数组成员：最少 1 个物品。最多 50 项。

必需：否

WorkflowDetails

指定要分配的工作流的工作流 ID 以及用于执行工作流的执行角色。

除了要在文件完全上传时执行的工作流，WorkflowDetails 还可能包含在部分文件上传时执行的工作流的工作流 ID（和执行角色）。在文件仍在上传时，如果断开连接，则会发生部分上传。

类型：WorkflowDetails 对象

必需：否

响应语法

```
{
   "ServerId": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

ServerId

创建的服务器的服务分配的 ID。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

AccessDeniedException

您没有足够的访问权限，无法执行该操作。

HTTP 状态代码：400

InternalServiceError

当 AWS Transfer Family 服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceExistsException

请求的资源不存在，或者存在于为命令指定的区域以外的区域。

HTTP 状态代码：400

ResourceNotFoundException

当 Transfer Family 服务找不到资源时，就会 AWS引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 Trans AWS fer Family 服务不可用。

HTTP 状态代码：500

ThrottlingException

由于请求限制而导致请求被拒绝。

HTTP 状态代码：400

示例

示例

以下 示例使用 创建一个新的 表。

示例请求

```
{
   "EndpointType": "VPC",
   "EndpointDetails":...,
   "HostKey": "Your RSA private key",
   "IdentityProviderDetails": "IdentityProvider",
   "IdentityProviderType": "SERVICE_MANAGED",
```

```
   "LoggingRole": "CloudWatchLoggingRole",
   "Tags": [
      {
         "Key": "Name",
         "Value": "MyServer"
      }
   ]
}
```

示例

这是此 API 调用的示例响应。

示例响应

```
{
   "ServerId": "s-01234567890abcdef"
}
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- AWS 适用于 .NET 的 SDK
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS JavaScript V3 版软件开发工具包
- AWS 适用于 PHP 的 SDK V3
- AWS Python 软件开发工具包
- AWS 适用于 Ruby V3 的 SDK

CreateUser

创建用户并将其与现有的安全文件传输协议 (SFTP) 服务器关联。您可以仅创建用户并将其与 IdentityProviderType 设置为 SERVICE_MANAGED 的服务器关联。使用的参数CreateUser，您可以指定用户名、设置主目录、存储用户的公钥以及分配用户的 AWS Identity and Access Management (IAM) 角色。您还可以选择添加会话策略，并向可用于分组和搜索用户的标签分配元数据。

请求语法

```
{
   "HomeDirectory": "string",
   "HomeDirectoryMappings": [
      {
         "Entry": "string",
         "Target": "string",
         "Type": "string"
      }
   ],
   "HomeDirectoryType": "string",
   "Policy": "string",
   "PosixProfile": {
      "Gid": number,
      "SecondaryGids": [ number ],
      "Uid": number
   },
   "Role": "string",
   "ServerId": "string",
   "SshPublicKeyBody": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ],
   "UserName": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

HomeDirectory

用户使用客户端登录服务器时的登录目录（文件夹）。

HomeDirectory 示例为 /bucket_name/home/mydirectory。

> **Note**
>
> HomeDirectory 参数仅在 HomeDirectoryType 设置为 PATH 时使用。

类型：字符串

长度约束：最小长度为 0。最大长度为 1024。

模式：(|/.*)

必需：否

HomeDirectoryMappings

逻辑目录映射指定哪些 Amazon S3 或 Amazon EFS 路径和密钥应对您的用户可见，以及使其对用户可见的方式。您需要指定Entry和Target对，其中 Entry 显示如何使路径可见，Target 是实际的 Amazon S3 或 Amazon EFS 路径。如果您只指定一个目标，则将按原样显示。您还必须确保您的 AWS Identity and Access Management (IAM) 角色提供对中路径的访问权限Target。只有当HomeDirectoryType设置为LOGICAL时，才能设置此值。

以下是负载的示例。

```
[ { "Entry": "/directory1", "Target": "/bucket_name/home/
mydirectory" } ]
```

在大多数情况下，您可以使用此值而不是会话策略将您的用户锁定到指定的主目录（“chroot”）。为此，您可以将 Entry 设置为 /，并将 Target 值设置为用户在登录时看到的主目录值。

以下是 的示例负载。

```
[ { "Entry": "/", "Target": "/bucket_name/home/mydirectory" } ]
```

类型：HomeDirectoryMapEntry 对象数组

数组成员：最少 1 个物品。最大数量为 50000 件物品。

必需：否

HomeDirectoryType

您希望用户在登录服务器时，用户主目录的登录目录（文件夹）的类型。如果您将其设置为PATH，则用户将在其文件传输协议客户端中原样看到 Amazon S3 存储桶或 Amazon EFS 的绝对路径。如果您将其设置为LOGICAL，则需要针对您希望如何使 Amazon S3 或 Amazon EFS 路径对用户可见，在HomeDirectoryMappings中提供映射。

> **Note**
>
> 如果HomeDirectoryType是LOGICAL，则必须使用HomeDirectoryMappings参数提供映射。另一方面，如果HomeDirectoryType是PATH，则使用HomeDirectory参数提供绝对路径。您的模板中不能同时使用HomeDirectory和HomeDirectoryMappings。

类型：字符串

有效值：`PATH | LOGICAL`

必需：否

Policy

适用于您的用户的会话策略，以便您可以跨多个用户使用同一 AWS Identity and Access Management (IAM) 角色。此策略将用户的访问权限缩小至 Amazon S3 存储桶的一部分。可在此策略中使用的变量包括 `${Transfer:UserName}`、`${Transfer:HomeDirectory}` 和 `${Transfer:HomeBucket}`。

> **Note**
>
> 仅当ServerId域为 Amazon S3 时，此政策才适用。Amazon EFS 不使用会话策略。
> 对于会话策略，将策略 AWS Transfer Family 存储为 JSON blob，而不是策略的 Amazon 资源名称 (ARN)。您将策略保存为 JSON blob 并将其传递给 Policy 参数。
> 有关会话策略的示例，请参阅示例会话策略。
> 有关更多信息，请参阅AssumeRole《 AWS 安全令牌服务 API 参考》。

类型：字符串

长度约束：最小长度为 0。最大长度为 2048。

必需：否

PosixProfile

完整的 POSIX 身份，包括用户 ID (`Uid`)、组 ID (`Gid`) 和任何辅助组 ID (`SecondaryGids`)，用于控制用户对 Amazon EFS 文件系统的访问。POSIX 权限针对文件系统中的文件和目录设置，用于确定用户在将文件传入和传出 Amazon EFS 文件系统时获得的访问权限级别。

类型：PosixProfile 对象

必需：否

Role

(IAM) 角色的亚马逊资源名称 (ARN)，用于控制您的用户对您的 Amazon S3 存储桶或 Amazon EFS 文件系统的访问权限。 AWS Identity and Access Management 附加到此角色的策略确定在将文件传入和传出 Amazon S3 桶或 Amazon EFS 文件系统时要为用户提供的访问权限级别。IAM 角色还应包含一个信任关系，从而允许服务器在为用户的传输请求提供服务时访问您的资源。

类型：字符串

长度约束：最小长度为 20。最大长度为 2048。

模式：`arn:.*role/\S+`

必需：是

ServerId

服务器实例的系统分配的唯一标识符。这是将您的用户添加到的特定服务器。

类型：字符串

长度限制：固定长度为 19。

模式：`s-([0-9a-f]{17})`

必需：是

SshPublicKeyBody

Secure Shell (SSH) 密钥的公共部分，用于为服务器验证用户身份。

三个标准的 SSH 公钥格式元素是<key type>、<body base64>和可选的<comment>，每个元素之间都有空格。

AWS Transfer Family 接受 RSA、ECDSA 和 ED25519 密钥。

- 对于 RSA 密钥，密钥类型为 ssh-rsa。
- 对于 ED25519 密钥，密钥类型为 ssh-ed25519。
- 对于 ECDSA 密钥，ecdsa-sha2-nistp256 字符串为ecdsa-sha2-nistp384、或，具体取决于您生成的密钥的大小。

类型：字符串

长度约束：最小长度为 0。最大长度为 2048。

必需：否

Tags

可用于分组和搜索用户的键/值对。标签是出于任何目的附加到用户的元数据。

类型：Tag 对象数组

数组成员：最少 1 个物品。最多 50 项。

必需：否

UserName

一个标识用户并与 ServerId 关联的唯一字符串。此用户名长度最少必须为 3 个字符，最多为 100 个字符。以下是有效的字符：a-z、A-Z、0-9、下划线“_”、连字符“-”、句点“.”和“@”符号。用户名不能以连字符、句点或 @ 符号开头。

类型：字符串

长度约束：最小长度为 3。最大长度为 100。

模式：[\w][\w@.-]{2,99}

必需：是

响应语法

```
{
   "ServerId": "string",
```

```
   "UserName": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

ServerId

用户连接到的服务器的 ID。

类型：字符串

长度限制：固定长度为 19。

模式：`s-([0-9a-f]{17})`

UserName

标识 TranserIn Family 用户的唯一字符串。

类型：字符串

长度约束：最小长度为 3。最大长度为 100。

模式：`[\w][\w@.-]{2,99}`

错误

有关所有操作返回的常见错误的信息，请参阅 常见错误。

InternalServiceError

当 AWS Transfer Family 服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceExistsException

请求的资源不存在，或者存在于为命令指定的区域以外的区域。

HTTP 状态代码：400

ResourceNotFoundException

当 Transfer Family 服务找不到资源时，就会 AWS引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 Trans AWS fer Family 服务不可用。

HTTP 状态代码：500

示例

示例

例如，要创建用户，您可以先将参数保存至`createUserParameters`等 JSON 文件，然后运行 create-user API 命令。

```
{
    "HomeDirectory": "/DOC-EXAMPLE-BUCKET",
    "HomeDirectoryType": "PATH",
    "Role": "arn:aws:iam::111122223333:role/bob-role",
    "ServerId": "s-1111aaaa2222bbbb3",
    "SshPublicKeyBody": "ecdsa-sha2-nistp521 AAAAE2VjZHNhLXNoYTItbmlzdHA...
 bobusa@mycomputer.us-east-1.amazon.com",
    "UserName": "bobusa-API"
}
```

示例请求

```
aws transfer create-user --cli-input-json file://createUserParameters
```

示例响应

```
{
```

```
    "ServerId": ""s-1111aaaa2222bbbb3",
    "UserName": "bobusa-API"
}
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- AWS 适用于 .NET 的 SDK
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS JavaScript V3 版软件开发工具包
- AWS 适用于 PHP 的 SDK V3
- AWS Python 软件开发工具包
- AWS 适用于 Ruby V3 的 SDK

CreateWorkflow

允许您使用文件传输完成后工作流调用的指定步骤和步骤详细信息创建工作流。创建工作流后，您可以通过在 CreateServer 和 UpdateServer 操作中指定 workflow-details 字段将创建的工作流与任何传输服务器相关联。

请求语法

```
{
   "Description": "string",
   "OnExceptionSteps": [
      {
         "CopyStepDetails": {
            "DestinationFileLocation": {
               "EfsFileLocation": {
                  "FileSystemId": "string",
                  "Path": "string"
               },
               "S3FileLocation": {
                  "Bucket": "string",
                  "Key": "string"
               }
            },
            "Name": "string",
            "OverwriteExisting": "string",
            "SourceFileLocation": "string"
         },
         "CustomStepDetails": {
            "Name": "string",
            "SourceFileLocation": "string",
            "Target": "string",
            "TimeoutSeconds": number
         },
         "DecryptStepDetails": {
            "DestinationFileLocation": {
               "EfsFileLocation": {
                  "FileSystemId": "string",
                  "Path": "string"
               },
               "S3FileLocation": {
                  "Bucket": "string",
                  "Key": "string"
               }
```

```
            },
            "Name": "string",
            "OverwriteExisting": "string",
            "SourceFileLocation": "string",
            "Type": "string"
         },
         "DeleteStepDetails": {
            "Name": "string",
            "SourceFileLocation": "string"
         },
         "TagStepDetails": {
            "Name": "string",
            "SourceFileLocation": "string",
            "Tags": [
               {
                  "Key": "string",
                  "Value": "string"
               }
            ]
         },
         "Type": "string"
      }
   ],
   "Steps": [
      {
         "CopyStepDetails": {
            "DestinationFileLocation": {
               "EfsFileLocation": {
                  "FileSystemId": "string",
                  "Path": "string"
               },
               "S3FileLocation": {
                  "Bucket": "string",
                  "Key": "string"
               }
            },
            "Name": "string",
            "OverwriteExisting": "string",
            "SourceFileLocation": "string"
         },
         "CustomStepDetails": {
            "Name": "string",
            "SourceFileLocation": "string",
            "Target": "string",
```

```
            "TimeoutSeconds": number
         },
         "DecryptStepDetails": {
            "DestinationFileLocation": {
               "EfsFileLocation": {
                  "FileSystemId": "string",
                  "Path": "string"
               },
               "S3FileLocation": {
                  "Bucket": "string",
                  "Key": "string"
               }
            },
            "Name": "string",
            "OverwriteExisting": "string",
            "SourceFileLocation": "string",
            "Type": "string"
         },
         "DeleteStepDetails": {
            "Name": "string",
            "SourceFileLocation": "string"
         },
         "TagStepDetails": {
            "Name": "string",
            "SourceFileLocation": "string",
            "Tags": [
               {
                  "Key": "string",
                  "Value": "string"
               }
            ]
         },
         "Type": "string"
      }
   ],
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

Description

指定工作流的文本描述。

类型：字符串

长度约束：最小长度为 0。最大长度为 256。

模式：`[\w- ]*`

必需：否

OnExceptionSteps

指定在工作流执行期间遇到错误时要采取的步骤（措施）。

> **Note**
>
> 对于自定义步骤，Lambda 函数需要发送`FAILURE`，回调 API 以启动异常步骤。此外，如果 Lambda 在超时之前没有发送`SUCCESS`，则会执行异常步骤。

类型：WorkflowStep 对象数组

数组成员：最少 0 个物品。最多 8 项。

必需：否

Steps

指定所指定工作流中步骤的详细信息。

TYPE 指定要对此步骤采取以下哪些操作。

- **`COPY`** – 将文件复制到另一个位置。
- **`CUSTOM`**-使用 AWS Lambda 函数目标执行自定义步骤。
- **`DECRYPT`** – 解密上传前加密的文件。

- **DELETE** – 删除文件。
- **TAG** – 向文件添加标签。

> **Note**
>
> 目前，仅 S3 支持复制和标记。

对于文件位置，您可指定 Amazon S3 存储桶和密钥，或者指定 Amazon EFS 文件系统 ID 和路径。

类型：WorkflowStep 对象数组

数组成员：最少 0 个物品。最多 5 项。

必需：是

Tags

可用于分组和搜索工作流的键值对。标签是出于任何目的附加到工作流的元数据。

类型：Tag 对象数组

数组成员：最少 1 个物品。最多 50 项。

必需：否

响应语法

```
{
   "WorkflowId": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

WorkflowId

工作流的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`w-([a-z0-9]{17})`

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

AccessDeniedException

您没有足够的访问权限，无法执行该操作。

HTTP 状态代码：400

InternalServiceError

当 AWS Transfer Family 服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceExistsException

请求的资源不存在，或者存在于为命令指定的区域以外的区域。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 Trans AWS fer Family 服务不可用。

HTTP 状态代码：500

ThrottlingException

由于请求限制而导致请求被拒绝。

HTTP 状态代码：400

示例

示例

您可以将工作流步骤信息保存到文本文件中，然后使用该文件创建工作流，如下例所示。以下示例假设您已将工作流步骤保存到 *example-file.json* （在运行该命令的同一文件夹中），并且您希望在弗吉尼亚北部（us-east-1）区域中创建工作流。

```
aws transfer create-workflow --description "example workflow from a file" --steps
 file://example-file.json --region us-east-1
```

```
// Example file containing workflow steps
[
   {
     "Type": "TAG",
     "TagStepDetails": {
        "Name": "TagStep",
        "Tags": [
          {
            "Key": "name",
            "Value": "testTag"
          }
        ]
     }
   },
   {
     "Type": "COPY",
     "CopyStepDetails": {
        "Name": "CopyStep",
        "DestinationFileLocation": {
        "S3FileLocation": {
           "Bucket": "DOC-EXAMPLE-BUCKET",
           "Key": "DOC-EXAMPLE-KEY/"
                  }
        },
        "OverwriteExisting": "TRUE",
        "SourceFileLocation": "${original.file}"
     }
   },
   {
     "Type": "DELETE",
     "DeleteStepDetails":{
```

```
      "Name":"DeleteStep",
      "SourceFileLocation": "${original.file}"
      }
    }
]
```

示例

`CreateWorkflow` 调用返回新工作流程的 ID。

示例响应

```
{
    "WorkflowId": "w-1234abcd5678efghi"
}
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- AWS 适用于 .NET 的 SDK
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS JavaScript V3 版软件开发工具包
- AWS 适用于 PHP 的 SDK V3
- AWS Python 软件开发工具包
- AWS 适用于 Ruby V3 的 SDK

DeleteAccess

允许您删除ServerID和ExternalID参数中指定的访问权限。

请求语法

```
{
   "ExternalId": "string",
   "ServerId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

ExternalId

识别目录中特定组所需的唯一标识符。您关联的组中的用户可以通过使用AWS Transfer Family的启用协议访问您的 Amazon S3 或 Amazon EFS 资源。如果您知道组名，则可以使用 Windows 运行以下命令来查看 SID 值 PowerShell。

```
Get-ADGroup -Filter {samAccountName -like "YourGroupName*"} -Properties
* | Select SamAccountName,ObjectSid
```

在该命令中，替换为您的 YourGroupNameActive Directory 组的名称。

用于验证此参数的正则表达式是由不带空格的大写和小写字母数字字符组成的字符串。此外，还可以包括下划线或以下任何字符：=、.@：/-

类型：字符串

长度限制：最小长度为 1。最大长度为 256。

模式：S-1-[\d-]+

必需：是

ServerId

分配了该用户的服务器的系统分配的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`s-([0-9a-f]{17})`

必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DeleteAgreement

删除提供的 AgreementId 中指定的协议。

请求语法

```
{
   "AgreementId": "string",
   "ServerId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

AgreementId

协议的唯一标识符。创建协议时会返回此标识符。

类型：字符串

长度限制：固定长度为 19。

模式：a-([0-9a-f]{17})

必需：是

ServerId

与要删除的协议关联的服务器标识符。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DeleteCertificate

删除CertificateId参数中指定的证书。

请求语法

```
{
   "CertificateId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

CertificateId

您要删除的证书对象的标识符。

类型：字符串

长度限制：固定长度为 22。

模式：cert-([0-9a-f]{17})

必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DeleteConnector

删除提供的 ConnectorId 中指定的连接器。

请求语法

```
{
   "ConnectorId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

ConnectorId

连接器的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：c-([0-9a-f]{17})

必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DeleteHostKey

删除在HostKeyId参数中指定的主机密钥。

请求语法

```
{
   "HostKeyId": "string",
   "ServerId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

HostKeyId

正在删除的主机密钥的标识符。

类型：字符串

长度限制：固定长度为 25。

模式：hostkey-[0-9a-f]{17}

必需：是

ServerId

包含正在删除的主机密钥的服务器的标识符。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

ThrottlingException

由于请求限制而导致请求被拒绝。

HTTP 状态代码：400

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK

- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DeleteProfile

删除 ProfileId 参数中指定的配置文件。

请求语法

```
{
   "ProfileId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

ProfileId

您要删除的配置文件的标识符。

类型：字符串

长度限制：固定长度为 19。

模式：p-([0-9a-f]{17})

必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DeleteServer

删除您指定的启用文件传输协议的服务器。

此操作未返回任何响应。

请求语法

```
{
   "ServerId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

ServerId

服务器实例的系统分配的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`s-([0-9a-f]{17})`

必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

AccessDeniedException

您没有足够的访问权限，无法执行该操作。

HTTP 状态代码：400

InternalServiceError

当 AWS Transfer Family 服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

示例

示例

以下示例将删除服务器。

示例请求

```
{
   "ServerId": "s-01234567890abcdef"
}
```

示例

如果成功，则不返回任何内容。

示例响应

```
{
```

```
}
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DeleteSshPublicKey

删除用户的 Secure Shell (SSH) 公有密钥。

请求语法

```
{
   "ServerId": "string",
   "SshPublicKeyId": "string",
   "UserName": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

ServerId

已向其分配用户的、文件传输协议服务器实例的系统分配唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`s-([0-9a-f]{17})`

必需：是

SshPublicKeyId

用于引用用户特定 SSH 密钥的唯一标识符。

类型：字符串

长度限制：固定长度为 21。

模式：`key-[0-9a-f]{17}`

必需：是

UserName

一个标识公有密钥被删除的用户的唯一字符串。

类型：字符串

长度限制：长度下限为 3。最大长度为 100。

模式：`[\w][\w@.-]{2,99}`

必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

ThrottlingException

由于请求限制而导致请求被拒绝。

HTTP 状态代码：400

示例

示例

以下示例删除用户的 SSH 公有密钥。

示例请求

```
{
   "ServerId": "s-01234567890abcdef",
   "SshPublicKeyId": "MyPublicKey",
   "UserName": "my_user"
}
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DeleteUser

删除属于您指定的启用文件传输协议的服务器的用户。

此操作未返回任何响应。

> **Note**
>
> 当您从服务器上删除用户时，该用户的信息就会丢失。

请求语法

```
{
   "ServerId": "string",
   "UserName": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

ServerId

系统为分配了用户的服务器实例分配的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`s-([0-9a-f]{17})`

必需：是

UserName

一个标识正在从服务器删除的用户的唯一字符串。

类型：字符串

长度限制：长度下限为 3。最大长度为 100。

模式：`[\w][\w@.-]{2,99}`

必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

示例

示例

以下示例将删除一个 Transfer Family 用户。

示例请求

```
{
   "ServerId": "s-01234567890abcdef",
   "UserNames": "my_user"
}
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DeleteWorkflow

删除指定的工作流程。

请求语法

```
{
   "WorkflowId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

WorkflowId

工作流的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：w-([a-z0-9]{17})

必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

AccessDeniedException

您没有足够的访问权限，无法执行该操作。

HTTP 状态代码：400

InternalServiceError

当 AWS Transfer Family 服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DescribeAccess

描述分配给启用文件传输协议的特定服务器的访问权限，该访问权限由其 ServerId 属性和其 ExternalId 标识。

此调用的响应返回与指定的 ServerId 值关联的访问权限的属性。

请求语法

```
{
   "ExternalId": "string",
   "ServerId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

ExternalId

识别目录中特定组所需的唯一标识符。您关联的组中的用户可以通过使用AWS Transfer Family的启用协议访问您的 Amazon S3 或 Amazon EFS 资源。如果您知道组名，则可以使用 Windows 运行以下命令来查看 SID 值 PowerShell。

```
Get-ADGroup -Filter {samAccountName -like "YourGroupName*"} -Properties
* | Select SamAccountName,ObjectSid
```

在该命令中，替换为您的 YourGroupNameActive Directory 组的名称。

用于验证此参数的正则表达式是由不带空格的大写和小写字母数字字符组成的字符串。此外，还可以包括下划线或以下任何字符：=、.@：/-

类型：字符串

长度限制：最小长度为 1。最大长度为 256。

模式：S-1-[\d-]+

必需：是

ServerId

分配了该访问权限的服务器的系统分配的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

必需：是

响应语法

```
{
   "Access": {
      "ExternalId": "string",
      "HomeDirectory": "string",
      "HomeDirectoryMappings": [
         {
            "Entry": "string",
            "Target": "string",
            "Type": "string"
         }
      ],
      "HomeDirectoryType": "string",
      "Policy": "string",
      "PosixProfile": {
         "Gid": number,
         "SecondaryGids": [ number ],
         "Uid": number
      },
      "Role": "string"
   },
   "ServerId": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

Access

访问权限连接到的服务器的外部标识符。

类型：DescribedAccess 对象

ServerId

分配了该访问权限的服务器的系统分配的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`s-([0-9a-f]{17})`

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DescribeAgreement

描述由AgreementId标识的协议。

请求语法

```
{
   "AgreementId": "string",
   "ServerId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

AgreementId

协议的唯一标识符。创建协议时会返回此标识符。

类型：字符串

长度限制：固定长度为 19。

模式：a-([0-9a-f]{17})

必需：是

ServerId

与协议关联的服务器标识符。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

必需：是

响应语法

```
{
```

```
   "Agreement": {
      "AccessRole": "string",
      "AgreementId": "string",
      "Arn": "string",
      "BaseDirectory": "string",
      "Description": "string",
      "LocalProfileId": "string",
      "PartnerProfileId": "string",
      "ServerId": "string",
      "Status": "string",
      "Tags": [
         {
            "Key": "string",
            "Value": "string"
         }
      ]
   }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

Agreement

指定协议的详细信息，以DescribedAgreement对象形式返回。

类型：DescribedAgreement 对象

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DescribeCertificate

描述由CertificateId标识的证书。

请求语法

```
{
   "CertificateId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

CertificateId

已导入证书的标识符数组。您可以使用此标识符来处理配置文件和合作伙伴配置文件。

类型：字符串

长度限制：固定长度为 22。

模式：cert-([0-9a-f]{17})

必需：是

响应语法

```
{
   "Certificate": {
      "ActiveDate": number,
      "Arn": "string",
      "Certificate": "string",
      "CertificateChain": "string",
      "CertificateId": "string",
      "Description": "string",
      "InactiveDate": number,
      "NotAfterDate": number,
      "NotBeforeDate": number,
```

```
      "Serial": "string",
      "Status": "string",
      "Tags": [
         {
            "Key": "string",
            "Value": "string"
         }
      ],
      "Type": "string",
      "Usage": "string"
   }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

Certificate

指定证书的详细信息，以对象形式返回。

类型：DescribedCertificate 对象

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DescribeConnector

描述由ConnectorId.标识的连接器

请求语法

```
{
   "ConnectorId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

ConnectorId

连接器的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：c-([0-9a-f]{17})

必需：是

响应语法

```
{
   "Connector": {
      "AccessRole": "string",
      "Arn": "string",
      "As2Config": {
         "BasicAuthSecretId": "string",
         "Compression": "string",
         "EncryptionAlgorithm": "string",
         "LocalProfileId": "string",
         "MdnResponse": "string",
         "MdnSigningAlgorithm": "string",
```

```
         "MessageSubject": "string",
         "PartnerProfileId": "string",
         "SigningAlgorithm": "string"
      },
      "ConnectorId": "string",
      "LoggingRole": "string",
      "SecurityPolicyName": "string",
      "ServiceManagedEgressIpAddresses": [ "string" ],
      "SftpConfig": {
         "TrustedHostKeys": [ "string" ],
         "UserSecretId": "string"
      },
      "Tags": [
         {
            "Key": "string",
            "Value": "string"
         }
      ],
      "Url": "string"
   }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

Connector

包含连接器详细信息的结构。

类型：DescribedConnector 对象

错误

有关所有操作的常见错误信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family 服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 Transfer Family 服务找不到资源时，就会 AWS引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 Trans AWS fer Family 服务不可用。

HTTP 状态代码：500

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- AWS 适用于 .NET 的 SDK
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS JavaScript V3 版软件开发工具包
- AWS 适用于 PHP 的 SDK V3
- AWS Python 软件开发工具包
- AWS 适用于 Ruby V3 的 SDK

DescribeExecution

您可以使用DescribeExecution来检查指定工作流程的执行细节。

> **Note**
>
> 此 API 调用仅返回正在进行的工作流程的详细信息。
> 如果您为未进行中的执行提供了 ID，或者如果执行与指定的工作流程 ID 不匹配，则会收到ResourceNotFound异常。

请求语法

```
{
   "ExecutionId": "string",
   "WorkflowId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

ExecutionId

用于执行工作流程的唯一标识符。

类型：字符串

长度限制：固定长度为 36。

模式：[0-9a-fA-F]{8}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{12}

必需：是

WorkflowId

工作流的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：w-([a-z0-9]{17})

必需：是

响应语法

```
{
   "Execution": {
      "ExecutionId": "string",
      "ExecutionRole": "string",
      "InitialFileLocation": {
         "EfsFileLocation": {
            "FileSystemId": "string",
            "Path": "string"
         },
         "S3FileLocation": {
            "Bucket": "string",
            "Etag": "string",
            "Key": "string",
            "VersionId": "string"
         }
      },
      "LoggingConfiguration": {
         "LoggingRole": "string",
         "LogGroupName": "string"
      },
      "PosixProfile": {
         "Gid": number,
         "SecondaryGids": [ number ],
         "Uid": number
      },
      "Results": {
         "OnExceptionSteps": [
            {
               "Error": {
                  "Message": "string",
                  "Type": "string"
               },
               "Outputs": "string",
               "StepType": "string"
            }
```

```
         ],
         "Steps": [
            {
               "Error": {
                  "Message": "string",
                  "Type": "string"
               },
               "Outputs": "string",
               "StepType": "string"
            }
         ]
      },
      "ServiceMetadata": {
         "UserDetails": {
            "ServerId": "string",
            "SessionId": "string",
            "UserName": "string"
         }
      },
      "Status": "string"
   },
   "WorkflowId": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

Execution

包含工作流程执行详细信息的结构。

类型：DescribedExecution 对象

WorkflowId

工作流的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`w-([a-z0-9]{17})`

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DescribeHostKey

返回HostKeyId和ServerId指定的主机密钥的详细信息。

请求语法

```
{
   "HostKeyId": "string",
   "ServerId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

HostKeyId

您要描述的主机密钥的标识符。

类型：字符串

长度限制：固定长度为 25。

模式：hostkey-[0-9a-f]{17}

必需：是

ServerId

包含要描述的主机密钥的服务器的标识符。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

必需：是

响应语法

```
{
```

```
   "HostKey": {
      "Arn": "string",
      "DateImported": number,
      "Description": "string",
      "HostKeyFingerprint": "string",
      "HostKeyId": "string",
      "Tags": [
         {
            "Key": "string",
            "Value": "string"
         }
      ],
      "Type": "string"
   }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

HostKey

返回指定主机密钥的详细信息。

类型：DescribedHostKey 对象

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DescribeProfile

返回由ProfileId指定的配置文件的详细信息。

请求语法

```
{
   "ProfileId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

ProfileId

您要描述的配置文件的标识符。

类型：字符串

长度限制：固定长度为 19。

模式：p-([0-9a-f]{17})

必需：是

响应语法

```
{
   "Profile": {
      "Arn": "string",
      "As2Id": "string",
      "CertificateIds": [ "string" ],
      "ProfileId": "string",
      "ProfileType": "string",
      "Tags": [
         {
            "Key": "string",
            "Value": "string"
```

```
        }
      ]
    }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

Profile

指定配置文件的详细信息，以对象形式返回。

类型：DescribedProfile 对象

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DescribeSecurityPolicy

描述附加到您的服务器或 SFTP 连接器的安全策略。响应包含安全策略的属性的描述。有关安全策略的更多信息，请参阅使用服务器的安全策略或使用 SFTP 连接器的安全策略。

请求语法

```
{
   "SecurityPolicyName": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

SecurityPolicyName

指定要获取详细信息的安全策略的文本名称。

类型：字符串

长度限制：长度下限为 0。最大长度为 100。

模式：Transfer[A-Za-z0-9]*SecurityPolicy-[A-Za-z0-9-]+

必需：是

响应语法

```
{
   "SecurityPolicy": {
      "Fips": boolean,
      "Protocols": [ "string" ],
      "SecurityPolicyName": "string",
      "SshCiphers": [ "string" ],
      "SshHostKeyAlgorithms": [ "string" ],
      "SshKexs": [ "string" ],
      "SshMacs": [ "string" ],
      "TlsCiphers": [ "string" ],
```

```
        "Type": "string"
    }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

SecurityPolicy

包含安全策略属性的数组。

类型：DescribedSecurityPolicy 对象

错误

有关所有操作的常见错误信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family 服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 Transfer Family 服务找不到资源时，就会 AWS引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 Trans AWS fer Family 服务不可用。

HTTP 状态代码：500

示例

示例

以下示例命令将安全策略名称作为参数，并返回指定安全策略的算法。

示例请求

```
aws transfer describe-security-policy --security-policy-name "TransferSecurityPolicy-
FIPS-2023-05"
```

示例响应

```
{
    "SecurityPolicy": {
        "Fips": true,
        "SecurityPolicyName": "TransferSecurityPolicy-FIPS-2023-05",
        "SshCiphers": [
            "aes256-gcm@openssh.com",
            "aes128-gcm@openssh.com",
            "aes256-ctr",
            "aes192-ctr"
        ],
        "SshKexs": [
            "diffie-hellman-group16-sha512",
            "diffie-hellman-group18-sha512",
            "diffie-hellman-group-exchange-sha256"
        ],
        "SshMacs": [
            "hmac-sha2-256-etm@openssh.com",
            "hmac-sha2-512-etm@openssh.com"
        ],
        "TlsCiphers": [
            "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384"
        ]
    }
```

```
}
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- AWS 适用于 .NET 的 SDK
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS JavaScript V3 版软件开发工具包
- AWS 适用于 PHP 的 SDK V3
- AWS Python 软件开发工具包
- AWS 适用于 Ruby V3 的 SDK

DescribeServer

描述通过传递 ServerId 参数指定的启用文件传输协议的服务器。

响应包含服务器属性的描述。当您将EndpointType设置为 VPC 时，响应将包含EndpointDetails。

请求语法

```
{
   "ServerId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

ServerId

系统为服务器分配的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

必需：是

响应语法

```
{
   "Server": {
      "Arn": "string",
      "As2ServiceManagedEgressIpAddresses": [ "string" ],
      "Certificate": "string",
      "Domain": "string",
      "EndpointDetails": {
         "AddressAllocationIds": [ "string" ],
         "SecurityGroupIds": [ "string" ],
         "SubnetIds": [ "string" ],
```

```
            "VpcEndpointId": "string",
            "VpcId": "string"
        },
        "EndpointType": "string",
        "HostKeyFingerprint": "string",
        "IdentityProviderDetails": {
            "DirectoryId": "string",
            "Function": "string",
            "InvocationRole": "string",
            "SftpAuthenticationMethods": "string",
            "Url": "string"
        },
        "IdentityProviderType": "string",
        "LoggingRole": "string",
        "PostAuthenticationLoginBanner": "string",
        "PreAuthenticationLoginBanner": "string",
        "ProtocolDetails": {
            "As2Transports": [ "string" ],
            "PassiveIp": "string",
            "SetStatOption": "string",
            "TlsSessionResumptionMode": "string"
        },
        "Protocols": [ "string" ],
        "S3StorageOptions": {
            "DirectoryListingOptimization": "string"
        },
        "SecurityPolicyName": "string",
        "ServerId": "string",
        "State": "string",
        "StructuredLogDestinations": [ "string" ],
        "Tags": [
            {
                "Key": "string",
                "Value": "string"
            }
        ],
        "UserCount": number,
        "WorkflowDetails": {
            "OnPartialUpload": [
                {
                    "ExecutionRole": "string",
                    "WorkflowId": "string"
                }
            ],
```

```
         "OnUpload": [
            {
               "ExecutionRole": "string",
               "WorkflowId": "string"
            }
         ]
      }
   }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

Server

一个数组，包含您指定的ServerID服务器的属性。

类型：DescribedServer 对象

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

示例

示例

以下示例返回分配给服务器的属性。

示例请求

```
{
   "ServerId": "s-01234567890abcdef"
}
```

示例

此示例说明了的一种用法 DescribeServer。

示例响应

```
{
    "Server": {
        "Arn": "arn:aws:transfer:us-east-1:176354371281:server/s-01234567890abcdef",
        "EndpointDetails": {
            "AddressAllocationIds": [
                "eipalloc-01a2eabe3c04d5678",
                "eipalloc-102345be"
            ],
            "SubnetIds": [
                "subnet-047eaa7f0187a7cde",
                "subnet-0a2d0f474daffde18"
            ],
            "VpcEndpointId": "vpce-03fe0080e7cb008b8",
            "VpcId": "vpc-09047a51f1c8e1634"
        },
        "EndpointType": "VPC",
```

```
        "HostKeyFingerprint": "your host key,
        "IdentityProviderType": "SERVICE_MANAGED",
        "ServerId": "s-01234567890abcdef",
        "State": "ONLINE",
        "Tags": [],
        "UserCount": 0
    }
}
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DescribeUser

描述分配给启用了文件传输协议的特定服务器的用户，由其ServerId属性标识。

此调用的响应返回与指定ServerId值关联的用户的属性。

请求语法

```
{
   "ServerId": "string",
   "UserName": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

ServerId

系统为已分配此用户的服务器分配的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

必需：是

UserName

分配给一个或多个服务器的用户的名称。用户名是使用AWS Transfer Family服务和执行文件传输任务的登录凭证的一部分。

类型：字符串

长度限制：长度下限为 3。最大长度为 100。

模式：[\w][\w@.-]{2,99}

必需：是

响应语法

```
{
   "ServerId": "string",
   "User": {
      "Arn": "string",
      "HomeDirectory": "string",
      "HomeDirectoryMappings": [
         {
            "Entry": "string",
            "Target": "string",
            "Type": "string"
         }
      ],
      "HomeDirectoryType": "string",
      "Policy": "string",
      "PosixProfile": {
         "Gid": number,
         "SecondaryGids": [ number ],
         "Uid": number
      },
      "Role": "string",
      "SshPublicKeys": [
         {
            "DateImported": number,
            "SshPublicKeyBody": "string",
            "SshPublicKeyId": "string"
         }
      ],
      "Tags": [
         {
            "Key": "string",
            "Value": "string"
         }
      ],
      "UserName": "string"
   }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

ServerId

系统为已分配此用户的服务器分配的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`s-([0-9a-f]{17})`

User

一个包含您指定`ServerID`值的 Transfer Family 用户属性的数组。

类型：DescribedUser 对象

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

示例

示例

以下示例显示了现有用户的详细信息。

示例请求

```
aws transfer describe-user --server-id s-1111aaaa2222bbbb3 --user-name bob-test
```

示例响应

```
{
    "ServerId": "s-1111aaaa2222bbbb3",
    "User": {
        "Arn": "arn:aws:transfer:us-east-1:111122223333:user/s-1111aaaa2222bbbb3/bob-
test",
        "HomeDirectory": "/DOC-EXAMPLE-BUCKET",
        "HomeDirectoryType": "PATH",
        "Role": "arn:aws:iam::111122223333:role/bob-role",
        "SshPublicKeys": [
            {
                "DateImported": "2022-03-31T12:27:52.614000-04:00",
                "SshPublicKeyBody": "ssh-rsa AAAAB3NzaC1yc..... bobusa@mycomputer.us-
east-1.amaazon.com",
                "SshPublicKeyId": "key-abcde12345fghik67"
            }
        ],
        "Tags": [],
        "UserName": "bob-test"
    }
}
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK

- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DescribeWorkflow

描述指定的工作流程。

请求语法

```
{
   "WorkflowId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

WorkflowId

工作流的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：w-([a-z0-9]{17})

必需：是

响应语法

```
{
   "Workflow": {
      "Arn": "string",
      "Description": "string",
      "OnExceptionSteps": [
         {
            "CopyStepDetails": {
               "DestinationFileLocation": {
                  "EfsFileLocation": {
                     "FileSystemId": "string",
                     "Path": "string"
                  },
                  "S3FileLocation": {
```

```
                "Bucket": "string",
                "Key": "string"
            }
        },
        "Name": "string",
        "OverwriteExisting": "string",
        "SourceFileLocation": "string"
    },
    "CustomStepDetails": {
        "Name": "string",
        "SourceFileLocation": "string",
        "Target": "string",
        "TimeoutSeconds": number
    },
    "DecryptStepDetails": {
        "DestinationFileLocation": {
            "EfsFileLocation": {
                "FileSystemId": "string",
                "Path": "string"
            },
            "S3FileLocation": {
                "Bucket": "string",
                "Key": "string"
            }
        },
        "Name": "string",
        "OverwriteExisting": "string",
        "SourceFileLocation": "string",
        "Type": "string"
    },
    "DeleteStepDetails": {
        "Name": "string",
        "SourceFileLocation": "string"
    },
    "TagStepDetails": {
        "Name": "string",
        "SourceFileLocation": "string",
        "Tags": [
            {
                "Key": "string",
                "Value": "string"
            }
        ]
    },
```

```
            "Type": "string"
         }
      ],
      "Steps": [
         {
            "CopyStepDetails": {
               "DestinationFileLocation": {
                  "EfsFileLocation": {
                     "FileSystemId": "string",
                     "Path": "string"
                  },
                  "S3FileLocation": {
                     "Bucket": "string",
                     "Key": "string"
                  }
               },
               "Name": "string",
               "OverwriteExisting": "string",
               "SourceFileLocation": "string"
            },
            "CustomStepDetails": {
               "Name": "string",
               "SourceFileLocation": "string",
               "Target": "string",
               "TimeoutSeconds": number
            },
            "DecryptStepDetails": {
               "DestinationFileLocation": {
                  "EfsFileLocation": {
                     "FileSystemId": "string",
                     "Path": "string"
                  },
                  "S3FileLocation": {
                     "Bucket": "string",
                     "Key": "string"
                  }
               },
               "Name": "string",
               "OverwriteExisting": "string",
               "SourceFileLocation": "string",
               "Type": "string"
            },
            "DeleteStepDetails": {
               "Name": "string",
```

```
                "SourceFileLocation": "string"
            },
            "TagStepDetails": {
               "Name": "string",
               "SourceFileLocation": "string",
               "Tags": [
                  {
                     "Key": "string",
                     "Value": "string"
                  }
               ]
            },
            "Type": "string"
         }
      ],
      "Tags": [
         {
            "Key": "string",
            "Value": "string"
         }
      ],
      "WorkflowId": "string"
   }
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

Workflow

包含工作流程详细信息的结构。

类型：DescribedWorkflow 对象

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ImportCertificate

导入创建本地（AS2）配置文件和合作伙伴配置文件时所需的签名和加密证书。

请求语法

```
{
   "ActiveDate": number,
   "Certificate": "string",
   "CertificateChain": "string",
   "Description": "string",
   "InactiveDate": number,
   "PrivateKey": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ],
   "Usage": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

ActiveDate

指定证书何时生效的可选日期。

类型：时间戳

必需：否

Certificate

- 对于 CLI，请提供 URI 格式的证书的文件路径。例如，`--certificate file://encryption-cert.pem`。或者，您也可以提供原始内容。
- 对于 SDK，请指定证书文件的原始内容。例如：`--certificate "`cat encryption-cert.pem`"`。

类型：字符串

长度限制：长度下限为 1。长度上限为 16384。

模式：`[\u0009\u000A\u000D\u0020-\u00FF]*`

必需：是

CertificateChain

构成正导入证书链的可选证书列表。

类型：字符串

长度限制：长度下限为 1。最大长度为 2097152。

模式：`[\u0009\u000A\u000D\u0020-\u00FF]*`

必需：否

Description

帮助识别证书的简短描述。

类型：字符串

长度限制：最小长度为 1。最大长度为 200。

模式：`[\p{Graph}]+`

必需：否

InactiveDate

指定证书何时失效的可选日期。

类型：时间戳

必需：否

PrivateKey

- 对于 CLI，请为 URI 格式的私有密钥提供文件路径。例如，`--private-key file://encryption-key.pem`。或者，您可以提供私有密钥文件的原始内容。
- 对于 SDK，请指定私有密钥文件的原始内容。例如，`--private-key "`cat encryption-key.pem`"`

类型：字符串

长度限制：长度下限为 1。长度上限为 16384。

模式：`[\u0009\u000A\u000D\u0020-\u00FF]*`

必需：否

Tags

可用于分组和搜索证书的键/值对。

类型：Tag 对象数组

数组成员：最少 1 个物品。最多 50 项。

必需：否

Usage

指定如何使用此证书。它可以通过以下方式使用：

- SIGNING: 用于签署 AS2 消息
- ENCRYPTION: 用于加密 AS2 消息
- TLS: 用于保护通过 HTTPS 发送的 AS2 通信

类型：字符串

有效值：`SIGNING | ENCRYPTION`

必需：是

响应语法

```
{
   "CertificateId": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

CertificateId

已导入证书的标识符数组。您可以使用此标识符来处理配置文件和合作伙伴配置文件。

类型：字符串

长度限制：固定长度为 22。

模式：`cert-([0-9a-f]{17})`

错误

有关所有操作的常见错误信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family 服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 Transfer Family 服务找不到资源时，就会 AWS引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 Trans AWS fer Family 服务不可用。

HTTP 状态代码：500

示例

示例

以下示例导入用于加密的证书。在第一个命令中，我们提供证书和证书链文件的内容。对 SDK 命令使用此格式。

```
aws transfer import-certificate --usage ENCRYPTION --certificate "`cat encryption-
cert.pem`" \
    --private-key "`cat encryption-key.pem`" --certificate-chain "`cat root-ca.pem`"
```

示例

以下示例与前面的命令相同，只是我们提供了私有密钥、证书和证书链文件的文件位置。如果您使用的是 SDK，则此版本的命令不起作用。

```
aws transfer import-certificate --usage ENCRYPTION --certificate file://encryption-
cert.pem \
    --private-key file://encryption-key.pem --certificate-chain file://root-ca.pem
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- AWS 适用于 .NET 的 SDK
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS JavaScript V3 版软件开发工具包
- AWS 适用于 PHP 的 SDK V3
- AWS Python 软件开发工具包
- AWS 适用于 Ruby V3 的 SDK

ImportHostKey

添加由 `ServerId` 参数指定的服务器主机密钥。

请求语法

```
{
   "Description": "string",
   "HostKeyBody": "string",
   "ServerId": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

Description

标识此主机密钥的文本描述。

类型：字符串

长度限制：长度下限为 0。最大长度为 200。

模式：`[\p{Print}]*`

必需：否

HostKeyBody

SSH 密钥对的私有密钥部分。

AWS Transfer Family 接受 RSA、ECDSA 和 ED25519 密钥。

类型：字符串

长度限制：长度下限为 0。最大长度为 4096。

必需：是

ServerId

包含要导入的主机密钥的服务器的标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`s-([0-9a-f]{17})`

必需：是

Tags

可用于分组和搜索主机密钥的键-值对。

类型：Tag 对象数组

数组成员：最少 1 个物品。最多 50 项。

必需：否

响应语法

```
{
   "HostKeyId": "string",
   "ServerId": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

HostKeyId

返回导入密钥的主机密钥标识符。

类型：字符串

长度限制：固定长度为 25。

模式：`hostkey-[0-9a-f]{17}`

ServerId

返回包含导入密钥的服务器标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`s-([0-9a-f]{17})`

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceExistsException

请求的资源不存在，或者存在于为命令指定的区域以外的区域。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

ThrottlingException

由于请求限制而导致请求被拒绝。

HTTP 状态代码：400

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ImportSshPublicKey

向 Transfer Family 用户添加一个由ServerId标识、分配给启用文件传输协议的特定服务器（由标识）的值的 Secure Shell (SSH)。向 Transfer Family 用户添加 Secure Shell (SSH) 公有密钥，该用户由指定给启用文件传输协议的特定服务器的UserName值标识，该服务器由标识。

响应返回的UserName值、ServerId值和SshPublicKeyId名称。

请求语法

```
{
   "ServerId": "string",
   "SshPublicKeyBody": "string",
   "UserName": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

ServerId

系统为服务器分配的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

必需：是

SshPublicKeyBody

SSH 密钥对的公有密钥部分。

AWS Transfer Family 接受 RSA、ECDSA 和 ED25519 密钥。

类型：字符串

长度约束：最小长度为 0。最大长度为 2048。

必需：是

UserName

分配给一台或多台服务器的 Transfer Family 用户的名称。

类型：字符串

长度约束：最小长度为 3。最大长度为 100。

模式：[\w][\w@.-]{2,99}

必需：是

响应语法

```
{
   "ServerId": "string",
   "SshPublicKeyId": "string",
   "UserName": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

ServerId

系统为服务器分配的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

SshPublicKeyId

导入的系统为公有密钥命名的名称。

类型：字符串

长度限制：固定长度为 21。

模式：`key-[0-9a-f]{17}`

UserName

分配给指定`ServerID`值的用户名。

类型：字符串

长度约束：最小长度为 3。最大长度为 100。

模式：`[\w][\w@.-]{2,99}`

错误

有关所有操作返回的常见错误的信息，请参阅 常见错误。

InternalServiceError

当 AWS Transfer Family 服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceExistsException

请求的资源不存在，或者存在于为命令指定的区域以外的区域。

HTTP 状态代码：400

ResourceNotFoundException

当 Transfer Family 服务找不到资源时，就会 AWS引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 Trans AWS fer Family 服务不可用。

HTTP 状态代码：500

ThrottlingException

由于请求限制而导致请求被拒绝。

HTTP 状态代码：400

示例

示例

此命令导入存储在id_ecdsa.pub文件中的 ECDSA 密钥。

```
aws transfer import-ssh-public-key --server-id s-021345abcdef6789 --ssh-public-key-body
 file://id_ecdsa.pub --user-name jane-doe
```

示例

如果运行上一个命令，系统将返回以下信息。

```
{
    "ServerId": "s-021345abcdef6789",
   "SshPublicKeyId": "key-1234567890abcdef0",
   "UserName": "jane-doe"
}
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- AWS 适用于 .NET 的 SDK
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS JavaScript V3 版软件开发工具包
- AWS 适用于 PHP 的 SDK V3
- AWS Python 软件开发工具包

- AWS 适用于 Ruby V3 的 SDK

ListAccesses

列出服务器上所有访问权限的详细信息。

请求语法

```
{
   "MaxResults": number,
   "NextToken": "string",
   "ServerId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

MaxResults

指定要返回的访问权限 SID 的最大数目。

类型：整数

有效范围：最小值为 1。最大值为 1000。

必需：否

NextToken

当您可以从`ListAccesses`调用中获得其他结果时，将在输出中返回一个`NextToken`参数。然后，您可以向`NextToken`参数传入后续命令以继续列出其他访问权限。

类型：字符串

长度限制：最小长度为 1。长度上限为 6144。

必需：否

ServerId

有分配到用户的服务器的系统分配的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

必需：是

响应语法

```
{
   "Accesses": [
      {
         "ExternalId": "string",
         "HomeDirectory": "string",
         "HomeDirectoryType": "string",
         "Role": "string"
      }
   ],
   "NextToken": "string",
   "ServerId": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

Accesses

根据您指定的 ServerId 值返回访问权限及其属性。

类型：ListedAccess 对象数组

NextToken

当您可以从ListAccesses调用中获得其他结果时，将在输出中返回一个NextToken参数。然后，您可以向NextToken参数传入后续命令以继续列出其他访问权限。

类型：字符串

长度限制：最小长度为 1。长度上限为 6144。

ServerId

有分配到用户的服务器的系统分配的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`s-([0-9a-f]{17})`

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidNextTokenException

传递的`NextToken`参数无效。

HTTP 状态代码：400

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ListAgreements

返回由您提供的 ServerId 标识的服务器协议列表。如果要将结果限制为某个数值，请为 MaxResults 参数提供一个值。如果您之前运行过该命令并收到了的NextToken值，则可以提供该值，以继续从上次中断的地方列出配置文件。

请求语法

```
{
   "MaxResults": number,
   "NextToken": "string",
   "ServerId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

MaxResults

要返回的最大结果数量。

类型：整数

有效范围：最小值为 1。最大值为 1000。

必需：否

NextToken

当您可以从ListAgreements调用中获得其他结果时，将在输出中返回一个NextToken参数。然后，您可以向NextToken参数传入后续命令以继续列出其他标签。

类型：字符串

长度限制：最小长度为 1。长度上限为 6144。

必需：否

ServerId

需要协议列表的服务器标识符。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

必需：是

响应语法

```
{
   "Agreements": [
      {
         "AgreementId": "string",
         "Arn": "string",
         "Description": "string",
         "LocalProfileId": "string",
         "PartnerProfileId": "string",
         "ServerId": "string",
         "Status": "string"
      }
   ],
   "NextToken": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

Agreements

返回数组，其中的每个项目都包含主机密钥的详细信息。

类型：ListedAgreement 对象数组

NextToken

返回一令牌，您可以使用此令牌再次调用 ListAgreements，并接收其他结果（如有）。

类型：字符串

长度限制：最小长度为 1。长度上限为 6144。

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidNextTokenException

传递的NextToken参数无效。

HTTP 状态代码：400

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK

- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ListCertificates

返回已导入AWS Transfer Family的当前证书的列表。如果要将结果限制为某个数值，请为MaxResults参数提供一个值。如果您之前运行过该命令并收到了该NextToken参数的值，则可以提供该值，以继续从上次中断的地方列出配置文件。

请求语法

```
{
   "MaxResults": number,
   "NextToken": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

MaxResults

要返回的证书的最大数量。

类型：整数

有效范围：最小值为 1。最大值为 1000。

必需：否

NextToken

当您可以从ListCertificates调用中获得其他结果时，将在输出中返回一个NextToken参数。然后，您可以向NextToken参数传入后续命令以继续列出其他证书。

类型：字符串

长度限制：最小长度为 1。长度上限为 6144。

必需：否

响应语法

```
{
```

```
   "Certificates": [
      {
         "ActiveDate": number,
         "Arn": "string",
         "CertificateId": "string",
         "Description": "string",
         "InactiveDate": number,
         "Status": "string",
         "Type": "string",
         "Usage": "string"
      }
   ],
   "NextToken": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

Certificates

返回ListCertificates调用中指定的证书的数组。

类型：ListedCertificate 对象数组

NextToken

返回下一个令牌，您可以使用该令牌列出下一个证书。

类型：字符串

长度限制：最小长度为 1。长度上限为 6144。

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidNextTokenException

传递的NextToken参数无效。

HTTP 状态代码：400

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ListConnectors

列出指定区域中的连接器。

请求语法

```
{
   "MaxResults": number,
   "NextToken": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

MaxResults

要返回的配置的最大数量。

类型：整数

有效范围：最小值为 1。最大值为 1000。

必需：否

NextToken

当您可以从ListConnectors调用中获得其他结果时，将在输出中返回一个NextToken参数。然后，您可以向NextToken参数传入后续命令以继续列出其他标签。

类型：字符串

长度限制：最小长度为 1。长度上限为 6144。

必需：否

响应语法

```
{
   "Connectors": [
```

```
      {
         "Arn": "string",
         "ConnectorId": "string",
         "Url": "string"
      }
   ],
   "NextToken": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

Connectors

返回数组，其中的每个项目都包含主机密钥的详细信息。

类型：ListedConnector 对象数组

NextToken

返回一令牌，您可以使用此令牌再次调用 `ListConnectors`，并接收其他结果（如有）。

类型：字符串

长度限制：最小长度为 1。长度上限为 6144。

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidNextTokenException

传递的`NextToken`参数无效。

HTTP 状态代码：400

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ListExecutions

列出指定工作流程的执行进度。

> **Note**
>
> 如果找不到指定的工作流程 ID，则 `ListExecutions` 返回 `ResourceNotFound`异常。

请求语法

```
{
   "MaxResults": number,
   "NextToken": "string",
   "WorkflowId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

MaxResults

指定要返回的最大记录数。

类型：整数

有效范围：最小值为 1。最大值为 1000。

必需：否

NextToken

`ListExecutions`在输出中返回`NextToken`参数。然后，您可以在后续命令中传递`NextToken`参数以继续列出其他工作流程。

例如，这对分页很有用。如果您的工作流程有 100 项执行，则可能只想列出前 10 项执行。如果是，请通过指定`max-results`调用 API：

```
aws transfer list-executions --max-results 10
```

这将返回前 10 次执行的详细信息，以及指向第 11 次执行的指针 (NextToken)。现在，您可以再次调用 API，提供您收到的 NextToken值：

```
aws transfer list-executions --max-results 10 --next-token
$somePointerReturnedFromPreviousListResult
```

此调用返回接下来的 10 次执行，即第 11 次到第 20 次执行。然后，您可重复调用，直到返回所有 100 项执行的详细信息。

类型：字符串

长度限制：最小长度为 1。长度上限为 6144。

必需：否

WorkflowId

工作流的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：w-([a-z0-9]{17})

必需：是

响应语法

```
{
   "Executions": [
      {
         "ExecutionId": "string",
         "InitialFileLocation": {
            "EfsFileLocation": {
               "FileSystemId": "string",
               "Path": "string"
            },
            "S3FileLocation": {
               "Bucket": "string",
               "Etag": "string",
               "Key": "string",
               "VersionId": "string"
```

```
            }
         },
         "ServiceMetadata": {
            "UserDetails": {
               "ServerId": "string",
               "SessionId": "string",
               "UserName": "string"
            }
         },
         "Status": "string"
      }
   ],
   "NextToken": "string",
   "WorkflowId": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

Executions

以 ListedExecution 数组形式返回每次执行的详细信息。

类型：ListedExecution 对象数组

NextToken

ListExecutions在输出中返回NextToken参数。然后，您可以在后续命令中传递NextToken参数以继续列出其他工作流程。

类型：字符串

长度限制：最小长度为 1。长度上限为 6144。

WorkflowId

工作流的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：w-([a-z0-9]{17})

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidNextTokenException

传递的NextToken参数无效。

HTTP 状态代码：400

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK

- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ListHostKeys

返回由 ServerId 参数指定的服务器主机密钥列表。

请求语法

```
{
   "MaxResults": number,
   "NextToken": "string",
   "ServerId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

MaxResults

要返回的最大行数。

类型：整数

有效范围：最小值为 1。最大值为 1000。

必需：否

NextToken

如果未返回其他结果，则返回NextToken参数。您可以将该值用于后续调用ListHostKeys，以继续列出结果。

类型：字符串

长度限制：最小长度为 1。长度上限为 6144。

必需：否

ServerId

包含要描述的主机密钥的服务器的标识符。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

必需：是

响应语法

```
{
   "HostKeys": [
      {
         "Arn": "string",
         "DateImported": number,
         "Description": "string",
         "Fingerprint": "string",
         "HostKeyId": "string",
         "Type": "string"
      }
   ],
   "NextToken": "string",
   "ServerId": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

HostKeys

返回数组，其中的每个项目都包含主机密钥的详细信息。

类型：ListedHostKey 对象数组

NextToken

返回一令牌，您可以使用此令牌再次调用 ListHostKeys，并接收其他结果（如有）。

类型：字符串

长度限制：最小长度为 1。长度上限为 6144。

ServerId

返回包含所列出主机密钥的服务器标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`s-([0-9a-f]{17})`

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidNextTokenException

传递的`NextToken`参数无效。

HTTP 状态代码：400

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ListProfiles

返回系统配置文件列表。如果要将结果限制为某个数值，请为 MaxResults 参数提供一个值。如果您之前运行过该命令并收到了的NextToken值，则可以提供该值，以继续从上次中断的地方列出配置文件。

请求语法

```
{
   "MaxResults": number,
   "NextToken": "string",
   "ProfileType": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

MaxResults

配置文件的最大数量。

类型：整数

有效范围：最小值为 1。最大值为 1000。

必需：否

NextToken

如果未返回其他结果，则返回NextToken参数。您可以将该值用于后续调用ListProfiles，以继续列出结果。

类型：字符串

长度限制：最小长度为 1。长度上限为 6144。

必需：否

ProfileType

指示是否仅列出 LOCAL 类型配置文件或仅列出 PARTNER 类型配置文件。如果请求中未提供，则该命令会列出所有类型的配置文件。

类型：字符串

有效值：LOCAL | PARTNER

必需：否

响应语法

```
{
   "NextToken": "string",
   "Profiles": [
      {
         "Arn": "string",
         "As2Id": "string",
         "ProfileId": "string",
         "ProfileType": "string"
      }
   ]
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

NextToken

返回一令牌，您可以使用此令牌再次调用 ListProfiles，并接收其他结果（如有）。

类型：字符串

长度限制：最小长度为 1。长度上限为 6144。

Profiles

返回数组，其中的每个项目都包含主机密钥的详细信息。

类型：ListedProfile 对象数组

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidNextTokenException

传递的NextToken参数无效。

HTTP 状态代码：400

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK

- 适用于 Ruby V3 的 AWS SDK

ListSecurityPolicies

列出连接到您的服务器和 SFTP 连接器的安全策略。有关安全策略的更多信息，请参阅使用服务器的安全策略或使用 SFTP 连接器的安全策略。

请求语法

```
{
   "MaxResults": number,
   "NextToken": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

MaxResults

指定作为对 ListSecurityPolicies 查询的响应而返回的安全策略的数量。

类型：整数

有效范围：最小值为 1。最大值为 1000。

必需：否

NextToken

当您可以从ListSecurityPolicies调用中获得其他结果时，将在输出中返回一个NextToken参数。然后，您可以在后续命令中传递NextToken参数以继续列出其他工作流程。

类型：字符串

长度限制：长度下限为 1。长度上限为 6144。

必需：否

响应语法

```
{
```

```
    "NextToken": "string",
    "SecurityPolicyNames": [ "string" ]
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

NextToken

当您可以从ListSecurityPolicies调用中获得其他结果时，将在输出中返回一个NextToken参数。在以下命令中，您可以传入 NextToken 参数以继续列出其他服务器。

类型：字符串

长度限制：长度下限为 1。长度上限为 6144。

SecurityPolicyNames

列出的一系列的安全策略。

类型：字符串数组

长度约束：最小长度为 0。最大长度为 100。

模式：Transfer[A-Za-z0-9]*SecurityPolicy-[A-Za-z0-9-]+

错误

有关所有操作的常见错误信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family 服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidNextTokenException

传递的NextToken参数无效。

HTTP 状态代码：400

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 Trans AWS fer Family 服务不可用。

HTTP 状态代码：500

示例

示例

以下示例列出了所有可用安全策略名称。

示例请求

```
aws transfer list-security-policies
```

示例响应

```
{
    "SecurityPolicyNames": [
        "TransferSecurityPolicy-2023-05",
        "TransferSecurityPolicy-2022-03",
        "TransferSecurityPolicy-FIPS-2024-01",
        "TransferSecurityPolicy-2024-01",
        "TransferSecurityPolicy-PQ-SSH-FIPS-Experimental-2023-04",
        "TransferSecurityPolicy-PQ-SSH-Experimental-2023-04",
        "TransferSecurityPolicy-FIPS-2020-06",
        "TransferSecurityPolicy-2020-06",
        "TransferSecurityPolicy-2018-11",
        "TransferSecurityPolicy-FIPS-2023-05"
    ]
}
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- AWS 适用于 .NET 的 SDK
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS JavaScript V3 版软件开发工具包
- AWS 适用于 PHP 的 SDK V3
- AWS Python 软件开发工具包
- AWS 适用于 Ruby V3 的 SDK

ListServers

列出与您的 AWS 账户关联的文件传输协议服务器。

请求语法

```
{
   "MaxResults": number,
   "NextToken": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

MaxResults

指定作为对ListServers请求的响应而返回的用户数。

类型：整数

有效范围：最小值为 1。最大值为 1000。

必需：否

NextToken

当您可以从ListServers调用中获得其他结果时，将在输出中返回一个NextToken参数。然后，您可以在后续命令中传递NextToken参数以继续列出其他工作流程。

类型：字符串

长度限制：最小长度为 1。长度上限为 6144。

必需：否

响应语法

```
{
```

```
   "NextToken": "string",
   "Servers": [
      {
         "Arn": "string",
         "Domain": "string",
         "EndpointType": "string",
         "IdentityProviderType": "string",
         "LoggingRole": "string",
         "ServerId": "string",
         "State": "string",
         "UserCount": number
      }
   ]
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

NextToken

当您可以从ListServers调用中获得其他结果时，将在输出中返回一个NextToken参数。在以下命令中，您可以传入 NextToken 参数以继续列出其他服务器。

类型：字符串

长度限制：最小长度为 1。长度上限为 6144。

Servers

已列出一系列服务器。

类型：ListedServer 对象数组

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidNextTokenException

传递的NextToken参数无效。

HTTP 状态代码：400

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

示例

示例

以下示例列出了您的AWS 账户中存在的服务器。

请注意，示例 NextToken 值不是真实的：它们旨在指示如何使用参数。

示例请求

```
{
   "MaxResults": 1,
   "NextToken": "token-from-previous-API-call"
}
```

示例响应

```
{
   "NextToken": "another-token-to-continue-listing",
   "Servers": [
      {
         "Arn": "arn:aws:transfer:us-east-1:111112222222:server/s-01234567890abcdef",
         "Domain": "S3",
         "IdentityProviderType": "SERVICE_MANAGED",
```

```
            "EndpointType": "PUBLIC",
            "LoggingRole": "arn:aws:iam::111112222222:role/my-role",
            "ServerId": "s-01234567890abcdef",
            "State": "ONLINE",
            "UserCount": 3
        }
    ]
}
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ListTagsForResource

列出与您指定 Amazon 资源名称（ARN）关联的所有标签。该资源可能是用户、服务器或角色。

请求语法

```
{
   "Arn": "string",
   "MaxResults": number,
   "NextToken": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

Arn

请求与特定 Amazon 资源名称（ARN）关联的标签。ARN 是特定AWS资源（例如服务器、用户或角色）的标识符。

类型：字符串

长度约束：最小长度为 20。长度上限为 1600。

模式：`arn:\S+`

必需：是

MaxResults

指定作为对`ListTagsForResource`请求的响应而返回的标签数量。

类型：整数

有效范围：最小值为 1。最大值为 1000。

必需：否

NextToken

当您请求`ListTagsForResource`操作的更多结果时，将在输入中返回一个`NextToken`参数。然后，您可以向`NextToken`参数传入后续命令以继续列出其他标签。

类型：字符串

长度限制：最小长度为 1。长度上限为 6144。

必需：否

响应语法

```
{
   "Arn": "string",
   "NextToken": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

Arn

您指定用于列出其标签的 ARN。

类型：字符串

长度约束：最小长度为 20。长度上限为 1600。

模式：`arn:\S+`

NextToken

当您可以从`ListTagsForResource`调用中获得其他结果时，将在输出中返回一个`NextToken`参数。然后，您可以向`NextToken`参数传入后续命令以继续列出其他标签。

类型：字符串

长度限制：最小长度为 1。长度上限为 6144。

Tags

分配给资源的键值对，通常用于分组和搜索项目。标签是您定义的元数据。

类型：Tag 对象数组

数组成员：最少 1 个物品。最多 50 项。

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidNextTokenException

传递的NextToken参数无效。

HTTP 状态代码：400

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

示例

示例

以下示例列出了带有您指定的 ARN 的资源的标签。

示例请求

```
{
   "Arn": "arn:aws:transfer:us-east-1:176354371281:server/s-01234567890abcdef"
}
```

示例

此示例说明了的一种用法 ListTagsForResource。

示例响应

```
{
   "Tags": [
      {
         "Key": "Name",
         "Value": "MyServer"
      }
   ]
}
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ListUsers

列出您通过传递ServerId参数指定的启用文件传输协议的服务器的用户。

请求语法

```
{
   "MaxResults": number,
   "NextToken": "string",
   "ServerId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

MaxResults

指定作为对ListUsers请求的响应而返回的用户数。

类型：整数

有效范围：最小值为 1。最大值为 1000。

必需：否

NextToken

如果ListUsers调用还有其他结果，则会在输出中返回一个NextToken参数。然后，您可以将NextToken传递给后续ListUsers命令，以继续列出其他用户。

类型：字符串

长度限制：最小长度为 1。长度上限为 6144。

必需：否

ServerId

有分配到用户的服务器的系统分配的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`s-([0-9a-f]{17})`

必需：是

响应语法

```
{
   "NextToken": "string",
   "ServerId": "string",
   "Users": [
      {
         "Arn": "string",
         "HomeDirectory": "string",
         "HomeDirectoryType": "string",
         "Role": "string",
         "SshPublicKeyCount": number,
         "UserName": "string"
      }
   ]
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

NextToken

当您可以从`ListUsers`调用中获得其他结果时，将在输出中返回一个`NextToken`参数。然后，您可以向`NextToken`参数传入后续命令以继续列出其他用户。

类型：字符串

长度限制：最小长度为 1。长度上限为 6144。

ServerId

分配给用户的服务器的系统分配的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`s-([0-9a-f]{17})`

Users

根据您指定的 `ServerId` 值返回 Transfer Family 用户及其属性。

类型：ListedUser 对象数组

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidNextTokenException

传递的`NextToken`参数无效。

HTTP 状态代码：400

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

示例

示例

ListUsers API 调用会返回与您指定的服务器关联的用户列表。

示例请求

```
        {
         "MaxResults": 100,
         "NextToken": "eyJNYXJrZXIiOiBudWxsLCAiYm90b1X0cnVuU2F0ZV9hbW91bnQiOiAyfQ==",
         "ServerId": "s-01234567890abcdef"
         }
```

示例

这是此 API 调用的示例响应。

示例响应

```
{
   "NextToken": "eyJNYXJrZXIiOiBudWxsLCAiYm90b1X0cnVuU2F0ZV9hbW91bnQiOiAyfQ==",
   "ServerId": "s-01234567890abcdef",
   "Users": [
      {
         "Arn": "arn:aws:transfer:us-east-1:176354371281:user/s-01234567890abcdef/
charlie",
         "HomeDirectory": "/tests/home/charlie",
         "SshPublicKeyCount": 1,
         "Role": "arn:aws:iam::176354371281:role/transfer-role1",
         "Tags": [
            {
               "Key": "Name",
               "Value": "user1"
            }
         ],
         "UserName": "my_user"
      }
   ]
}
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ListWorkflows

列出您当前所在地区与您AWS 账户关联的所有工作流程。

请求语法

```
{
   "MaxResults": number,
   "NextToken": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

MaxResults

指定要返回的最大工作流程数。

类型：整数

有效范围：最小值为 1。最大值为 1000。

必需：否

NextToken

ListWorkflows在输出中返回NextToken参数。然后，您可以在后续命令中传递NextToken参数以继续列出其他工作流程。

类型：字符串

长度限制：最小长度为 1。长度上限为 6144。

必需：否

响应语法

```
{
   "NextToken": "string",
```

```
   "Workflows": [
      {
         "Arn": "string",
         "Description": "string",
         "WorkflowId": "string"
      }
   ]
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

NextToken

ListWorkflows在输出中返回NextToken参数。然后，您可以在后续命令中传递NextToken参数以继续列出其他工作流程。

类型：字符串

长度限制：最小长度为 1。长度上限为 6144。

Workflows

返回每个工作流程的Arn、WorkflowId和Description。

类型：ListedWorkflow 对象数组

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidNextTokenException

传递的NextToken参数无效。

HTTP 状态代码：400

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

SendWorkflowStepState

为异步自定义步骤发送回调。

ExecutionId、WorkflowId 和 Token 将在执行工作流程的自定义步骤期间传递到目标资源。您必须在它们的回调中包含这些信息，并提供状态。

请求语法

```
{
   "ExecutionId": "string",
   "Status": "string",
   "Token": "string",
   "WorkflowId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

ExecutionId

用于执行工作流程的唯一标识符。

类型：字符串

长度限制：固定长度为 36。

模式：[0-9a-fA-F]{8}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{12}

必需：是

Status

指示指定的步骤是成功还是失败。

类型：字符串

有效值：SUCCESS | FAILURE

必需：是

Token

用于区分同一执行中多个 Lambda 步骤的多个回调。

类型：字符串

长度限制：长度下限为 1。长度上限为 64。

模式：\w+

必需：是

WorkflowId

工作流的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：w-([a-z0-9]{17})

必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

AccessDeniedException

您没有足够的访问权限，无法执行该操作。

HTTP 状态代码：400

InternalServiceError

当 AWS Transfer Family 服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

ThrottlingException

由于请求限制而导致请求被拒绝。

HTTP 状态代码：400

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

StartFileTransfer

开始在本地AWS存储和远程 AS2 或 SFTP 服务器之间文件传输功能。

- 对于 AS2 连接器，您可以指定ConnectorId和一个或多个SendFilePaths来标识要传输的文件。
- 对于 SFTP 连接器，文件传输可以是出站的，也可以是入站的。在这两种情况下，您都要指定ConnectorId。根据传输方向，您还可以指定以下项目：
  - 如果您要将文件从合作伙伴的 SFTP 服务器传输到 Amazon Web Services 存储，则可以指定一个或多个RetreiveFilePaths来标识要传输的文件，并指定一个LocalDirectoryPath来指定目的地文件夹。
  - 如果要将文件从AWS存储传输到合作伙伴的 SFTP 服务器，则可以指定一个或多个SendFilePaths来标识要传输的文件，并指定一个RemoteDirectoryPath来指定目的地文件夹。

请求语法

```
{
   "ConnectorId": "string",
   "LocalDirectoryPath": "string",
   "RemoteDirectoryPath": "string",
   "RetrieveFilePaths": [ "string" ],
   "SendFilePaths": [ "string" ]
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

ConnectorId

连接器的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：c-([0-9a-f]{17})

必需：是

LocalDirectoryPath

对于入站传输，LocaDirectoryPath 指定从合作伙伴的 SFTP 服务器传输的一个或多个文件的目的地。

类型：字符串

长度限制：最小长度为 1。长度上限为 1024。

模式：(.)+

必需：否

RemoteDirectoryPath

对于出站传输，RemoteDirectoryPath 指定传输到合作伙伴的 SFTP 服务器的一个或多个文件的目的地。如果未指定RemoteDirectoryPath，则传输文件的目的地是 SFTP 用户的主目录。

类型：字符串

长度限制：最小长度为 1。长度上限为 1024。

模式：(.)+

必需：否

RetrieveFilePaths

合作伙伴的 SFTP 服务器的一个或多个源路径。每个字符串代表一次入站文件传输的源文件路径。

类型：字符串数组

数组成员：最少 1 个物品。最多 10 项。

长度限制：长度下限为 1。长度上限为 1024。

模式：(.)+

必需：否

SendFilePaths

Amazon S3 存储的一个或多个源路径。每个字符串代表一次出站文件传输的源文件路径。例如，*DOC-EXAMPLE-BUCKET/myfile.txt* 。

> **Note**
>
> 用您的实际存储桶替换 *DOC-EXAMPLE-BUCKET* 。

类型：字符串数组

数组成员：最少 1 个物品。最多 10 项。

长度限制：长度下限为 1。长度上限为 1024。

模式：(.)+

必需：否

响应语法

```
{
   "TransferId": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

TransferId

返回文件传输的唯一标识符。

类型：字符串

长度限制：最小长度为 1。最大长度为 512。

模式：[0-9a-zA-Z./-]+

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

ThrottlingException

由于请求限制而导致请求被拒绝。

HTTP 状态代码：400

示例

示例

以下示例启动从 Transfer Family 服务器到远程交易伙伴端点的 AS2 文件传输功能。用您的实际存储桶替换 *DOC-EXAMPLE-BUCKET* 。

示例请求

```
{
    "ConnectorId": "c-AAAA1111BBBB2222C",
    "SendFilePaths": [
        "/DOC-EXAMPLE-BUCKET/myfile-1.txt",
        "/DOC-EXAMPLE-BUCKET/myfile-2.txt",
        "/DOC-EXAMPLE-BUCKET/myfile-3.txt"
```

```
    ]
}
```

示例响应

```
{
   "TransferId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

示例

以下示例启动从本地AWS存储到远程 SFTP 服务器的文件传输功能。

示例请求

```
{
    "ConnectorId": "c-01234567890abcdef",
    "SendFilePaths": [
        "/DOC-EXAMPLE-BUCKET/myfile-1.txt",
        "/DOC-EXAMPLE-BUCKET/myfile-2.txt",
        "/DOC-EXAMPLE-BUCKET/myfile-3.txt"
    ],
    "RemoteDirectoryPath": "/MySFTPRootFolder/fromTranferFamilyServer"
}
```

示例响应

```
{
   "TransferId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

示例

以下示例启动从远程 SFTP 服务器到本地AWS存储的文件传输功能。

示例请求

```
{
    "ConnectorId": "c-111122223333AAAAA",
    "RetrieveFilePaths": [
        "/MySFTPFolder/toTranferFamily/myfile-1.txt",
```

```
        "/MySFTPFolder/toTranferFamily/myfile-2.txt",
        "/MySFTPFolder/toTranferFamily/myfile-3.txt"
    ],
    "LocalDirectoryPath": "/DOC-EXAMPLE-BUCKET/mySourceFiles"
}
```

示例响应

```
{
   "TransferId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa"
}
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

StartServer

将启用文件传输协议的服务器的状态从 OFFLINE 更改为 ONLINE。它对已经是 ONLINE 的服务器没有影响。ONLINE 服务器可以接受和处理文件传输作业。

状态 STARTING 表示服务器处于中间状态，要么无法完全响应，要么未完全联机。START_FAILED 的值可以表示错误情况。

此次调用未返回任何响应。

请求语法

```
{
   "ServerId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

ServerId

已启动服务器的系统分配唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

ThrottlingException

由于请求限制而导致请求被拒绝。

HTTP 状态代码：400

示例

示例

以下示例启动服务器。

示例请求

```
{
   "ServerId": "s-01234567890abcdef"
}
```

示例

这是此 API 调用的示例响应。

示例响应

```
{
   "ServerId": "s-01234567890abcdef"
}
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

StopServer

将启用文件传输协议的服务器的状态从 ONLINE 更改为 OFFLINE。OFFLINE 服务器无法接受和处理文件传输任务。与服务器相关的信息（例如服务器和用户属性）不会因停止服务器而受到影响。

> **Note**
>
> 停止服务器不会减少或影响您的文件传输协议端点账单；您必须删除服务器才能停止计费。

状态 STOPPING 表示服务器处于中间状态，要么无法完全响应，要么未完全脱机。STOP_FAILED 的值可以表示错误情况。

此次调用未返回任何响应。

请求语法

```
{
   "ServerId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

ServerId

已停止服务器的系统分配唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

ThrottlingException

由于请求限制而导致请求被拒绝。

HTTP 状态代码：400

示例

示例

以下示例停止服务器。

示例请求

```
{
   "ServerId": "s-01234567890abcdef"
}
```

示例

这是此 API 调用的示例响应。

示例响应

```
{
   "ServerId": "s-01234567890abcdef"
}
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

TagResource

将通过 Amazon 资源名称 (ARN) 标识的键/值对附加到资源。资源是用户、服务器、角色和其他实体。

此次调用未返回任何回应。

请求语法

```
{
   "Arn": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

Arn

特定 AWS 资源（如服务器、用户或角色）的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最小长度为 20。长度上限为 1600。

模式：`arn:\S+`

必需：是

Tags

分配给 ARN 的键值对，可用于按类型对资源进行分组和搜索。您可以出于任何目的将此元数据附加到资源（服务器、用户、工作流程等）。

类型：Tag 对象数组

数组成员：最少 1 个物品。最多 50 项。

必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

示例

示例

以下示例向启用文件传输协议的服务器添加标签。

示例请求

```
{
   "Arn": "arn:aws:transfer:us-east-1:176354371281:server/s-01234567890abcdef",
   "Tags": [
      {
         "Key": "Group",
         "Value": "Europe"
      }
   ]
}
```

示例

此示例说明了的一种用法 TagResource。

示例响应

```
         HTTP 200 response with an empty HTTP body.
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

TestConnection

测试您的 SFTP 连接器是否设置成功。我们强烈建议您调用此操作来测试您在本地 AWS 存储和贸易伙伴的 SFTP 服务器之间传输文件的能力。

请求语法

```
{
   "ConnectorId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

ConnectorId

连接器的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`c-([0-9a-f]{17})`

必需：是

响应语法

```
{
   "ConnectorId": "string",
   "Status": "string",
   "StatusMessage": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

ConnectorId

返回您正在测试的连接器对象的标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`c-([0-9a-f]{17})`

Status

如果测试成功，则会返回 OK，如果测试失败，则会返回 ERROR。

类型：字符串

StatusMessage

如果测试成功，则返回 `Connection succeeded`。或者，如果测试失败，则返回描述性错误消息。以下列表根据您收到的错误消息提供了疑难解答的详细信息。

- 确认您的密钥名称与传输角色权限中的密钥名称一致。
- 验证连接器配置中的服务器 URL，并验证登录凭据在连接器之外是否成功运行。
- 验证密钥是否存在且格式正确。
- 验证连接器配置中的可信主机密钥是否与 `ssh-keyscan` 输出相匹配。

类型：字符串

错误

有关所有操作返回的常见错误的信息，请参阅 常见错误。

InternalServiceError

当 AWS Transfer Family 服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 Transfer Family 服务找不到资源时，就会 AWS引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 Trans AWS fer Family 服务不可用。

HTTP 状态代码：500

示例

示例

以下示例测试与远程服务器的连接。

```
aws transfer test-connection --connector-id c-abcd1234567890fff
```

示例响应

如果成功，API 调用将返回以下详细信息。

```
{
    "Status": "OK",
    "StatusMessage": "Connection succeeded"
}
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- AWS 适用于 .NET 的 SDK
- AWS 适用于 C++ 的 SDK

- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS JavaScript V3 版软件开发工具包
- AWS 适用于 PHP 的 SDK V3
- AWS Python 软件开发工具包
- AWS 适用于 Ruby V3 的 SDK

TestIdentityProvider

如果启用文件传输协议的服务器的 IdentityProviderType 为 AWS_DIRECTORY_SERVICE 或 API_Gateway，则测试您的身份提供程序是否已成功设置。我们强烈建议您在创建服务器后立即调用此操作来测试您的身份验证方法。这样，您就可以对身份提供程序集成问题进行故障排除，以确保您的用户可以成功使用该服务。

ServerId 和 UserName 参数是必需的。ServerProtocol、SourceIp 和 UserPassword 是可选的。

请注意以下几点：

- 如果您的服务器的 IdentityProviderType 是 SERVICE_MANAGED，则无法使用 TestIdentityProvider。
- TestIdentityProvider 不适用于密钥：它只接受密码。
- TestIdentityProvider 可以测试处理密钥和密码的自定义身份提供程序的密码操作。
- 如果您为任何参数提供的值不正确，则 Response 字段为空。
- 如果您为使用服务托管用户的服务器提供服务器 ID，则会出现错误：

  An error occurred (InvalidRequestException) when calling the TestIdentityProvider operation: s-*server-ID* not configured for external auth
- 如果您为 --server-id 参数输入的服务器 ID 不能识别实际的传输服务器，则会收到以下错误：

  An error occurred (ResourceNotFoundException) when calling the TestIdentityProvider operation: Unknown server.

  您的服务器可能位于不同的区域。您可以通过添加以下内容来指定区域：--region region-code，例如使用 --region us-east-2 来指定在美国东部（俄亥俄州）的服务器。

请求语法

```
{
   "ServerId": "string",
   "ServerProtocol": "string",
   "SourceIp": "string",
   "UserName": "string",
   "UserPassword": "string"
```

```
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

ServerId

特定服务器的系统分配的标识符。服务器的用户名和密码测试服务器的用户身份验证方法。

类型：字符串

长度限制：固定长度为 19。

模式：`s-([0-9a-f]{17})`

必需：是

ServerProtocol

要测试的文件传输协议的类型。

可用的协议包括：

- Secure Shell (SSH) File Transfer Protocol (SFTP)
- 安全文件传输协议 (FTPS)
- 文件传输协议 (FTP)
- 适用性声明 2 (AS2)

类型：字符串

有效值：`SFTP | FTP | FTPS | AS2`

必需：否

SourceIp

要测试账户的源 IP 地址。

类型：字符串

长度约束：最小长度为 0。最大长度为 32。

模式：\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}

必需：否

UserName

要测试账户名称。

类型：字符串

长度约束：最小长度为 3。最大长度为 100。

模式：[\w][\w@.-]{2,99}

必需：是

UserPassword

要测试的账户的密码。

类型：字符串

长度约束：最小长度为 0。最大长度为 1024。

必需：否

响应语法

```
{
   "Message": "string",
   "Response": "string",
   "StatusCode": number,
   "Url": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

Message

一条表明测试是否成功的消息。

> **Note**
> 如果返回空字符串，则最有可能的原因是由于用户名或密码不正确导致身份验证失败。

类型：字符串

Response

从您的 API Gateway 或 Lambda 函数返回的响应。

类型：字符串

StatusCode

HTTP 状态代码，即来自您的 API Gateway 或 Lambda 函数的响应。

类型：整数

Url

提供用于验证用户身份的服务端点。

类型：字符串

长度约束：最小长度为 0。最大长度为 255。

错误

有关所有操作返回的常见错误的信息，请参阅 常见错误。

InternalServiceError

当 AWS Transfer Family 服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 Transfer Family 服务找不到资源时，就会 AWS引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 Trans AWS fer Family 服务不可用。

HTTP 状态代码：500

示例

示例

以下请求返回来自身份提供商的消息，说明用户名和密码组合是可以使用的有效身份 AWS Transfer Family。

示例请求

```
{
   "ServerID": "s-01234567890abcdef",
   "UserName": "my_user",
   "UserPassword": "MyPassword-1"
}
```

示例

以下响应显示成功测试的示例响应。

示例响应

```
      "Response":"
      {\"homeDirectory\":\"/mybucket001\",\"homeDirectoryDetails\":null,
\"homeDirectoryType\":\"PATH\",\"posixProfile\":null,
      \"publicKeys\":\"[ssh-rsa-key]\",\"role\":\"arn:aws:iam::123456789012:role/
my_role\",\"policy\":null,\"username\":\"transferuser002\",
```

```
      \"identityProviderType\":null,\"userConfigMessage\":null)"}
      "StatusCode": "200",
      "Message": ""
```

示例

以下响应表明指定的用户属于多个具有访问权限的群组。

```
        "Response":"",
        "StatusCode":200,
        "Message":"More than one associated access found for user's groups."
```

示例

如果您使用 API Gateway 创建并配置了自定义身份提供程序，则可以输入以下命令来测试您的用户：

`aws transfer test-identity-provider --server-id` *`s-0123456789abcdefg`* `--user-name` *`myuser`*

其中 s-0123456789abcdefg 是您的传输服务器，myuser 是您的自定义用户的用户名。

如果命令成功，则响应的形式与下方类似，其中：

- AWS 账户 身份证是 0 12345678901
- 用户角色是 user-role-api-gateway
- 主目录是 myuser-bucket
- 公钥是 public-key
- 调用网址为 invocation-URL

```
{
    "Response": "{\"Role\": \"arn:aws:iam::012345678901:role/user-role-api-gateway\",
\"HomeDirectory\": \"/myuser-bucket\",\"PublicKeys\": \"[public-key]\"}",
    "StatusCode": 200,
    "Message": "",
    "Url": "https://invocation-URL/servers/s-0123456789abcdefg/users/myuser/config"
}
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- AWS 适用于 .NET 的 SDK
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS JavaScript V3 版软件开发工具包
- AWS 适用于 PHP 的 SDK V3
- AWS Python 软件开发工具包
- AWS 适用于 Ruby V3 的 SDK

UntagResource

从资源中分离一个键值对，如其 Amazon 资源名称（ARN）所标识的。资源是用户、服务器、角色和其他实体。

此次调用未返回任何响应。

请求语法

```
{
   "Arn": "string",
   "TagKeys": [ "string" ]
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

Arn

要移除标签的资源的值。Amazon 资源名称 (ARN) 是特定 AWS 资源（例如服务器、用户或角色）的标识符。

类型：字符串

长度约束：最小长度为 20。长度上限为 1600。

模式：`arn:\S+`

必需：是

TagKeys

TagKeys 是分配给 ARN 的键值对，可用于按类型对资源进行分组和搜索。可以出于任何目的将此元数据附加到资源中。

类型：字符串数组

数组成员：最少 1 个物品。最多 50 项。

长度限制：长度下限为 0。长度上限为 128。

必需：是

响应元素

如果此操作成功，则该服务会发送回带有空 HTTP 正文的 HTTP 200 响应。

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

示例

示例

以下示例删除了启用文件传输协议的服务器的标签。

示例请求

```
{
   "Arn": "arn:aws:transfer:us-east-1:176354371281:server/s-01234567890abcdef",
   "TagKeys": "Europe" ]
}
```

示例

此示例说明了的一种用法 UntagResource。

示例响应

```
        HTTP 200 response with an empty HTTP body.
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

UpdateAccess

允许您更新ServerID和ExternalID参数中指定的访问权限的参数。

请求语法

```
{
   "ExternalId": "string",
   "HomeDirectory": "string",
   "HomeDirectoryMappings": [
      {
         "Entry": "string",
         "Target": "string",
         "Type": "string"
      }
   ],
   "HomeDirectoryType": "string",
   "Policy": "string",
   "PosixProfile": {
      "Gid": number,
      "SecondaryGids": [ number ],
      "Uid": number
   },
   "Role": "string",
   "ServerId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

ExternalId

识别目录中特定组所需的唯一标识符。您关联的组中的用户可以通过使用AWS Transfer Family的启用协议访问您的 Amazon S3 或 Amazon EFS 资源。如果您知道组名，则可以使用 Windows 运行以下命令来查看 SID 值 PowerShell。

```
Get-ADGroup -Filter {samAccountName -like "YourGroupName*"} -Properties
* | Select SamAccountName,ObjectSid
```

在该命令中，替换为您的 YourGroupNameActive Directory 组的名称。

用于验证此参数的正则表达式是由不带空格的大写和小写字母数字字符组成的字符串。此外，还可以包括下划线或以下任何字符：=、.@：/-

类型：字符串

长度限制：最小长度为 1。最大长度为 256。

模式：S-1-[\d-]+

必需：是

HomeDirectory

用户使用客户端登录服务器时的登录目录（文件夹）。

HomeDirectory 示例为 /bucket_name/home/mydirectory。

> **Note**
>
> HomeDirectory 参数仅在 HomeDirectoryType 设置为 PATH 时使用。

类型：字符串

长度限制：长度下限为 0。最大长度为 1024。

模式：(|/.*)

必需：否

HomeDirectoryMappings

逻辑目录映射指定哪些 Amazon S3 或 Amazon EFS 路径和密钥应对您的用户可见，以及使其对用户可见的方式。您需要指定Entry和Target对，其中 Entry 显示如何使路径可见，Target 是实际的 Amazon S3 或 Amazon EFS 路径。如果您只指定一个目标，则将按原样显示。您还必须确保您的AWS Identity and Access Management（IAM）角色提供对Target中路径的访问权限。只有当HomeDirectoryType设置为LOGICAL时，才能设置此值。

以下是 Entry 和 Target 对示例。

```
[ { "Entry": "/directory1", "Target": "/bucket_name/home/
mydirectory" } ]
```

在大多数情况下，您可以使用此值而不是会话策略将您的用户锁定到指定的主目录（“chroot”）。为此，您可以将 Entry 设置为 / 并将 Target 设置为 HomeDirectory 参数值。

以下是chroot的 Entry 和 Target 对示例。

```
[ { "Entry": "/", "Target": "/bucket_name/home/mydirectory" } ]
```

类型：HomeDirectoryMapEntry 对象数组

数组成员：最少 1 个物品。最大数量为 50000 个。

必需：否

HomeDirectoryType

您希望用户在登录服务器时，用户主目录的登录目录（文件夹）的类型。如果您将其设置为PATH，则用户将在其文件传输协议客户端中原样看到 Amazon S3 存储桶或 Amazon EFS 的绝对路径。如果您将其设置为LOGICAL，则需要针对您希望如何使 Amazon S3 或 Amazon EFS 路径对用户可见，在HomeDirectoryMappings中提供映射。

> **Note**
>
> 如果HomeDirectoryType是LOGICAL，则必须使用HomeDirectoryMappings参数提供映射。另一方面，如果HomeDirectoryType是PATH，则使用HomeDirectory参数提供绝对路径。您的模板中不能同时使用HomeDirectory和HomeDirectoryMappings。

类型：字符串

有效值：PATH | LOGICAL

必需：否

Policy

适用于您的用户的会话策略，可让您跨多个用户使用相同的 AWS Identity and Access Management(IAM) 角色。此策略将用户的访问权限缩小至 Amazon S3 存储桶的一部分。可在此策略中使用的变量包括 ${Transfer:UserName}、${Transfer:HomeDirectory} 和 ${Transfer:HomeBucket}。

> **Note**
>
> 仅当ServerId域为 Amazon S3 时，此政策才适用。Amazon EFS 不使用会话策略。

对于会话测量，AWS Transfer Family 将策略存储为 JSON blob，而不是策略的 Amazon 资源名称（ARN）。您将策略保存为 JSON blob 并将其传递给 `Policy` 参数。
有关会话策略的示例，请参阅示例会话策略。
有关更多信息，请参阅AssumeRole《AWS安全令牌服务 API 参考》。

类型：字符串

长度约束：最小长度为 0。最大长度为 2048。

必需：否

PosixProfile

完整的 POSIX 身份，包括用户 ID (`Uid`)、组 ID (`Gid`) 和任何辅助组 ID (`SecondaryGids`)，用于控制用户对 Amazon EFS 文件系统的访问。POSIX 权限针对文件系统中的文件和目录设置，用于确定用户在将文件传入和传出 Amazon EFS 文件系统时获得的访问权限级别。

类型：PosixProfile 对象

必需：否

Role

控制用户对 Amazon S3 桶或 Amazon EFS 文件系统访问权限的 AWS Identity and Access Management（IAM）角色的 Amazon 资源名称（ARN）。附加到此角色的策略确定在将文件传入和传出 Amazon S3 桶或 Amazon EFS 文件系统时要为用户提供的访问权限级别。IAM 角色还应包含一个信任关系，从而允许服务器在为用户的传输请求提供服务时访问您的资源。

类型：字符串

长度约束：最小长度为 20。最大长度为 2048。

模式：`arn:.*role/\S+`

必需：否

ServerId

服务器实例的系统分配的唯一标识符。这是将您的用户添加到的特定服务器。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

必需：是

响应语法

```
{
   "ExternalId": "string",
   "ServerId": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

ExternalId

用户所在组的外部标识符可以通过使用AWS Transfer Family 的启用协议访问您的 Amazon S3 或 Amazon EFS 资源。

类型：字符串

长度限制：最小长度为 1。最大长度为 256。

模式：S-1-[\d-]+

ServerId

用户附加到的服务器的标识符。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceExistsException

请求的资源不存在，或者存在于为命令指定的区域以外的区域。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

ThrottlingException

由于请求限制而导致请求被拒绝。

HTTP 状态代码：400

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK

- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版 SDK
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

UpdateAgreement

更新现有协议的某些参数。为要更新的协议提供 AgreementId 和 ServerId，以及要更新的参数的新值。

请求语法

```
{
   "AccessRole": "string",
   "AgreementId": "string",
   "BaseDirectory": "string",
   "Description": "string",
   "LocalProfileId": "string",
   "PartnerProfileId": "string",
   "ServerId": "string",
   "Status": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

AccessRole

连接器用于使用 AS2 或 SFTP 协议发送文件。对于访问权限角色，请提供要使用的AWS Identity and Access Management角色的 Amazon 资源名称 (ARN)。

对于 AS2 连接器

借助 AS2，您可以通过调用 StartFileTransfer 并在请求参数中指定文件路径 SendFilePaths 来发送文件。我们使用文件的父目录（例如 --send-file-paths /bucket/dir/file.txt，父目录是 /bucket/dir/）来临时存储经过处理的 AS2 消息文件，存储 MDN（当从合作伙伴那里收到时），以及写入包含传输相关元数据的最终 JSON 文件。因此，AccessRole 需要提供对 StartFileTransfer 请求中所使用文件位置父目录的读取和写入权限。此外，您还需要提供对您想要使用 StartFileTransfer 发送的文件父目录的读取和写入权限。

如果您对 AS2 连接器执行基本身份验证，则访问角色需要密钥secretsmanager:GetSecretValue权限。如果 Secrets Manager 中的秘钥加密方式使用客户托管秘钥，而非AWS托管秘钥，则该角色需要此秘钥kms:Decrypt的权限。

对于 SFTP 连接器

因此，确保提供对 StartFileTransfer 请求中所使用文件位置父目录的读取和写入权限。此外，请确保该角色向secretsmanager:GetSecretValue提供AWS Secrets Manager权限。

类型：字符串

长度约束：最小长度为 20。最大长度为 2048。

模式：arn:.*role/\S+

必需：否

AgreementId

的唯一标识符。创建协议时会返回此标识符。

类型：字符串

长度限制：固定长度为 19。

模式：a-([0-9a-f]{17})

必需：是

BaseDirectory

要更改传输的文件的登录目录（文件夹），请提供您要使用的存储桶文件夹，例如 */DOC-EXAMPLE-BUCKET/home/mydirectory* 。

类型：字符串

长度限制：长度下限为 0。最大长度为 1024。

模式：(|/.*)

必需：否

Description

要替换现有描述，请提供协议的简短描述。

类型：字符串

长度限制：最小长度为 1。最大长度为 200。

模式：[\p{Graph}]+

必需：否

LocalProfileId

AS2 本地配置文件的唯一标识符。

要更改本地配置文件标识符，请在此处提供一个新值。

类型：字符串

长度限制：固定长度为 19。

模式：p-([0-9a-f]{17})

必需：否

PartnerProfileId

合作伙伴配置文件的唯一标识符。要更改合作伙伴配置文件标识符，请在此处提供一个新值。

类型：字符串

长度限制：固定长度为 19。

模式：p-([0-9a-f]{17})

必需：否

ServerId

服务器实例的系统分配的唯一标识符。此标识符表示协议使用的特定服务器。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

必需：是

Status

您可以更新协议的状态，可以激活无效的协议，反之亦然。

类型：字符串

有效值：ACTIVE | INACTIVE

必需：否

响应语法

```
{
   "AgreementId": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

AgreementId

协议的唯一标识符。创建协议时会返回此标识符。

类型：字符串

长度限制：固定长度为 19。

模式：a-([0-9a-f]{17})

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceExistsException

请求的资源不存在，或者存在于为命令指定的区域以外的区域。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

ThrottlingException

由于请求限制而导致请求被拒绝。

HTTP 状态代码：400

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

UpdateCertificate

更新证书的活动日期和非活动日期。

请求语法

```
{
   "ActiveDate": number,
   "CertificateId": "string",
   "Description": "string",
   "InactiveDate": number
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

ActiveDate

指定证书何时生效的可选日期。

类型：时间戳

必需：否

CertificateId

您要更新的证书对象的标识符。

类型：字符串

长度限制：固定长度为 22。

模式：`cert-([0-9a-f]{17})`

必需：是

Description

帮助识别证书的简短描述。

类型：字符串

长度限制：最小长度为 1。最大长度为 200。

模式：`[\p{Graph}]+`

必需：否

InactiveDate

指定证书何时失效的可选日期。

类型：时间戳

必需：否

响应语法

```
{
   "CertificateId": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

CertificateId

返回您要更新的证书对象的标识符。

类型：字符串

长度限制：固定长度为 22。

模式：`cert-([0-9a-f]{17})`

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

ThrottlingException

由于请求限制而导致请求被拒绝。

HTTP 状态代码：400

示例

示例

以下示例更新了证书的有效日期，将有效日期设置为 2022 年 1 月 16 日 16:12:07 UTC — 5 小时。

示例请求

```
aws transfer update-certificate --certificate-id c-abcdefg123456hijk --active-date
 2022-01-16T16:12:07-05:00
```

示例

以下是该 API 调用的示例响应。

示例响应

```
"CertificateId": "c-abcdefg123456hijk"
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

UpdateConnector

更新现有连接器的某些参数。为要更新的协议提供ConnectorId，以及要更新的参数的新值。

请求语法

```
{
   "AccessRole": "string",
   "As2Config": {
      "BasicAuthSecretId": "string",
      "Compression": "string",
      "EncryptionAlgorithm": "string",
      "LocalProfileId": "string",
      "MdnResponse": "string",
      "MdnSigningAlgorithm": "string",
      "MessageSubject": "string",
      "PartnerProfileId": "string",
      "SigningAlgorithm": "string"
   },
   "ConnectorId": "string",
   "LoggingRole": "string",
   "SecurityPolicyName": "string",
   "SftpConfig": {
      "TrustedHostKeys": [ "string" ],
      "UserSecretId": "string"
   },
   "Url": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

AccessRole

连接器用于使用 AS2 或 SFTP 协议发送文件。对于访问角色，请提供要使用的 AWS Identity and Access Management 角色的 Amazon 资源名称 (ARN)。

对于 AS2 连接器

借助 AS2，您可以通过调用 StartFileTransfer 并在请求参数中指定文件路径 SendFilePaths 来发送文件。我们使用文件的父目录（例如 --send-file-paths /bucket/dir/file.txt，父目录是 /bucket/dir/）来临时存储经过处理的 AS2 消息文件，存储 MDN（当从合作伙伴那里收到时），以及写入包含传输相关元数据的最终 JSON 文件。因此，AccessRole 需要提供对 StartFileTransfer 请求中所使用文件位置父目录的读取和写入权限。此外，您还需要提供对您想要使用 StartFileTransfer 发送的文件父目录的读取和写入权限。

如果您对 AS2 连接器执行基本身份验证，则访问角色需要密钥secretsmanager:GetSecretValue权限。如果使用客户管理的密钥而不是 Secrets Manager 中的 AWS 托管密钥对密钥进行加密，则该角色还需要该密钥的kms:Decrypt权限。

对于 SFTP 连接器

因此, 确保提供对 StartFileTransfer 请求中所使用文件位置父目录的读取和写入权限。此外，请确保该角色向提供secretsmanager:GetSecretValue权限 AWS Secrets Manager。

类型：字符串

长度约束：最小长度为 20。最大长度为 2048。

模式：arn:.*role/\S+

必需：否

As2Config

包含 AS2 连接器对象参数的结构。

类型：As2ConnectorConfig 对象

必需：否

ConnectorId

连接器的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：c-([0-9a-f]{17})

必需：是

LoggingRole

(IAM) 角色的亚马逊资源名称 AWS Identity and Access Management (ARN)，它允许连接器开启对 Amazon S3 CloudWatch 事件的日志记录。设置后，您可以在 CloudWatch 日志中查看连接器活动。

类型：字符串

长度约束：最小长度为 20。最大长度为 2048。

模式：`arn:.*role/\S+`

必需：否

SecurityPolicyName

为连接器指定安全策略的名称。

类型：字符串

长度限制：长度下限为 0。最大长度为 100。

模式：`TransferSFTPConnectorSecurityPolicy-[A-Za-z0-9-]+`

必需：否

SftpConfig

包含 SFTP 连接器对象参数的结构。

类型：SftpConnectorConfig 对象

必需：否

Url

合作伙伴的 AS2 或 SFTP 端点的 URL。

类型：字符串

长度约束：最小长度为 0。最大长度为 255。

必需：否

响应语法

```
{
   "ConnectorId": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

ConnectorId

返回您正在更新的连接器对象的标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`c-([0-9a-f]{17})`

错误

有关所有操作的常见错误信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family 服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceExistsException

请求的资源不存在，或者存在于为命令指定的区域以外的区域。

HTTP 状态代码：400

ResourceNotFoundException

当 Transfer Family 服务找不到资源时，就会 AWS引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 Trans AWS fer Family 服务不可用。

HTTP 状态代码：500

ThrottlingException

由于请求限制而导致请求被拒绝。

HTTP 状态代码：400

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- AWS 适用于 .NET 的 SDK
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS JavaScript V3 版软件开发工具包
- AWS 适用于 PHP 的 SDK V3
- AWS Python 软件开发工具包
- AWS 适用于 Ruby V3 的 SDK

UpdateHostKey

更新由ServerId和HostKeyId参数指定的主机密钥的描述。

请求语法

```
{
   "Description": "string",
   "HostKeyId": "string",
   "ServerId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

Description

更新了主机密钥的描述。

类型：字符串

长度限制：长度下限为 0。最大长度为 200。

模式：[\p{Print}]*

必需：是

HostKeyId

正在更新的主机密钥的标识符。

类型：字符串

长度限制：固定长度为 25。

模式：hostkey-[0-9a-f]{17}

必需：是

ServerId

包含正在更新的主机密钥的服务器的标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`s-([0-9a-f]{17})`

必需：是

响应语法

```
{
   "HostKeyId": "string",
   "ServerId": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

HostKeyId

返回更新的主机密钥的主机密钥标识符。

类型：字符串

长度限制：固定长度为 25。

模式：`hostkey-[0-9a-f]{17}`

ServerId

返回包含更新的主机密钥的服务器的服务器标识符。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

ThrottlingException

由于请求限制而导致请求被拒绝。

HTTP 状态代码：400

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK

- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

UpdateProfile

更新现有配置文件的某些参数。为要更新的配置文件提供ProfileId，以及要更新的参数的新值。

请求语法

```
{
   "CertificateIds": [ "string" ],
   "ProfileId": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

CertificateIds

已导入证书的标识符数组。您可以使用此标识符来处理配置文件和合作伙伴配置文件。

类型：字符串数组

长度限制：固定长度为 22。

模式：cert-([0-9a-f]{17})

必需：否

ProfileId

您要更新的配置文件对象的标识符。

类型：字符串

长度限制：固定长度为 19。

模式：p-([0-9a-f]{17})

必需：是

响应语法

```
{
```

```
   "ProfileId": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

ProfileId

返回正在更新的配置文件的标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`p-([0-9a-f]{17})`

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

ThrottlingException

由于请求限制而导致请求被拒绝。

HTTP 状态代码：400

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

UpdateServer

创建启用文件传输协议的服务器后，更新该服务器的属性。

UpdateServer调用会返回您更新的服务器的ServerId。

请求语法

```
{
   "Certificate": "string",
   "EndpointDetails": {
      "AddressAllocationIds": [ "string" ],
      "SecurityGroupIds": [ "string" ],
      "SubnetIds": [ "string" ],
      "VpcEndpointId": "string",
      "VpcId": "string"
   },
   "EndpointType": "string",
   "HostKey": "string",
   "IdentityProviderDetails": {
      "DirectoryId": "string",
      "Function": "string",
      "InvocationRole": "string",
      "SftpAuthenticationMethods": "string",
      "Url": "string"
   },
   "LoggingRole": "string",
   "PostAuthenticationLoginBanner": "string",
   "PreAuthenticationLoginBanner": "string",
   "ProtocolDetails": {
      "As2Transports": [ "string" ],
      "PassiveIp": "string",
      "SetStatOption": "string",
      "TlsSessionResumptionMode": "string"
   },
   "Protocols": [ "string" ],
   "S3StorageOptions": {
      "DirectoryListingOptimization": "string"
   },
   "SecurityPolicyName": "string",
   "ServerId": "string",
   "StructuredLogDestinations": [ "string" ],
   "WorkflowDetails": {
```

```
      "OnPartialUpload": [
         {
            "ExecutionRole": "string",
            "WorkflowId": "string"
         }
      ],
      "OnUpload": [
         {
            "ExecutionRole": "string",
            "WorkflowId": "string"
         }
      ]
   }
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

Certificate

Certifice Manager (ACM) AWS证书的亚马逊资源名称 (ARN)。在 `Protocols` 设置为 `FTPS` 时是必需的。

要请求新的公有证书，请参阅 AWS Certificate Manager 用户指南中的请求公有证书。

要将现有证书导入到 ACM，请参阅 AWS Certificate Manager 用户指南中的将证书导入到 ACM。

要请求私有证书以通过私有 IP 地址使用 FTPS，请参阅 AWS Certificate Manager 用户指南中的请求私有证书。

支持具有以下加密算法和密钥大小的证书：

- 2048 位 RSA (RSA_2048)
- 4096 位 RSA (RSA_4096)
- Elliptic Prime Curve 256 位 (EC_prime256v1)
- Elliptic Prime Curve 384 位 (EC_secp384r1)
- Elliptic Prime Curve 521 位 (EC_secp521r1)

> **Note**
>
> 证书必须是指定了 FQDN 或 IP 地址且具有有关颁发者的信息的有效 SSL/TLS X.509 版本 3 证书。

类型：字符串

长度约束：最小长度为 0。长度上限为 1600。

必需：否

EndpointDetails

要为服务器配置的 Virtual Private Cloud (VPC) 端点设置。当您在 VPC 中托管端点时，您可以使端点仅可供 VPC 内的资源访问，也可以附加弹性 IP 地址并使端点可由客户端通过 Internet 访问。您 VPC 的默认安全组会自动分配到您的端点。

类型：EndpointDetails 对象

必需：否

EndpointType

您希望服务器使用的端点类型。您可以选择使服务器的端点可公开访问（PUBLIC），或将其托管在 VPC 内。如果端点在 VPC 中托管，您可以仅在 VPC 内限制对服务器和资源的访问，或者通过将弹性 IP 地址直接附加到其上，使其面向 Internet。

> **Note**
>
> 2021 年 5 月 19 日之后，如果您的 AWS账户EndpointType=VPC_ENDPOINT在 2021 年 5 月 19 日之前尚未使用您的账户创建服务器，则您将无法创建服务器。如果您在 2021 年 5 月 19 日当天或之前已经在 AWS账户EndpointType=VPC_ENDPOINT中创建了服务器，则不会受到影响。在此日期之后，使用 EndpointType = VPC。
> 有关更多信息，请参阅 停止使用 VPC_ENDPOINT。
> 建议您使用 VPC 作为 EndpointType。使用此终端节点类型，您可以选择将最多三个弹性 IPv4 地址（包括 BYO IP）直接与服务器的终端节点相关联，并使用 VPC 安全组限制客户端的公有 IP 地址的流量。如果 EndpointType 设置为 VPC_ENDPOINT，则无法实现此操作。

类型：字符串

有效值：PUBLIC | VPC | VPC_ENDPOINT

必需：否

HostKey

用于启用 SFTP 的服务器的 RSA、ECDSA 或 ED25519 私钥。如果要轮换密钥，可以添加多个主机密钥，也可以添加一组使用不同算法的活动密钥。

使用以下命令生成不带密码的 RSA 2048 位密钥：

```
ssh-keygen -t rsa -b 2048 -N "" -m PEM -f my-new-server-key.
```

为 -b 选项使用最小值 2048。您可以使用 3072 或 4096 创建更强的密钥。

使用以下命令生成不带密码的 ECDSA 256 位密钥：

```
ssh-keygen -t ecdsa -b 256 -N "" -m PEM -f my-new-server-key.
```

ECDSA 的 -b 选项的有效值为 256、384 和 521。

使用以下命令生成不带密码的 ED25519 密钥：

```
ssh-keygen -t ed25519 -N "" -f my-new-server-key.
```

对于所有这些命令，你可以my-new-server-key用你选择的字符串替换。

> **Important**
>
> 如果您不打算将现有用户从启用 SFTP 的现有服务器迁移到新服务器，不要更新主机密钥。意外更改服务器的主机密钥会导致中断。

有关更多信息，请参阅《用户指南》中的更新启用 SFTP 的服务器的主机密钥。 AWS Transfer Family

类型：字符串

长度约束：最小长度为 0。最大长度为 4096。

必需：否

IdentityProviderDetails

包含调用客户身份验证 API 方法所需的所有信息的数组。

类型：IdentityProviderDetails 对象

必需：否

LoggingRole

(IAM) 角色的亚马逊资源名称 (ARN)，它允许服务器为亚马逊 S3 或 Amazon efSevents 开启亚马逊 CloudWatch 日志记录。 AWS Identity and Access Management 设置后，您可以在 CloudWatch 日志中查看用户活动。

类型：字符串

长度约束：最小长度为 0。最大长度为 2048。

模式：`(|arn:.*role/\S+)`

必需：否

PostAuthenticationLoginBanner

指定用户连接到服务器时要显示的字符串。此字符串在用户进行身份验证后显示。

> **Note**
>
> SFTP 协议不支持身份验证后显示横幅。

类型：字符串

长度约束：最小长度为 0。最大长度为 4096。

模式：`[\x09-\x0D\x20-\x7E]*`

必需：否

PreAuthenticationLoginBanner

指定用户连接到服务器时要显示的字符串。此字符串在用户进行身份验证前显示。例如，以下横幅显示有关使用系统的详细信息：

```
This system is for the use of authorized users only. Individuals using
this computer system without authority, or in excess of their authority,
are subject to having all of their activities on this system monitored
and recorded by system personnel.
```

类型：字符串

长度约束：最小长度为 0。最大长度为 4096。

模式：`[\x09-\x0D\x20-\x7E]*`

必需：否

ProtocolDetails

为服务器配置的协议设置。

- 要指示被动模式（适用于 FTP 和 FTPS 协议），请使用 `PassiveIp` 参数。输入一个点分四组的 IPv4 地址，例如防火墙、路由器或负载均衡器的外部 IP 地址。
- 要忽略当客户端尝试对您上传到 S3 桶的文件使用 `SETSTAT` 命令时生成的错误，请使用 `SetStatOption` 参数。要让 AWS Transfer Family 服务器忽略`SETSTAT`命令并上传文件而不必对 SFTP 客户端进行任何更改，请将该值设置为。`ENABLE_NO_OP`如果您将`SetStatOption`参数设置为`ENABLE_NO_OP`，Transfer Family 会生成一个到 Amazon Logs 的 CloudWatch 日志条目，以便您可以确定客户何时`SETSTAT`拨打电话。
- 要确定您的 AWS Transfer Family 服务器是否通过唯一的会话 ID 恢复最近协商的会话，请使用`TlsSessionResumptionMode`参数。
- `As2Transports` 指示 AS2 消息的传输方法。目前仅支持 HTTP。

类型：ProtocolDetails 对象

必需：否

Protocols

指定文件传输协议客户端可以用来连接到服务器端点的一个或多个文件传输协议。可用的协议包括：

- SFTP（Secure Shell (SSH) 文件传输协议）：通过 SSH 的文件传输
- FTPS（文件传输协议安全）：使用 TLS 加密的文件传输
- FTP（文件传输协议）：未加密的文件传输
- AS2（适用性声明 2）：用于传输结构化 business-to-business 数据

> **Note**
>
> - 如果选择FTPS，则必须选择存储在 AWS Certificate Manager (ACM) 中的证书，当客户端通过 FTPS 连接到服务器时，该证书用于识别您的服务器。
> - 如果 Protocol 包括 FTP 或 FTPS，则 EndpointType 必须为 VPC，且 IdentityProviderType 必须为 AWS_DIRECTORY_SERVICE、AWS_LAMBDA 或 API_GATEWAY。
> - 如果 Protocol 包含 FTP，则无法关联 AddressAllocationIds。
> - 如果仅将 Protocol 设置为 SFTP，则可以将 EndpointType 设置为 PUBLIC，并且可以将 IdentityProviderType 设置为任何支持的身份类型：SERVICE_MANAGED、AWS_DIRECTORY_SERVICE、AWS_LAMBDA 或 API_GATEWAY。
> - 如果 Protocol 包括 AS2，则 EndpointType 必须是 VPC，并且域必须是 Amazon S3。

类型：字符串数组

数组成员：最少 1 个物品。最多 4 项。

有效值：SFTP | FTP | FTPS | AS2

必需：否

S3StorageOptions

指定是否对您的 Amazon S3 目录的性能进行了优化。默认情况下，将禁用该功能。

默认情况下，主目录映射TYPE的值为。DIRECTORY如果启用此选项，则需要将显式设置为，HomeDirectoryMapEntryType以FILE使映射具有文件目标。

类型：S3StorageOptions 对象

必需：否

SecurityPolicyName

指定服务器安全策略的名称。

类型：字符串

长度限制：长度下限为 0。最大长度为 100。

模式：Transfer[A-Za-z0-9]*SecurityPolicy-[A-Za-z0-9-]+

必需：否

ServerId

分配给 Transfer Family 用户的服务器实例的系统分配的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

必需：是

StructuredLogDestinations

指定将服务器日志发送到日志组。

要指定日志组，必须提供现有日志组的 ARN。在这种情况下，日志组的格式如下所示：

arn:aws:logs:region-name:amazon-account-id:log-group:log-group-name:*

例如，arn:aws:logs:us-east-1:111122223333:log-group:mytestgroup:*

如果您之前为服务器指定了日志组，则可以通过在update-server调用中为该参数提供空值来清除该日志组，从而关闭结构化日志记录。例如：

```
update-server --server-id s-1234567890abcdef0 --structured-log-
destinations
```

类型：字符串数组

数组成员：最少 0 项。最多 1 项。

长度约束：最小长度为 20。长度上限为 1600。

模式：arn:\S+

必需：否

WorkflowDetails

指定要分配的工作流的工作流 ID 以及用于执行工作流的执行角色。

除了要在文件完全上传时执行的工作流，WorkflowDetails 还可能包含在部分文件上传时执行的工作流的工作流 ID（和执行角色）。在文件仍在上传时，如果断开连接，则会发生部分上传。

要从服务器中删除关联的工作流，您可以提供一个空的 OnUpload 对象，如下例所示。

```
aws transfer update-server --server-id s-01234567890abcdef --workflow-details '{"OnUpload":[]}'
```

类型：WorkflowDetails 对象

必需：否

响应语法

```
{
   "ServerId": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

ServerId

分配给 Transfer Family 用户的服务器的系统分配的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

AccessDeniedException

您没有足够的访问权限，无法执行该操作。

HTTP 状态代码：400

ConflictException

当为启用文件传输协议的服务器调用 UpdateServer 时，而该服务器以 VPC 作为端点类型且该服务器的 VpcEndpointID 未处于可用状态时，会引发此异常。

HTTP 状态代码：400

InternalServiceError

当 AWS Transfer Family 服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceExistsException

请求的资源不存在，或者存在于为命令指定的区域以外的区域。

HTTP 状态代码：400

ResourceNotFoundException

当 Transfer Family 服务找不到资源时，就会 AWS引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 Trans AWS fer Family 服务不可用。

HTTP 状态代码：500

ThrottlingException

由于请求限制而导致请求被拒绝。

HTTP 状态代码：400

示例

示例

以下示例将更新服务器的角色。

示例请求

```
{
   "EndpointDetails": {
   "VpcEndpointId": "vpce-01234f056f3g13",
   "LoggingRole": "CloudWatchS3Events",
   "ServerId": "s-01234567890abcdef"
   }
}
```

示例

以下示例从服务器中移除所有关联的工作流。

示例请求

```
aws transfer update-server --server-id s-01234567890abcdef --workflow-details
 '{"OnUpload":[]}'
```

示例

这是此 API 调用的示例响应。

示例响应

```
{
   "ServerId": "s-01234567890abcdef"
}
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面

- AWS 适用于 .NET 的 SDK
- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS JavaScript V3 版软件开发工具包
- AWS 适用于 PHP 的 SDK V3
- AWS Python 软件开发工具包
- AWS 适用于 Ruby V3 的 SDK

UpdateUser

将新属性分配给用户。您传递的参数会修改以下任何或所有内容：您指定的UserName和ServerId的主目录、角色和策略。

响应为更新后的用户返回ServerId和UserName。

在控制台中，您可以在创建或更新用户时选择 “受限”。这样可以确保用户无法访问其主目录之外的任何内容。配置此行为的编程方法是更新用户。将它们设置HomeDirectoryType为LOGICAL，并HomeDirectoryMappings使用Entry作为root (/) 和Target主目录来指定。

例如，如果用户的主目录是/test/admin-user，则以下命令会更新用户，以便他们在控制台中的配置显示 Restricted 标志为选中。

```
aws transfer update-user --server-id <server-id> --user-name admin-user --
home-directory-type LOGICAL --home-directory-mappings "[{\"Entry\":\"/\",
\"Target\":\"/test/admin-user\"}]"
```

请求语法

```
{
   "HomeDirectory": "string",
   "HomeDirectoryMappings": [
      {
         "Entry": "string",
         "Target": "string",
         "Type": "string"
      }
   ],
   "HomeDirectoryType": "string",
   "Policy": "string",
   "PosixProfile": {
      "Gid": number,
      "SecondaryGids": [ number ],
      "Uid": number
   },
   "Role": "string",
   "ServerId": "string",
   "UserName": "string"
}
```

请求参数

有关所有操作的通用参数的信息，请参阅通用参数。

请求接受采用 JSON 格式的以下数据。

HomeDirectory

用户使用客户端登录服务器时的登录目录（文件夹）。

HomeDirectory 示例为 /bucket_name/home/mydirectory。

> **Note**
>
> HomeDirectory 参数仅在 HomeDirectoryType 设置为 PATH 时使用。

类型：字符串

长度限制：长度下限为 0。最大长度为 1024。

模式：(|/.*)

必需：否

HomeDirectoryMappings

逻辑目录映射指定哪些 Amazon S3 或 Amazon EFS 路径和密钥应对您的用户可见，以及使其对用户可见的方式。您需要指定Entry和Target对，其中 Entry 显示如何使路径可见，Target 是实际的 Amazon S3 或 Amazon EFS 路径。如果您只指定一个目标，则将按原样显示。您还必须确保您的AWS Identity and Access Management（IAM）角色提供对Target中路径的访问权限。只有当HomeDirectoryType设置为LOGICAL时，才能设置此值。

以下是 Entry 和 Target 对示例。

```
[ { "Entry": "/directory1", "Target": "/bucket_name/home/
mydirectory" } ]
```

在大多数情况下，您可以使用此值而不是会话策略将您的用户锁定到指定的主目录（“chroot”）。为此，您可以将 Entry 设置为“/”并将 Target 设置为 HomeDirectory 参数值。

以下是chroot的 Entry 和 Target 对示例。

```
[ { "Entry": "/", "Target": "/bucket_name/home/mydirectory" } ]
```

类型：HomeDirectoryMapEntry 对象数组

数组成员：最少 1 个物品。最大数量为 50000 件物品。

必需：否

HomeDirectoryType

您希望用户在登录服务器时，用户主目录的登录目录（文件夹）的类型。如果您将其设置为PATH，则用户将在其文件传输协议客户端中原样看到 Amazon S3 存储桶或 Amazon EFS 的绝对路径。如果您将其设置为LOGICAL，则需要针对您希望如何使 Amazon S3 或 Amazon EFS 路径对用户可见，在HomeDirectoryMappings中提供映射。

> **Note**
>
> 如果HomeDirectoryType是LOGICAL，则必须使用HomeDirectoryMappings参数提供映射。另一方面，如果HomeDirectoryType是PATH，则使用HomeDirectory参数提供绝对路径。您的模板中不能同时使用HomeDirectory和HomeDirectoryMappings。

类型：字符串

有效值：PATH | LOGICAL

必需：否

Policy

适用于您的用户的会话策略，可让您跨多个用户使用相同的 AWS Identity and Access Management(IAM) 角色。此策略将用户的访问权限缩小至 Amazon S3 存储桶的一部分。可在此策略中使用的变量包括 ${Transfer:UserName}、${Transfer:HomeDirectory} 和 ${Transfer:HomeBucket}。

> **Note**
>
> 仅当ServerId域为 Amazon S3 时，此政策才适用。Amazon EFS 不使用会话策略。
> 对于会话测量，AWS Transfer Family 将策略存储为 JSON blob，而不是策略的 Amazon 资源名称（ARN）。您将策略保存为 JSON blob 并将其传递给 Policy 参数。
> 有关会话策略的示例，请参阅示例会话策略。

有关更多信息，请参阅AssumeRole《AWS安全令牌服务 API 参考》。

类型：字符串

长度约束：最小长度为 0。最大长度为 2048。

必需：否

PosixProfile

指定完整的 POSIX 身份，包括用户 ID (Uid)、组 ID (Gid) 和任何辅助组 ID (SecondaryGids)，用于控制用户对 Amazon Elastic File System (Amazon EFS) 文件系统的访问权限。POSIX 权限针对文件系统中的文件和目录设置，用于确定用户在将文件传入和传出 Amazon EFS 文件系统时获得的访问权限级别。

类型：PosixProfile 对象

必需：否

Role

控制用户对 Amazon S3 桶或 Amazon EFS 文件系统访问权限的 AWS Identity and Access Management（IAM）角色的 Amazon 资源名称（ARN）。附加到此角色的策略确定在将文件传入和传出 Amazon S3 桶或 Amazon EFS 文件系统时要为用户提供的访问权限级别。IAM 角色还应包含一个信任关系，从而允许服务器在为用户的传输请求提供服务时访问您的资源。

类型：字符串

长度约束：最小长度为 20。最大长度为 2048。

模式：`arn:.*role/\S+`

必需：否

ServerId

系统为分配给用户的 Transfer Family 服务器实例指定的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`s-([0-9a-f]{17})`

必需：是

UserName

一个标识用户并与 ServerId 指定的服务器关联的唯一字符串。此用户名长度最少必须为 3 个字符，最多为 100 个字符。以下是有效的字符：a-z、A-Z、0-9、下划线“_”、连字符“-”、句点“.”和“@”符号。用户名不能以连字符、句点或 @ 符号开头。

类型：字符串

长度限制：长度下限为 3。最大长度为 100。

模式：[\w][\w@.-]{2,99}

必需：是

响应语法

```
{
   "ServerId": "string",
   "UserName": "string"
}
```

响应元素

如果此操作成功，则该服务将会发送回 HTTP 200 响应。

服务以 JSON 格式返回以下数据。

ServerId

系统为分配给账户的 Transfer Family 服务器实例指定的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

UserName

分配给请求中指定的服务器实例的用户的唯一标识符。

类型：字符串

长度限制：长度下限为 3。最大长度为 100。

模式：`[\w][\w@.-]{2,99}`

错误

有关所有操作返回的常见错误的信息，请参阅常见错误。

InternalServiceError

当 AWS Transfer Family服务中发生错误时，会引发此异常。

HTTP 状态代码：500

InvalidRequestException

当客户端提交格式错误的请求时，会引发此异常。

HTTP 状态代码：400

ResourceNotFoundException

当 AWSTransfer Family 服务找不到资源时，就会引发此异常。

HTTP 状态代码：400

ServiceUnavailableException

请求失败，因为 AWSTransfer Family 服务不可用。

HTTP 状态代码：500

ThrottlingException

由于请求限制而导致请求被拒绝。

HTTP 状态代码：400

示例

示例

以下示例将更新一个 Transfer Family 用户。

示例请求

```
{
   "HomeDirectory": "/bucket2/documentation",
   "HomeDirectoryMappings": [
      {
         "Entry": "/directory1",
         "Target": "/bucket_name/home/mydirectory"
      }
   ],
   "HomeDirectoryType:" "PATH",
   "Role": "AssumeRole",
   "ServerId": "s-01234567890abcdef",
   "UserName": "my_user"
}
```

示例

这是此 API 调用的示例响应。

示例响应

```
{
   "ServerId": "s-01234567890abcdef",
   "UserName": "my_user"
}
```

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 命令行界面
- 适用于 .NET 的 AWS SDK
- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- AWS JavaScript V3 版软件开发工具包
- 适用于 PHP V3 的 AWS SDK
- 适用于 Python 的 AWS SDK

- 适用于 Ruby V3 的 AWS SDK

数据类型

支持以下数据类型：

- As2ConnectorConfig
- CopyStepDetails
- CustomStepDetails
- DecryptStepDetails
- DeleteStepDetails
- DescribedAccess
- DescribedAgreement
- DescribedCertificate
- DescribedConnector
- DescribedExecution
- DescribedHostKey
- DescribedProfile
- DescribedSecurityPolicy
- DescribedServer
- DescribedUser
- DescribedWorkflow
- EfsFileLocation
- EndpointDetails
- ExecutionError
- ExecutionResults
- ExecutionStepResult
- FileLocation
- HomeDirectoryMapEntry
- IdentityProviderDetails
- InputFileLocation

- ListedAccess
- ListedAgreement
- ListedCertificate
- ListedConnector
- ListedExecution
- ListedHostKey
- ListedProfile
- ListedServer
- ListedUser
- ListedWorkflow
- LoggingConfiguration
- PosixProfile
- ProtocolDetails
- S3FileLocation
- S3InputFileLocation
- S3StorageOptions
- S3Tag
- ServiceMetadata
- SftpConnectorConfig
- SshPublicKey
- Tag
- TagStepDetails
- UserDetails
- WorkflowDetail
- WorkflowDetails
- WorkflowStep

As2ConnectorConfig

包含 AS2 连接器对象详细信息。连接器对象用于 AS2 出站流程，用于将 AWS Transfer Family 客户与贸易伙伴连接起来。

内容

BasicAuthSecretId

为 AS2 连接器 API 提供基本身份验证支持。要使用基本身份验证，您必须在 AWS Secrets Manager提供密钥的名称或 Amazon 资源名称 (ARN)。

此参数的默认值为`null`，表示未启用连接器的基本身份验证。

如果连接器应使用基本身份验证，则密钥必须采用以下格式：

```
{ "Username": "user-name", "Password": "user-password" }
```

将`user-name`和`user-password`替换为正在进行身份验证的实际用户的凭证。

请注意以下几点：

- 您将这些凭证存储在 Secrets Manager 中，而不是将它们直接传递到此 API 中。
- 如果您正在使用 API、SDK 或 CloudFormation 配置连接器，则必须先创建密钥，然后才能启用基本身份验证。但是，如果您使用的是 AWS 管理控制台，则可以让系统为您创建密钥。

如果您之前为连接器启用了基本身份验证，则可以使用`UpdateConnector` API 调用将其禁用。例如，如果您使用的是 CLI，则可以运行以下命令来删除基本身份验证：

```
update-connector --connector-id my-connector-id --as2-config
'BasicAuthSecretId=""'
```

类型：字符串

长度约束：最小长度为 0。最大长度为 2048。

必需：否

Compression

指定 AS2 文件是否被压缩。

类型：字符串

有效值：ZLIB | DISABLED

必需：否

EncryptionAlgorithm

用于加密文件的算法。

请注意以下几点：

- 除非必须支持需要该DES_EDE3_CBC算法的旧版客户端，否则请勿使用该算法，因为它是一种弱加密算法。
- 如果连接器的 URL 使用 HTTPS，则只能指定NONE。使用 HTTPS 可确保不会以明文形式发送任何流量。

类型：字符串

有效值：AES128_CBC | AES192_CBC | AES256_CBC | DES_EDE3_CBC | NONE

必需：否

LocalProfileId

AS2 本地配置文件的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：p-([0-9a-f]{17})

必需：否

MdnResponse

用于出站请求（从 AWS Transfer Family 服务器到伙伴 AS2 服务器），以确定合作伙伴对传输的响应是同步的还是异步的。指定以下任一值：

- SYNC：系统需要同步的 MDN 响应，确认文件已成功传输（或未失败）。
- NONE：指定不需要 MDN 响应。

类型：字符串

有效值：SYNC | NONE

必需：否

MdnSigningAlgorithm

MDN 响应的签名算法。

> **Note**
>
> 如果设置为 DEFAULT（或根本未设置），则使用SigningAlgorithm值。

类型：字符串

有效值：SHA256 | SHA384 | SHA512 | SHA1 | NONE | DEFAULT

必需：否

MessageSubject

用作使用连接器发送的 AS2 消息中的 SubjectHTTP 标头属性。

类型：字符串

长度限制：长度下限为 1。长度上限为 1024。

模式：[\p{Print}\p{Blank}]+

必需：否

PartnerProfileId

连接器的合作伙伴配置文件的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：p-([0-9a-f]{17})

必需：否

SigningAlgorithm

用于对通过连接器发送的 AS2 消息进行签名的算法。

类型：字符串

有效值：SHA256 | SHA384 | SHA512 | SHA1 | NONE

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS 适用于 Ruby V3 的 SDK

CopyStepDetails

每种步骤类型都有自己的StepDetails结构。

目录

DestinationFileLocation

指定要复制的文件的位置。在此字段中使用${Transfer:UserName}或${Transfer:UploadDate}按用户名或上传日期参数化目标前缀。

- 将DestinationFileLocation的值设置为${Transfer:UserName}，将上传的文件复制到以上传文件的 Transfer Family 用户名为前缀的 Amazon S3 存储桶。
- 将DestinationFileLocation的值设置为${Transfer:UploadDate}，将上传的文件复制到以上传日期为前缀的 Amazon S3 存储桶。

> **Note**
>
> 根据以 UTC 格式上传文件的日期，系统会解析UploadDate为 YYYY-MM-DD 的日期格式。

类型：InputFileLocation 对象

必需：否

Name

步骤的名称，用作标识符。

类型：字符串

长度约束：最小长度为 0。最大长度为 30。

模式：[\w-]*

必需：否

OverwriteExisting

指示是否覆盖现有同名文件的标志。默认为 FALSE。

如果工作流程正在处理与现有文件同名的文件，则行为如下：

- 如果OverwriteExisting是TRUE，则现有文件会被正在处理的文件替换。
- 如果OverwriteExisting是FALSE，则什么也不会发生，工作流程处理就会停止。

类型：字符串

有效值：TRUE | FALSE

必需：否

SourceFileLocation

指定将哪个文件用作工作流程步骤的输入：要么是上一步的输出，要么是为工作流程最初上传的文件。

- 要使用前一个文件作为输入，请输入${previous.file}。在这种情况下，此工作流程步骤使用上一个工作流程步骤的输出文件作为输入。这是默认值。
- 要使用最初上传的文件位置作为此步骤的输入，请输入${original.file}。

类型：字符串

长度约束：最小长度为 0。长度上限为 256。

模式：\$\{(\w+.)+\w+\}

必需：否

另请参阅

有关在特定语言的 AWS 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

CustomStepDetails

每种步骤类型都有自己的StepDetails结构。

目录

Name

步骤的名称，用作标识符。

类型：字符串

长度约束：最小长度为 0。最大长度为 30。

模式：[\w-]*

必需：否

SourceFileLocation

指定将哪个文件用作工作流程步骤的输入：要么是上一步的输出，要么是为工作流程最初上传的文件。

- 要使用前一个文件作为输入，请输入${previous.file}。在这种情况下，此工作流程步骤使用上一个工作流程步骤的输出文件作为输入。这是默认值。
- 要使用最初上传的文件位置作为此步骤的输入，请输入${original.file}。

类型：字符串

长度约束：最小长度为 0。长度上限为 256。

模式：\$\{(\w+.)+\w+\}

必需：否

Target

所调用的 Lambda 函数的 ARN。

类型：字符串

长度约束：最小长度为 0。长度上限为 170。

模式：arn:[a-z-]+:lambda:.*

必需：否

TimeoutSeconds

步骤的超时值（以秒为单位）。

类型：整数

有效范围：最小值为 1。最大值为 1800。

必需：否

另请参阅

有关在特定语言的 AWS 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DecryptStepDetails

每种步骤类型都有自己的StepDetails结构。

目录

DestinationFileLocation

指定正在解密的文件的位置。在此字段中使用${Transfer:UserName}或${Transfer:UploadDate}按用户名或上传日期参数化目标前缀。

- 将DestinationFileLocation的值设置为${Transfer:UserName}，将上传的文件解密到以上传文件的 Transfer Family 用户名为前缀的 Amazon S3 存储桶。
- 将DestinationFileLocation的值设置为${Transfer:UploadDate}，将上传的文件解密到以上传日期为前缀的 Amazon S3 存储桶。

> **Note**
>
> 根据以 UTC 格式上传文件的日期，系统会解析UploadDate为 YYYY-MM-DD 的日期格式。

类型：InputFileLocation 对象

必需：是

Type

使用的加密类型。目前，该值必须为 PGP。

类型：字符串

有效值：PGP

必需：是

Name

步骤的名称，用作标识符。

类型：字符串

长度约束：最小长度为 0。最大长度为 30。

模式：[\w-]*

必需：否

OverwriteExisting

指示是否覆盖现有同名文件的标志。默认为 FALSE。

如果工作流程正在处理与现有文件同名的文件，则行为如下：

- 如果OverwriteExisting是TRUE，则现有文件会被正在处理的文件替换。
- 如果OverwriteExisting是FALSE，则什么也不会发生，工作流程处理就会停止。

类型：字符串

有效值：TRUE | FALSE

必需：否

SourceFileLocation

指定将哪个文件用作工作流程步骤的输入：要么是上一步的输出，要么是为工作流程最初上传的文件。

- 要使用前一个文件作为输入，请输入${previous.file}。在这种情况下，此工作流程步骤使用上一个工作流程步骤的输出文件作为输入。这是默认值。
- 要使用最初上传的文件位置作为此步骤的输入，请输入${original.file}。

类型：字符串

长度约束：最小长度为 0。长度上限为 256。

模式：\$\{(\w+.)+\w+\}

必需：否

另请参阅

有关在特定语言的 AWS 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK

- 适用于 Ruby V3 的 AWS SDK

DeleteStepDetails

步骤的名称，用于标识删除步骤。

目录

Name

步骤的名称，用作标识符。

类型：字符串

长度约束：最小长度为 0。最大长度为 30。

模式：[\w-]*

必需：否

SourceFileLocation

指定将哪个文件用作工作流程步骤的输入：要么是上一步的输出，要么是为工作流程最初上传的文件。

- 要使用前一个文件作为输入，请输入${previous.file}。在这种情况下，此工作流程步骤使用上一个工作流程步骤的输出文件作为输入。这是默认值。
- 要使用最初上传的文件位置作为此步骤的输入，请输入${original.file}。

类型：字符串

长度约束：最小长度为 0。长度上限为 256。

模式：\$\{(\w+.)+\w+\}

必需：否

另请参阅

有关在特定语言的 AWS 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK

- 适用于 Ruby V3 的 AWS SDK

DescribedAccess

描述指定用户的属性。

内容

ExternalId

识别目录中特定群组所需的唯一标识符。您关联的组中的用户可以通过使用AWS Transfer Family的启用协议访问您的 Amazon S3 或 Amazon EFS 资源。如果您知道组名，则可以使用 Windows 运行以下命令来查看 SID 值 PowerShell。

```
Get-ADGroup -Filter {samAccountName -like "YourGroupName*"} -Properties
* | Select SamAccountName,ObjectSid
```

在该命令中，替换为您的 YourGroupNameActive Directory 组的名称。

用于验证此参数的正则表达式是由不带空格的大写和小写字母数字字符组成的字符串。此外，还可以包括下划线或以下任何字符：=、.@：/-

类型：字符串

长度限制：最小长度为 1。最大长度为 256。

模式：`S-1-[\d-]+`

必需：否

HomeDirectory

用户使用客户端登录服务器时的登录目录（文件夹）。

`HomeDirectory` 示例为 `/bucket_name/home/mydirectory`。

> **Note**
>
> `HomeDirectory` 参数仅在 `HomeDirectoryType` 设置为 `PATH` 时使用。

类型：字符串

长度限制：长度下限为 0。最大长度为 1024。

模式：`(|/.*)`

必需：否

HomeDirectoryMappings

逻辑目录映射指定哪些 Amazon S3 或 Amazon EFS 路径和密钥应对您的用户可见，以及使其对用户可见的方式。您需要指定Entry和Target对，其中 Entry 显示如何使路径可见，Target 是实际的 Amazon S3 或 Amazon EFS 路径。如果您只指定一个目标，则将按原样显示。您还必须确保您的AWS Identity and Access Management（IAM）角色提供对Target中路径的访问权限。只有当HomeDirectoryType设置为LOGICAL时，才能设置此值。

在大多数情况下，您可以使用此值而不是会话策略将您的用户锁定到指定的主目录（“chroot”）。为此，您可以将 Entry 设置为“/”并将 Target 设置为 HomeDirectory 参数值。

类型：HomeDirectoryMapEntry 对象数组

数组成员：最少 1 个物品。最大数量为 50000 个。

必需：否

HomeDirectoryType

您希望用户在登录服务器时，用户主目录的登录目录（文件夹）的类型。如果您将其设置为PATH，则用户将在其文件传输协议客户端中原样看到 Amazon S3 存储桶或 Amazon EFS 的绝对路径。如果您将其设置为LOGICAL，则需要针对您希望如何使 Amazon S3 或 Amazon EFS 路径对用户可见，在HomeDirectoryMappings中提供映射。

> **Note**
>
> 如果HomeDirectoryType是LOGICAL，则必须使用HomeDirectoryMappings参数提供映射。另一方面，如果HomeDirectoryType是PATH，则使用HomeDirectory参数提供绝对路径。您的模板中不能同时使用HomeDirectory和HomeDirectoryMappings。

类型：字符串

有效值：PATH | LOGICAL

必需：否

Policy

适用于您的用户的会话策略，可让您跨多个用户使用相同的 AWS Identity and Access Management(IAM) 角色。此策略将用户的访问权限缩小至 Amazon S3 存储桶的一部分。可在

此策略中使用的变量包括 ${Transfer:UserName}、${Transfer:HomeDirectory} 和 ${Transfer:HomeBucket}。

类型：字符串

长度约束：最小长度为 0。最大长度为 2048。

必需：否

PosixProfile

完整的 POSIX 身份，包括用户 ID (Uid)、组 ID (Gid) 和任何辅助组 ID (SecondaryGids)，用于控制用户对 Amazon EFS 文件系统的访问。POSIX 权限针对文件系统中的文件和目录设置，用于确定用户在将文件传入和传出 Amazon EFS 文件系统时获得的访问权限级别。

类型：PosixProfile 对象

必需：否

Role

控制用户对 Amazon S3 桶或 Amazon EFS 文件系统访问权限的 AWS Identity and Access Management（IAM）角色的 Amazon 资源名称（ARN）。附加到此角色的策略确定在将文件传入和传出 Amazon S3 桶或 Amazon EFS 文件系统时要为用户提供的访问权限级别。IAM 角色还应包含一个信任关系，从而允许服务器在为用户的传输请求提供服务时访问您的资源。

类型：字符串

长度约束：最小长度为 20。最大长度为 2048。

模式：arn:.*role/\S+

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DescribedAgreement

描述协议属性。

内容

Arn

的唯一 Amazon 资源名称（ARN）。

类型：字符串

长度约束：最小长度为 20。长度上限为 1600。

模式：`arn:\S+`

必需：是

AccessRole

连接器用于通过 AS2 或 SFTP 协议发送文件。对于访问权限角色，请提供要使用的AWS Identity and Access Management角色的 Amazon 资源名称 (ARN)。

对于 AS2 连接器

借助 AS2，您可以通过调用 `StartFileTransfer` 并在请求参数中指定文件路径 `SendFilePaths` 来发送文件。我们使用文件的父目录（例如 `--send-file-paths /bucket/dir/file.txt`，父目录是 `/bucket/dir/`）来临时存储经过处理的 AS2 消息文件，存储 MDN（当从合作伙伴那里收到时），以及写入包含传输相关元数据的最终 JSON 文件。因此，`AccessRole` 需要提供对 `StartFileTransfer` 请求中所使用文件位置父目录的读取和写入权限。此外，您还需要提供对您想要使用 `StartFileTransfer` 发送的文件父目录的读取和写入权限。

如果您对 AS2 连接器执行基本身份验证，则访问角色需要密钥`secretsmanager:GetSecretValue`权限。如果 Secrets Manager 中的秘钥加密方式使用客户托管秘钥，而非AWS托管秘钥，则该角色需要此秘钥`kms:Decrypt`的权限。

对于 SFTP 连接器

因此, 确保提供对 `StartFileTransfer` 请求中所使用文件位置父目录的读取和写入权限。此外，请确保该角色向`secretsmanager:GetSecretValue`提供AWS Secrets Manager权限。

类型：字符串

长度约束：最小长度为 20。最大长度为 2048。

模式：arn:.*role/\S+

必需：否

AgreementId

的唯一标识符。创建协议时会返回此标识符。

类型：字符串

长度限制：固定长度为 19。

模式：a-([0-9a-f]{17})

必需：否

BaseDirectory

使用 AS2 协议传输的文件的登录目录（文件夹）。

类型：字符串

长度限制：长度下限为 0。最大长度为 1024。

模式：(|/.*)

必需：否

Description

用于识别协议的名称或简短描述。

类型：字符串

长度限制：最小长度为 1。最大长度为 200。

模式：[\p{Graph}]+

必需：否

LocalProfileId

AS2 本地配置文件的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`p-([0-9a-f]{17})`

必需：否

PartnerProfileId

协议中使用的合作伙伴配置文件的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`p-([0-9a-f]{17})`

必需：否

ServerId

服务器实例的系统分配的唯一标识符。此标识符表示协议使用的特定服务器。

类型：字符串

长度限制：固定长度为 19。

模式：`s-([0-9a-f]{17})`

必需：否

Status

协议的当前状态为 `ACTIVE` 或 `INACTIVE`。

类型：字符串

有效值：`ACTIVE | INACTIVE`

必需：否

Tags

可用于分组和搜索协议的键/值对。

类型：Tag 对象数组

数组成员：最少 1 个物品。最多 50 项。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DescribedCertificate

描述证书的属性。

内容

Arn

证书的唯一 Amazon 资源名称（ARN）。

类型：字符串

长度约束：最小长度为 20。长度上限为 1600。

模式：`arn:\S+`

必需：是

ActiveDate

指定证书何时生效的可选日期。

类型：时间戳

必需：否

Certificate

证书的文件名。

类型：字符串

长度限制：长度下限为 1。长度上限为 16384。

模式：`[\u0009\u000A\u000D\u0020-\u00FF]*`

必需：否

CertificateChain

构成证书链的证书列表。

类型：字符串

长度限制：长度下限为 1。最大长度为 2097152。

模式：[\u0009\u000A\u000D\u0020-\u00FF]*

必需：否

CertificateId

已导入证书的标识符数组。您可以使用此标识符来处理配置文件和合作伙伴配置文件。

类型：字符串

长度限制：固定长度为 22。

模式：cert-([0-9a-f]{17})

必需：否

Description

用于识别证书的名称或描述。

类型：字符串

长度限制：最小长度为 1。最大长度为 200。

模式：[\p{Graph}]+

必需：否

InactiveDate

指定证书何时失效的可选日期。

类型：时间戳

必需：否

NotAfterDate

证书有效的最后日期。

类型：时间戳

必需：否

NotBeforeDate

证书有效的最早日期。

类型：时间戳

必需：否

Serial

证书的序列号。

类型：字符串

长度约束：最小长度为 0。最大长度为 48。

模式：`[\p{XDigit}{2}:?]*`

必需：否

Status

证书可以是 ACTIVE、PENDING_ROTATION、或 INACTIVE。PENDING_ROTATION 意味着此证书将在当前证书到期时替换当前证书。

类型：字符串

有效值：`ACTIVE | PENDING_ROTATION | INACTIVE`

必需：否

Tags

可用于分组和搜索证书的键/值对。

类型：Tag 对象数组

数组成员：最少 1 个物品。最多 50 项。

必需：否

Type

如果为该证书指定了私有密钥，则其类型为 CERTIFICATE_WITH_PRIVATE_KEY。如果没有私有密钥，则类型为 CERTIFICATE。

类型：字符串

有效值：`CERTIFICATE | CERTIFICATE_WITH_PRIVATE_KEY`

必需：否

Usage

指定如何使用此证书。它可以通过以下方式使用：

- SIGNING: 用于对 AS2 消息进行签名
- ENCRYPTION: 用于加密 AS2 消息
- TLS: 用于保护通过 HTTPS 发送的 AS2 通信

类型：字符串

有效值：SIGNING | ENCRYPTION

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS 适用于 Ruby V3 的 SDK

DescribedConnector

描述连接器的参数，由ConnectorId标识。

内容

Arn

的唯一 Amazon 资源名称（ARN）。

类型：字符串

长度约束：最小长度为 20。长度上限为 1600。

模式：arn:\S+

必需：是

AccessRole

连接器用于通过 AS2 或 SFTP 协议发送文件。对于访问角色，请提供要使用的 AWS Identity and Access Management 角色的 Amazon 资源名称 (ARN)。

对于 AS2 连接器

借助 AS2，您可以通过调用 StartFileTransfer 并在请求参数中指定文件路径 SendFilePaths 来发送文件。我们使用文件的父目录（例如 --send-file-paths /bucket/dir/file.txt，父目录是 /bucket/dir/）来临时存储经过处理的 AS2 消息文件，存储 MDN（当从合作伙伴那里收到时），以及写入包含传输相关元数据的最终 JSON 文件。因此，AccessRole 需要提供对 StartFileTransfer 请求中所使用文件位置父目录的读取和写入权限。此外，您还需要提供对您想要使用 StartFileTransfer 发送的文件父目录的读取和写入权限。

如果您对 AS2 连接器执行基本身份验证，则访问角色需要密钥secretsmanager:GetSecretValue权限。如果使用客户管理的密钥而不是 Secrets Manager 中的 AWS 托管密钥对密钥进行加密，则该角色还需要该密钥的kms:Decrypt权限。

对于 SFTP 连接器

因此, 确保提供对 StartFileTransfer 请求中所使用文件位置父目录的读取和写入权限。此外，请确保该角色向提供secretsmanager:GetSecretValue权限 AWS Secrets Manager。

类型：字符串

长度约束：最小长度为 20。最大长度为 2048。

模式：`arn:.*role/\S+`

必需：否

As2Config

包含 AS2 连接器对象参数的结构。

类型：As2ConnectorConfig 对象

必需：否

ConnectorId

连接器的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`c-([0-9a-f]{17})`

必需：否

LoggingRole

(IAM) 角色的亚马逊资源名称 AWS Identity and Access Management (ARN)，它允许连接器开启对 Amazon S3 CloudWatch 事件的日志记录。设置后，您可以在 CloudWatch 日志中查看连接器活动。

类型：字符串

长度约束：最小长度为 20。最大长度为 2048。

模式：`arn:.*role/\S+`

必需：否

SecurityPolicyName

指定连接器的安全策略的文本名称。

类型：字符串

长度限制：长度下限为 0。最大长度为 100。

模式：`TransferSFTPConnectorSecurityPolicy-[A-Za-z0-9-]+`

必需：否

ServiceManagedEgressIpAddresses

此连接器的出口 IP 地址列表。这些 IP 地址是在您创建连接器时自动分配的。

类型：字符串数组

模式：`\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}`

必需：否

SftpConfig

包含 SFTP 连接器对象参数的结构。

类型：SftpConnectorConfig 对象

必需：否

Tags

可用于分组和搜索连接器的键/值对。

类型：Tag 对象数组

数组成员：最少 1 个物品。最多 50 项。

必需：否

Url

合作伙伴的 AS2 或 SFTP 端点的 URL。

类型：字符串

长度约束：最小长度为 0。最大长度为 255。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS 适用于 Ruby V3 的 SDK

DescribedExecution

执行对象的详细信息。

内容

ExecutionId

工作流的唯一标识符。

类型：字符串

长度限制：固定长度为 36。

模式：`[0-9a-fA-F]{8}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{12}`

必需：否

ExecutionRole

与执行相关联的 IAM 角色。

类型：字符串

长度约束：最小长度为 20。最大长度为 2048。

模式：`arn:.*role/\S+`

必需：否

InitialFileLocation

说明 Amazon S3 或 EFS 文件位置的结构。这是执行开始时的文件位置：如果正在复制文件，则这是初始（而非目标）文件位置。

类型：FileLocation 对象

必需：否

LoggingConfiguration

与执行相关联的 IAM 日志记录角色。

类型：LoggingConfiguration 对象

必需：否

PosixProfile

完整的 POSIX 身份，包括用户 ID (Uid)、组 ID (Gid) 和任何辅助组 ID (SecondaryGids)，用于控制用户对 Amazon EFS 文件系统的访问。POSIX 权限针对文件系统中的文件和目录设置，用于确定用户在将文件传入和传出 Amazon EFS 文件系统时获得的访问权限级别。

类型：PosixProfile 对象

必需：否

Results

说明执行结果的结构。这包括步骤列表以及每个步骤的详细信息、错误类型和消息（如有）以及 OnExceptionSteps 结构。

类型：ExecutionResults 对象

必需：否

ServiceMetadata

与工作流程关联的、会话详细信息的容器对象。

类型：ServiceMetadata 对象

必需：否

Status

任务执行的状态。可能正在进行中、已完成、遇到异常或者正在处理异常。

类型：字符串

有效值：IN_PROGRESS | COMPLETED | EXCEPTION | HANDLING_EXCEPTION

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK

- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DescribedHostKey

服务器主机密钥详细信息。

内容

Arn

的唯一 Amazon 资源名称（ARN）。

类型：字符串

长度约束：最小长度为 20。长度上限为 1600。

模式：`arn:\S+`

必需：是

DateImported

将主机密钥添加至服务器的日期。

类型：时间戳

必需：否

Description

此主机密钥的文字描述。

类型：字符串

长度限制：长度下限为 0。最大长度为 200。

模式：`[\p{Print}]*`

必需：否

HostKeyFingerprint

公钥指纹，仅一小段字节序列，用于识别较长的公钥。

类型：字符串

必需：否

HostKeyId

的唯一标识符。

类型：字符串

长度限制：固定长度为 25。

模式：`hostkey-[0-9a-f]{17}`

必需：否

Tags

可用于分组和搜索服务器的键/值对。

类型：Tag 对象数组

数组成员：最少 1 个物品。最多 50 项。

必需：否

Type

用于加密的密钥类型。通过以下值指定 Type 参数：

- `ssh-rsa`
- `ssh-ed25519`
- `ecdsa-sha2-nistp256`
- `ecdsa-sha2-nistp384`
- `ecdsa-sha2-nistp521`

类型：字符串

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK

- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DescribedProfile

本地或合作伙伴 AS2 配置文件的详细信息。

内容

Arn

配置文件的唯一 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最小长度为 20。长度上限为 1600。

模式：`arn:\S+`

必需：是

As2Id

`As2Id` 是 AS2-name，如 RFC 4130 中所定义。对于入站传输，这是合作伙伴发送的 AS2 消息的 `AS2-From` 标头。对于出站连接器，这是使用 `AS2-To` API 操作发送给合作伙伴的 AS2 消息的 `StartFileTransfer` 标头。此 ID 不能包含空格。

类型：字符串

长度限制：长度下限为 1。长度上限为 128。

模式：`[\p{Print}\s]*`

必需：否

CertificateIds

已导入证书的标识符数组。您可以使用此标识符来处理配置文件和合作伙伴配置文件。

类型：字符串数组

长度限制：固定长度为 22。

模式：`cert-([0-9a-f]{17})`

必需：否

ProfileId

本地或合作伙伴 AS2 配置文件的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`p-([0-9a-f]{17})`

必需：否

ProfileType

指示是否仅列出 LOCAL 类型配置文件或仅列出 PARTNER 类型配置文件。如果请求中未提供，则该命令会列出所有类型的配置文件。

类型：字符串

有效值：`LOCAL | PARTNER`

必需：否

Tags

可用于分组和搜索配置文件的键/值对。

类型：Tag 对象数组

数组成员：最少 1 个物品。最多 50 项。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DescribedSecurityPolicy

描述您指定的安全策略的属性。有关安全策略的更多信息，请参阅使用服务器的安全策略或使用 SFTP 连接器的安全策略。

内容

SecurityPolicyName

指定安全策略的文本名称。

类型：字符串

长度限制：长度下限为 0。最大长度为 100。

模式：`Transfer[A-Za-z0-9]*SecurityPolicy-[A-Za-z0-9-]+`

必需：是

Fips

指定此策略是否启用联邦信息处理标准 (FIPS)。此参数适用于服务器和连接器安全策略。

类型：布尔值

必需：否

Protocols

列出安全策略适用的文件传输协议。

类型：字符串数组

数组成员：最少 1 个物品。最多 5 项。

有效值：`SFTP | FTPS`

必需：否

SshCiphers

列出附加到服务器或连接器的安全策略中启用的安全外壳 (SSH) 密码加密算法。此参数适用于服务器和连接器安全策略。

类型：字符串数组

长度约束：最小长度为 0。最大长度为 50。

必需：否

SshHostKeyAlgorithms

列出安全策略的主机密钥算法。

> **Note**
> 此参数仅适用于连接器的安全策略。

类型：字符串数组

长度约束：最小长度为 0。最大长度为 50。

必需：否

SshKexs

列出附加到服务器或连接器的安全策略中启用的 SSH 密钥交换 (KEX) 加密算法。此参数适用于服务器和连接器安全策略。

类型：字符串数组

长度约束：最小长度为 0。最大长度为 50。

必需：否

SshMacs

列出附加到服务器或连接器的安全策略中启用的 SSH 消息身份验证码 (MAC) 加密算法。此参数适用于服务器和连接器安全策略。

类型：字符串数组

长度约束：最小长度为 0。最大长度为 50。

必需：否

TlsCiphers

列出附加到服务器的安全策略中启用的传输层安全 (TLS) 密码加密算法。

> **Note**
>
> 此参数仅适用于服务器的安全策略。

类型：字符串数组

长度约束：最小长度为 0。最大长度为 50。

必需：否

Type

安全策略适用的资源类型，可以是服务器或连接器。

类型：字符串

有效值：`SERVER | CONNECTOR`

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS 适用于 Ruby V3 的 SDK

DescribedServer

描述指定的启用文件传输协议的服务器的属性。

内容

Arn

指定服务器的唯一 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最小长度为 20。长度上限为 1600。

模式：`arn:\S+`

必需：是

As2ServiceManagedEgressIpAddresses

此服务器的出口 IP 地址列表。这些 IP 地址仅与使用 AS2 协议的服务器相关。它们用于发送异步 mDN。

这些 IP 地址是在您创建 AS2 服务器时自动分配的。此外，如果您更新现有服务器并添加 AS2 协议，则还会分配静态 IP 地址。

类型：字符串数组

模式：`\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}`

必需：否

Certificate

指定 Certifice Manager (ACM) AWS证书的 ARN。在 `Protocols` 设置为 FTPS 时是必需的。

类型：字符串

长度约束：最小长度为 0。长度上限为 1600。

必需：否

Domain

指定用于文件传输的存储系统的域。有两个域可用：Amazon Simple Storage Service 和 Amazon Elastic File System System (Amazon EFS)。默认值为 3。

类型：字符串

有效值：`S3 | EFS`

必需：否

EndpointDetails

要为服务器配置的 Virtual Private Cloud (VPC) 端点设置。当您在 VPC 中托管端点时，您可以使端点仅可供 VPC 内的资源访问，也可以附加弹性 IP 地址并使端点可由客户端通过 Internet 访问。您 VPC 的默认安全组会自动分配到您的端点。

类型：EndpointDetails 对象

必需：否

EndpointType

定义服务器连接到的端点类型。如果您的服务器连接到 VPC 端点，则无法通过公共互联网访问您的服务器。

类型：字符串

有效值：`PUBLIC | VPC | VPC_ENDPOINT`

必需：否

HostKeyFingerprint

指定服务器主机密钥的 Base64 编码的 SHA256 指纹。该值等效于 `ssh-keygen -l -f my-new-server-key` 命令的输出。

类型：字符串

必需：否

IdentityProviderDetails

指定调用客户提供的身份验证 API 的信息。当服务器的 `IdentityProviderType` 为 `AWS_DIRECTORY_SERVICE` 或 `SERVICE_MANAGED` 时，不会填充此字段。

类型：IdentityProviderDetails 对象

必需：否

IdentityProviderType

服务器的身份验证模式。默认值为SERVICE_MANAGED，允许您在 AWS Transfer Family 服务中存储和访问用户凭证。

用于提供AWS_DIRECTORY_SERVICE对本地环境中的活动目录组 AWS Directory Service for Microsoft Active Directory 或 Microsoft Active Directory 组的访问权限，或者 AWS 使用 AD Connector 提供访问权限。此选项还要求您使用 IdentityProviderDetails 参数提供 Directory ID。

使用 API_GATEWAY 值以与您选择的身份提供者集成。API_GATEWAY 设置要求您提供 Amazon API Gateway 端点 URL 以使用 IdentityProviderDetails 参数调用身份验证。

使用该AWS_LAMBDA值可直接使用 AWS Lambda 函数作为您的身份提供商。如果选择该值，则必须在 Function 数据类型的 IdentityProviderDetails 参数中指定 Lambda 函数的 ARN。

类型：字符串

有效值：SERVICE_MANAGED | API_GATEWAY | AWS_DIRECTORY_SERVICE | AWS_LAMBDA

必需：否

LoggingRole

(IAM) 角色的亚马逊资源名称 (ARN)，允许服务器为亚马逊 S3 或 Amazon efSevents 开启亚马逊 CloudWatch 日志记录。 AWS Identity and Access Management 设置后，您可以在 CloudWatch 日志中查看用户活动。

类型：字符串

长度约束：最小长度为 0。最大长度为 2048。

模式：(|arn:.*role/\S+)

必需：否

PostAuthenticationLoginBanner

指定用户连接到服务器时要显示的字符串。此字符串在用户进行身份验证后显示。

> **Note**
>
> SFTP 协议不支持身份验证后显示横幅。

类型：字符串

长度约束：最小长度为 0。最大长度为 4096。

模式：[\x09-\x0D\x20-\x7E]*

必需：否

PreAuthenticationLoginBanner

指定用户连接到服务器时要显示的字符串。此字符串在用户进行身份验证前显示。例如，以下横幅显示有关使用系统的详细信息：

```
This system is for the use of authorized users only. Individuals using
this computer system without authority, or in excess of their authority,
are subject to having all of their activities on this system monitored
and recorded by system personnel.
```

类型：字符串

长度约束：最小长度为 0。最大长度为 4096。

模式：[\x09-\x0D\x20-\x7E]*

必需：否

ProtocolDetails

为服务器配置的协议设置。

- 要指示被动模式（适用于 FTP 和 FTPS 协议），请使用 PassiveIp 参数。输入一个点分四组的 IPv4 地址，例如防火墙、路由器或负载均衡器的外部 IP 地址。
- 要忽略当客户端尝试对您上传到 S3 桶的文件使用 SETSTAT 命令时生成的错误，请使用 SetStatOption 参数。要让 AWS Transfer Family 服务器忽略SETSTAT命令并上传文件而不必对 SFTP 客户端进行任何更改，请将该值设置为。ENABLE_NO_OP如果您将SetStatOption参数设置为ENABLE_NO_OP，Transfer Family 会生成一个到 Amazon Logs 的 CloudWatch 日志条目，以便您可以确定客户何时SETSTAT拨打电话。
- 要确定您的 AWS Transfer Family 服务器是否通过唯一的会话 ID 恢复最近协商的会话，请使用TlsSessionResumptionMode参数。
- As2Transports 指示 AS2 消息的传输方法。目前仅支持 HTTP。

类型：ProtocolDetails 对象

必需：否

Protocols

指定文件传输协议客户端可以用来连接到服务器端点的一个或多个文件传输协议。可用的协议包括：

- SFTP（Secure Shell (SSH) 文件传输协议）：通过 SSH 的文件传输
- FTPS（文件传输协议安全）：使用 TLS 加密的文件传输
- FTP（文件传输协议）：未加密的文件传输
- AS2（适用性声明 2）：用于传输结构化 business-to-business 数据

> **Note**
>
> - 如果选择FTPS，则必须选择存储在 AWS Certificate Manager (ACM) 中的证书，当客户端通过 FTPS 连接到服务器时，该证书用于识别您的服务器。
> - 如果 `Protocol` 包括 `FTP` 或 `FTPS`，则 `EndpointType` 必须为 `VPC`，且 `IdentityProviderType` 必须为 `AWS_DIRECTORY_SERVICE`、`AWS_LAMBDA` 或 `API_GATEWAY`。
> - 如果 `Protocol` 包含 `FTP`，则无法关联 `AddressAllocationIds`。
> - 如果仅将 `Protocol` 设置为 `SFTP`，则可以将 `EndpointType` 设置为 `PUBLIC`，并且可以将 `IdentityProviderType` 设置为任何支持的身份类型：`SERVICE_MANAGED`、`AWS_DIRECTORY_SERVICE`、`AWS_LAMBDA` 或 `API_GATEWAY`。
> - 如果 `Protocol` 包括 `AS2`，则 `EndpointType` 必须是 `VPC`，并且域必须是 Amazon S3。

类型：字符串数组

数组成员：最少 1 个物品。最多 4 项。

有效值：`SFTP | FTP | FTPS | AS2`

必需：否

S3StorageOptions

指定是否对您的 Amazon S3 目录的性能进行了优化。默认情况下，将禁用该功能。

默认情况下，主目录映射TYPE的值为。DIRECTORY如果启用此选项，则需要将显式设置为，HomeDirectoryMapEntryType以FILE使映射具有文件目标。

类型：S3StorageOptions 对象

必需：否

SecurityPolicyName

指定服务器安全策略的名称。

类型：字符串

长度限制：长度下限为 0。最大长度为 100。

模式：Transfer[A-Za-z0-9]*SecurityPolicy-[A-Za-z0-9-]+

必需：否

ServerId

为您实例化的服务器指定系统分配的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

必需：否

State

所描述的服务器的状况。值为 ONLINE 表示服务器可以接受作业和传输文件。State值为OFFLINE表示服务器无法执行文件传输操作。

状态STARTING和STOPPING表示服务器处于中间状态，要么无法完全响应，要么未完全脱机。值START_FAILED或STOP_FAILED可以表示错误情况。

类型：字符串

有效值：OFFLINE | ONLINE | STARTING | STOPPING | START_FAILED | STOP_FAILED

必需：否

StructuredLogDestinations

指定将服务器日志发送到日志组。

要指定日志组，必须提供现有日志组的 ARN。在这种情况下，日志组的格式如下所示：

`arn:aws:logs:region-name:amazon-account-id:log-group:log-group-name:*`

例如，`arn:aws:logs:us-east-1:111122223333:log-group:mytestgroup:*`

如果您之前为服务器指定了日志组，则可以通过在`update-server`调用中为该参数提供空值来清除该日志组，从而关闭结构化日志记录。例如：

```
update-server --server-id s-1234567890abcdef0 --structured-log-destinations
```

类型：字符串数组

数组成员：最少 0 项。最多 1 项。

长度约束：最小长度为 20。长度上限为 1600。

模式：`arn:\S+`

必需：否

Tags

指定可用于搜索和分组已分配给所描述服务器的服务器的键值对。

类型：Tag 对象数组

数组成员：最少 1 个物品。最多 50 项。

必需：否

UserCount

指定分配给您在`ServerId`中指定的服务器的用户数。

类型：整数

必需：否

WorkflowDetails

指定要分配的工作流的工作流 ID 以及用于执行工作流的执行角色。

除了要在文件完全上传时执行的工作流，`WorkflowDetails` 还可能包含在部分文件上传时执行的工作流的工作流 ID（和执行角色）。在文件仍在上传时，如果断开连接，则会发生部分上传。

类型：WorkflowDetails 对象

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS 适用于 Ruby V3 的 SDK

DescribedUser

描述指定用户的属性。

内容

Arn

为请求描述的用户指定唯一 Amazon 资源名称（ARN）。

类型：字符串

长度约束：最小长度为 20。长度上限为 1600。

模式：`arn:\S+`

必需：是

HomeDirectory

用户使用客户端登录服务器时的登录目录（文件夹）。

`HomeDirectory` 示例为 `/bucket_name/home/mydirectory`。

> **Note**
>
> `HomeDirectory` 参数仅在 `HomeDirectoryType` 设置为 `PATH` 时使用。

类型：字符串

长度限制：长度下限为 0。最大长度为 1024。

模式：`(|/.*)`

必需：否

HomeDirectoryMappings

逻辑目录映射指定哪些 Amazon S3 或 Amazon EFS 路径和密钥应对您的用户可见，以及使其对用户可见的方式。您需要指定`Entry`和`Target`对，其中 `Entry` 显示如何使路径可见，`Target` 是实际的 Amazon S3 或 Amazon EFS 路径。如果您只指定一个目标，则将按原样显示。您还必须确保

您的AWS Identity and Access Management（IAM）角色提供对Target中路径的访问权限。只有当HomeDirectoryType设置为LOGICAL时，才能设置此值。

在大多数情况下，您可以使用此值而不是会话策略将您的用户锁定到指定的主目录（“chroot”）。为此，您可以将 Entry 设置为“/”并将 Target 设置为 HomeDirectory 参数值。

类型：HomeDirectoryMapEntry 对象数组

数组成员：最少 1 个物品。最大数量为 50000 个。

必需：否

HomeDirectoryType

您希望用户在登录服务器时，用户主目录的登录目录（文件夹）的类型。如果您将其设置为PATH，则用户将在其文件传输协议客户端中原样看到 Amazon S3 存储桶或 Amazon EFS 的绝对路径。如果您将其设置为LOGICAL，则需要针对您希望如何使 Amazon S3 或 Amazon EFS 路径对用户可见，在HomeDirectoryMappings中提供映射。

> **Note**
>
> 如果HomeDirectoryType是LOGICAL，则必须使用HomeDirectoryMappings参数提供映射。另一方面，如果HomeDirectoryType是PATH，则使用HomeDirectory参数提供绝对路径。您的模板中不能同时使用HomeDirectory和HomeDirectoryMappings。

类型：字符串

有效值：PATH | LOGICAL

必需：否

Policy

适用于您的用户的会话策略，可让您跨多个用户使用相同的 AWS Identity and Access Management(IAM) 角色。此策略将用户的访问权限缩小至 Amazon S3 存储桶的一部分。可在此策略中使用的变量包括 ${Transfer:UserName}、${Transfer:HomeDirectory} 和 ${Transfer:HomeBucket}。

类型：字符串

长度约束：最小长度为 0。最大长度为 2048。

必需：否

PosixProfile

指定完整的 POSIX 身份，包括用户 ID (`Uid`)、组 ID (`Gid`) 和任何辅助组 ID (`SecondaryGids`)，用于控制用户对 Amazon Elastic File System (Amazon EFS) 文件系统的访问。POSIX 权限针对文件系统中的文件和目录设置，用于确定用户在将文件传入和传出 Amazon EFS 文件系统时获得的访问权限级别。

类型：PosixProfile 对象

必需：否

Role

控制用户对 Amazon S3 桶或 Amazon EFS 文件系统访问权限的 AWS Identity and Access Management（IAM）角色的 Amazon 资源名称（ARN）。附加到此角色的策略确定在将文件传入和传出 Amazon S3 桶或 Amazon EFS 文件系统时要为用户提供的访问权限级别。IAM 角色还应包含一个信任关系，从而允许服务器在为用户的传输请求提供服务时访问您的资源。

类型：字符串

长度约束：最小长度为 20。最大长度为 2048。

模式：`arn:.*role/\S+`

必需：否

SshPublicKeys

指定为所述用户存储的 Secure Shell (SSH) 密钥的公有密钥部分。

类型：SshPublicKey 对象数组

数组成员：最少 0 个物品。最多 5 项。

必需：否

Tags

为请求的用户指定键值对。出于各种目的，标签可用于搜索和分组用户。

类型：Tag 对象数组

数组成员：最少 1 个物品。最多 50 项。

必需：否

UserName

指定请求描述的用户的名称。用户名用于身份验证。这是您的用户登录您的服务器时将使用的字符串。

类型：字符串

长度限制：长度下限为 3。最大长度为 100。

模式：`[\w][\w@.-]{2,99}`

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

DescribedWorkflow

描述指定工作流程的属性

内容

Arn

指定工作流程的唯一 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最小长度为 20。长度上限为 1600。

模式：`arn:\S+`

必需：是

Description

指定工作流的文本描述。

类型：字符串

长度约束：最小长度为 0。最大长度为 256。

模式：`[\w- ]*`

必需：否

OnExceptionSteps

指定在工作流执行期间遇到错误时要采取的步骤（措施）。

类型：WorkflowStep 对象数组

数组成员：最少 0 个物品。最多 8 项。

必需：否

Steps

指定所指定工作流中步骤的详细信息。

类型：WorkflowStep 对象数组

数组成员：最少 0 个物品。最多 8 项。

必需：否

Tags

可用于分组和搜索工作流的键值对。标签是出于任何目的附加到工作流的元数据。

类型：Tag 对象数组

数组成员：最少 1 个物品。最多 50 项。

必需：否

WorkflowId

工作流的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`w-([a-z0-9]{17})`

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

EfsFileLocation

为工作流程中使用的文件的位置详细信息。仅当您使用 Amazon Elastic File System (Amazon EFS) 进行存储时才适用。

目录

FileSystemId

由 Amazon EFS 分配的文件系统标识符。

类型：字符串

长度限制：最小长度为 0。最大长度为 128。

模式：`(arn:aws[-a-z]*:elasticfilesystem:[0-9a-z-:]+:(access-point/fsap|file-system/fs)-[0-9a-f]{8,40}|fs(ap)?-[0-9a-f]{8,40})`

必需：否

Path

工作流程正在使用的文件夹的路径名。

类型：字符串

长度限制：最小长度为 1。最大长度为 65536。

模式：`[^\x00]+`

必需：否

另请参阅

有关在特定语言的 AWS 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

EndpointDetails

要为文件传输协议服务器配置的 Virtual Private Cloud (VPC) 端点设置。使用 VPC 终端节点，您可以将对服务器和资源的访问权限限制在 VPC 内。要控制传入的互联网流量，请调用 `UpdateServer` API 并将弹性 IP 地址连接到服务器端点。

> **Note**
>
> 2021 年 5 月 19 日之后，如果您在 2021 年 5 月 19 日之前尚未使用 `EndpointType=VPC_ENDPOINT` 在您的 AWS 账户中创建服务器，则您将无法创建服务器。如果您在 2021 年 5 月 19 日当天或之前已经在 `EndpointType=VPC_ENDPOINT` 账户中使用 AWS 创建了服务器，则不会受到影响。在此日期之后，使用 `EndpointType = VPC`。
> 有关更多信息，请参阅 停止使用 VPC_ENDPOINT。

内容

AddressAllocationIds

将弹性 IP 地址连接到服务器端点所需的地址分配 ID 列表。

地址分配 ID 对应于弹性 IP 地址的分配 ID。此值可以从 Amazon EC2 地址数据类型的`allocationId`字段中检索。检索此值的一种方法是调用 EC2 DescribeAddressesAPI。

此参数为可选的。如果您想将 VPC 终端节点设为公开，请设置此参数。有关详细信息，请参阅为您的服务器创建面向 Internet 的终端节点。

> **Note**
>
> 只能按以下方式设置此属性：
>
> - `EndpointType`必须设置为 `VPC`
> - Transfer Family 服务器必须处于离线状态。
> - 您无法为使用 FTP 协议的 Transfer Family 服务器设置此参数。
> - 服务器必须已`SubnetIds`填充（`SubnetIds`且`AddressAllocationIds`不能同时更新）。
> - `AddressAllocationIds`不能包含重复项，且长度必须等于。`SubnetIds`例如，如果您有三个子网 ID，则还必须指定三个地址分配 ID。

- 调用 UpdateServer API 来设置或更改此参数。

类型：字符串数组

必需：否

SecurityGroupIds

可连接到服务器端点的安全组 ID 的列表。

> **Note**
>
> 此属性只能在 EndpointType 设置为 VPC 时使用。
> 只有在将SecurityGroupIds属性EndpointType从PUBLIC或VPC_ENDPOINT更改为时，才能在 UpdateServerAPI 中编辑该属性VPC。要在创建服务器的 VPC 终端节点后更改与其关联的安全组，请使用 Amazon EC2 ModifyVpcEndpointAPI。

类型：字符串数组

长度限制：最小长度为 11。长度上限为 20。

模式：sg-[0-9a-f]{8,17}

必需：否

SubnetIds

在 VPC 中托管服务器端点所需的子网 ID 的列表。

> **Note**
>
> 此属性只能在 EndpointType 设置为 VPC 时使用。

类型：字符串数组

必需：否

VpcEndpointId

VPC 端点的标识符。

> **Note**
>
> 此属性只能在 EndpointType 设置为 VPC_ENDPOINT 时使用。
> 有关更多信息，请参阅 停止使用 VPC_ENDPOINT。

类型：字符串

长度限制：固定长度为 22。

模式：vpce-[0-9a-f]{17}

必需：否

VpcId

服务器端点所在的 VPC 的 VPC 标识符。

> **Note**
>
> 此属性只能在 EndpointType 设置为 VPC 时使用。

类型：字符串

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ExecutionError

为执行工作流程期间出现的错误指定错误消息和类型。

目录

Message

指定与 ErrorType 对应的描述性消息。

类型：字符串

必需：是

Type

指定错误类型。

- ALREADY_EXISTS：如果未选择覆盖选项且目标位置中已存在同名文件，则在复制步骤中发生。
- BAD_REQUEST：一般的错误请求：例如，尝试标记 EFS 文件的步骤会返回 BAD_REQUEST，因为只能标记 S3 文件。
- CUSTOM_STEP_FAILED：会在自定义步骤提供了指示失败的回调时出现。
- INTERNAL_SERVER_ERROR：一种包罗万象的错误，可能由于多个原因引发。
- NOT_FOUND：当请求的实体（例如复制步骤的源文件）不存在时发生。
- PERMISSION_DENIED：如果您的策略不包含完成工作流程中一个或多个步骤的正确权限，则会出现。
- TIMEOUT：在执行超时的时候发生。

  > **Note**
  >
  > 您可以将 TimeoutSeconds 设置为自定义步骤，介于 1 秒到 1800 秒（30 分钟）之间。

- THROTTLED：如果您超过了每秒一个工作流程的新执行重新填充速率，则会出现。

类型：字符串

有效值：PERMISSION_DENIED | CUSTOM_STEP_FAILED | THROTTLED | ALREADY_EXISTS | NOT_FOUND | BAD_REQUEST | TIMEOUT | INTERNAL_SERVER_ERROR

必需：是

另请参阅

有关在特定语言的 AWS 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ExecutionResults

指定工作流程中的步骤，以及在工作流程执行过程中出现任何错误时要执行的步骤。

目录

OnExceptionSteps

指定在工作流执行期间遇到错误时要采取的步骤（措施）。

类型：ExecutionStepResult 对象数组

数组成员：最少 1 项。最多 50 项。

必需：否

Steps

指定所指定工作流中步骤的详细信息。

类型：ExecutionStepResult 对象数组

数组成员：最少 1 项。最多 50 项。

必需：否

另请参阅

有关在特定语言的 AWS 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ExecutionStepResult

为步骤指定以下详细信息：错误（如果有）、输出（如果有）和步骤类型。

目录

Error

如果错误是在执行指定工作流程步骤期间发生的，则指定该错误的详细信息。

类型：ExecutionError 对象

必需：否

Outputs

作为标签应用于文件的键/值对的值。仅当步骤类型为 TAG 时才适用。

类型：字符串

长度限制：最小长度为 0。最大长度为 65536。

必需：否

StepType

可用步骤类型之一。

- **COPY** – 将文件复制到另一个位置。
- **CUSTOM** – 使用 AWS Lambda 函数目标执行自定义步骤。
- **DECRYPT** – 解密上传前加密的文件。
- **DELETE** – 删除文件。
- **TAG** – 向文件添加标签。

类型：字符串

有效值：`COPY | CUSTOM | TAG | DELETE | DECRYPT`

必需：否

另请参阅

有关在特定语言的 AWS 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

FileLocation

指定要在步骤中使用的 Amazon S3 或 EFS 文件详细信息。

目录

EfsFileLocation

指定 Amazon EFS 标识符和正在使用的文件的路径。

类型：EfsFileLocation 对象

必需：否

S3FileLocation

为正在使用的文件指定 S3 详细信息，例如存储桶、ETag 等。

类型：S3FileLocation 对象

必需：否

另请参阅

有关在特定语言的 AWS 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

HomeDirectoryMapEntry

表示包含 HomeDirectoryMappings 的条目和目标的对象。

以下是chroot的Entry 和 Target对示例。

```
[ { "Entry": "/", "Target": "/bucket_name/home/mydirectory" } ]
```

内容

Entry

表示 HomeDirectoryMappings 的条目。

类型：字符串

长度限制：长度下限为 0。最大长度为 1024。

模式：/.*

必需：是

Target

表示 HomeDirectoryMapEntry 中使用的映射目标。

类型：字符串

长度限制：长度下限为 0。最大长度为 1024。

模式：/.*

必需：是

Type

指定映射的类型。FILE如果希望映射指向文件，或者DIRECTORY让目录指向目录，则将类型设置为。

> **Note**
>
> 默认情况下，创建 Transfer Family 服务器DIRECTORY时，主目录映射的映射为。TypeFILE如果您Type希望映射具有文件目标，则需要明确设置为。

类型：字符串

有效值：`FILE | DIRECTORY`

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

IdentityProviderDetails

返回与启用文件传输协议的服务器用户使用的用户身份验证类型相关的信息。一个服务器只能有一个身份验证方法。

内容

DirectoryId

您要用作身份提供商的AWS Directory Service目录的标识符。

类型：字符串

长度限制：固定长度为 12。

模式：`d-[0-9a-f]{10}`

必需：否

Function

用于身份提供商的 lambda 函数的 ARN。

类型：字符串

长度限制：最小长度为 1。长度上限为 170。

模式：`arn:[a-z-]+:lambda:.*`

必需：否

InvocationRole

此参数仅在您的`IdentityProviderType`是`API_GATEWAY`时才适用。提供了用于验证用户账户身份的 `InvocationRole` 的类型。

类型：字符串

长度约束：最小长度为 20。最大长度为 2048。

模式：`arn:.*role/\S+`

必需：否

SftpAuthenticationMethods

对于启用 SFTP 的服务器以及仅限自定义身份提供商，您可以指定是使用密码、SSH 密钥对还是两者兼而有之进行身份验证。

- PASSWORD — 用户必须提供密码才能连接。
- PUBLIC_KEY — 用户必须提供私有密钥才能连接。
- PUBLIC_KEY_OR_PASSWORD — 用户可以使用自己的密码或密钥进行身份验证。这是默认值。
- PUBLIC_KEY_AND_PASSWORD — 用户必须同时提供私有密钥和密码才能连接。服务器首先检查密钥，如果密钥有效，系统会提示输入密码。如果提供的私有密钥与存储的公有密钥不匹配，则身份验证失败。

类型：字符串

有效值：PASSWORD | PUBLIC_KEY | PUBLIC_KEY_OR_PASSWORD | PUBLIC_KEY_AND_PASSWORD

必需：否

Url

提供用于验证用户身份的服务端点的位置。

类型：字符串

长度限制：长度下限为 0。最大长度为 255。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

InputFileLocation

指定正在处理的文件的位置。

目录

EfsFileLocation

指定正在解密的 Amazon Elastic File System (Amazon EFS) 文件的详细信息。

类型：EfsFileLocation 对象

必需：否

S3FileLocation

指定正在复制或解密的 Amazon S3 文件的详细信息。

类型：S3InputFileLocation 对象

必需：否

另请参阅

有关在特定语言的 AWS 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ListedAccess

列出一个或多个指定关联访问权限的属性。

内容

ExternalId

识别目录中特定群组所需的唯一标识符。您关联的组中的用户可以通过使用AWS Transfer Family的启用协议访问您的 Amazon S3 或 Amazon EFS 资源。如果您知道组名，则可以使用 Windows 运行以下命令来查看 SID 值 PowerShell。

```
Get-ADGroup -Filter {samAccountName -like "YourGroupName*"} -Properties
* | Select SamAccountName,ObjectSid
```

在该命令中，替换为您的 YourGroupNameActive Directory 组的名称。

用于验证此参数的正则表达式是由不带空格的大写和小写字母数字字符组成的字符串。此外，还可以包括下划线或以下任何字符：=、.@：/-

类型：字符串

长度限制：最小长度为 1。最大长度为 256。

模式：`S-1-[\d-]+`

必需：否

HomeDirectory

用户使用客户端登录服务器时的登录目录（文件夹）。

`HomeDirectory` 示例为 `/bucket_name/home/mydirectory`。

> **Note**
>
> `HomeDirectory` 参数仅在 `HomeDirectoryType` 设置为 `PATH` 时使用。

类型：字符串

长度限制：长度下限为 0。最大长度为 1024。

模式：(|/.*)

必需：否

HomeDirectoryType

您希望用户在登录服务器时，用户主目录的登录目录（文件夹）的类型。如果您将其设置为PATH，则用户将在其文件传输协议客户端中原样看到 Amazon S3 存储桶或 Amazon EFS 的绝对路径。如果您将其设置为LOGICAL，则需要针对您希望如何使 Amazon S3 或 Amazon EFS 路径对用户可见，在HomeDirectoryMappings中提供映射。

> **Note**
>
> 如果HomeDirectoryType是LOGICAL，则必须使用HomeDirectoryMappings参数提供映射。另一方面，如果HomeDirectoryType是PATH，则使用HomeDirectory参数提供绝对路径。您的模板中不能同时使用HomeDirectory和HomeDirectoryMappings。

类型：字符串

有效值：PATH | LOGICAL

必需：否

Role

控制用户对 Amazon S3 桶或 Amazon EFS 文件系统访问权限的 AWS Identity and Access Management（IAM）角色的 Amazon 资源名称（ARN）。附加到此角色的策略确定在将文件传入和传出 Amazon S3 桶或 Amazon EFS 文件系统时要为用户提供的访问权限级别。IAM 角色还应包含一个信任关系，从而允许服务器在为用户的传输请求提供服务时访问您的资源。

类型：字符串

长度约束：最小长度为 20。最大长度为 2048。

模式：arn:.*role/\S+

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ListedAgreement

描述协议属性。

内容

AgreementId

协议的唯一标识符。创建协议时会返回此标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`a-([0-9a-f]{17})`

必需：否

Arn

指定协议的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最小长度为 20。长度上限为 1600。

模式：`arn:\S+`

必需：否

Description

协议的当前描述。您可以通过调用`UpdateAgreement`操作并提供新的描述来对其进行更改。

类型：字符串

长度限制：最小长度为 1。最大长度为 200。

模式：`[\p{Graph}]+`

必需：否

LocalProfileId

AS2 本地配置文件的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：p-([0-9a-f]{17})

必需：否

PartnerProfileId

合作伙伴配置文件的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：p-([0-9a-f]{17})

必需：否

ServerId

协议的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

必需：否

Status

协议可以是ACTIVE 或 INACTIVE。

类型：字符串

有效值：ACTIVE | INACTIVE

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ListedCertificate

描述证书的属性。

内容

ActiveDate

指定证书何时生效的可选日期。

类型：时间戳

必需：否

Arn

指定证书的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最小长度为 20。长度上限为 1600。

模式：`arn:\S+`

必需：否

CertificateId

已导入证书的标识符数组。您可以使用此标识符来处理配置文件和合作伙伴配置文件。

类型：字符串

长度限制：固定长度为 22。

模式：`cert-([0-9a-f]{17})`

必需：否

Description

用于识别证书的名称或简短描述。

类型：字符串

长度限制：最小长度为 1。最大长度为 200。

模式：[\p{Graph}]+

必需：否

InactiveDate

指定证书何时失效的可选日期。

类型：时间戳

必需：否

Status

证书可以是 ACTIVE、PENDING_ROTATION、或 INACTIVE。PENDING_ROTATION 意味着此证书将在当前证书到期时替换当前证书。

类型：字符串

有效值：ACTIVE | PENDING_ROTATION | INACTIVE

必需：否

Type

证书的类型。如果为该证书指定了私有密钥，则其类型为 CERTIFICATE_WITH_PRIVATE_KEY。如果没有私有密钥，则类型为 CERTIFICATE。

类型：字符串

有效值：CERTIFICATE | CERTIFICATE_WITH_PRIVATE_KEY

必需：否

Usage

指定如何使用此证书。它可以通过以下方式使用：

- SIGNING: 用于对 AS2 消息进行签名
- ENCRYPTION: 用于加密 AS2 消息
- TLS: 用于保护通过 HTTPS 发送的 AS2 通信

类型：字符串

有效值：SIGNING | ENCRYPTION

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS 适用于 Ruby V3 的 SDK

ListedConnector

返回指定连接器的详细信息。

内容

Arn

指定连接器的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最小长度为 20。长度上限为 1600。

模式：`arn:\S+`

必需：否

ConnectorId

连接器的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`c-([0-9a-f]{17})`

必需：否

Url

合作伙伴的 AS2 或 SFTP 端点的 URL。

类型：字符串

长度限制：长度下限为 0。最大长度为 255。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ListedExecution

返回指定执行的属性。

目录

ExecutionId

用于执行工作流程的唯一标识符。

类型：字符串

长度限制：固定长度为 36。

模式：`[0-9a-fA-F]{8}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{4}\-[0-9a-fA-F]{12}`

必需：否

InitialFileLocation

说明 Amazon S3 或 EFS 文件位置的结构。这是执行开始时的文件位置：如果正在复制文件，则这是初始（而非目标）文件位置。

类型：FileLocation 对象

必需：否

ServiceMetadata

与工作流程关联的、会话详细信息的容器对象。

类型：ServiceMetadata 对象

必需：否

Status

执行之一的状态。可能正在进行中、已完成、遇到异常或者正在处理异常。

类型：字符串

有效值：`IN_PROGRESS | COMPLETED | EXCEPTION | HANDLING_EXCEPTION`

必需：否

另请参阅

有关在特定语言的 AWS 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ListedHostKey

返回指定执行的主机密钥属性。

内容

Arn

主机密钥的唯一 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最小长度为 20。长度上限为 1600。

模式：`arn:\S+`

必需：是

DateImported

将主机密钥添加至服务器的日期。

类型：时间戳

必需：否

Description

主机密钥的当前描述。您可以通过调用UpdateHostKey操作并提供新的描述来对其进行更改。

类型：字符串

长度限制：长度下限为 0。最大长度为 200。

模式：`[\p{Print}]*`

必需：否

Fingerprint

公钥指纹，仅一小段字节序列，用于识别较长的公钥。

类型：字符串

必需：否

HostKeyId

的唯一标识符。

类型：字符串

长度限制：固定长度为 25。

模式：`hostkey-[0-9a-f]{17}`

必需：否

Type

用于加密的密钥类型。通过以下值指定 Type 参数：

- `ssh-rsa`
- `ssh-ed25519`
- `ecdsa-sha2-nistp256`
- `ecdsa-sha2-nistp384`
- `ecdsa-sha2-nistp521`

类型：字符串

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ListedProfile

返回指定配置文件的属性。

内容

Arn

指定配置文件的 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最小长度为 20。长度上限为 1600。

模式：`arn:\S+`

必需：否

As2Id

`As2Id` 是 AS2-name，如 RFC 4130 中所定义。对于入站传输，这是合作伙伴发送的 AS2 消息的 `AS2-From` 标头。对于出站连接器，这是使用 `AS2-To` API 操作发送给合作伙伴的 AS2 消息的 `StartFileTransfer` 标头。此 ID 不能包含空格。

类型：字符串

长度限制：长度下限为 1。长度上限为 128。

模式：`[\p{Print}\s]*`

必需：否

ProfileId

本地或合作伙伴 AS2 配置文件的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`p-([0-9a-f]{17})`

必需：否

ProfileType

指示是否仅列出 LOCAL 类型配置文件或仅列出 PARTNER 类型配置文件。如果请求中未提供，则该命令会列出所有类型的配置文件。

类型：字符串

有效值：LOCAL | PARTNER

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ListedServer

返回指定的启用文件传输协议的服务器的属性。

内容

Arn

为要列出的服务器指定唯一 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最小长度为 20。长度上限为 1600。

模式：`arn:\S+`

必需：是

Domain

指定用于文件传输的存储系统的域。有两个域可用：Amazon Simple Storage Service 和 Amazon Elastic File System System (Amazon EFS)。默认值为 3。

类型：字符串

有效值：`S3 | EFS`

必需：否

EndpointType

指定服务器连接到的 VPC 端点类型。如果您的服务器连接到 VPC 端点，则无法通过公共互联网访问您的服务器。

类型：字符串

有效值：`PUBLIC | VPC | VPC_ENDPOINT`

必需：否

IdentityProviderType

服务器的身份验证模式。默认值为`SERVICE_MANAGED`，允许您在 AWS Transfer Family 服务中存储和访问用户凭证。

用于提供AWS_DIRECTORY_SERVICE对本地环境中的活动目录组 AWS Directory Service for Microsoft Active Directory 或 Microsoft Active Directory 组的访问权限，或者 AWS 使用 AD Connector 提供访问权限。此选项还要求您使用 IdentityProviderDetails 参数提供 Directory ID。

使用 API_GATEWAY 值以与您选择的身份提供者集成。API_GATEWAY 设置要求您提供 Amazon API Gateway 端点 URL 以使用 IdentityProviderDetails 参数调用身份验证。

使用该AWS_LAMBDA值可直接使用 AWS Lambda 函数作为您的身份提供商。如果选择该值，则必须在 Function 数据类型的 IdentityProviderDetails 参数中指定 Lambda 函数的 ARN。

类型：字符串

有效值：SERVICE_MANAGED | API_GATEWAY | AWS_DIRECTORY_SERVICE | AWS_LAMBDA

必需：否

LoggingRole

(IAM) 角色的亚马逊资源名称 (ARN)，允许服务器为亚马逊 S3 或 Amazon efSevents 开启亚马逊 CloudWatch 日志记录。 AWS Identity and Access Management 设置后，您可以在 CloudWatch 日志中查看用户活动。

类型：字符串

长度约束：最小长度为 20。最大长度为 2048。

模式：arn:.*role/\S+

必需：否

ServerId

为列出的服务器指定系统分配的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：s-([0-9a-f]{17})

必需：否

State

所描述的服务器的状况。值为 ONLINE 表示服务器可以接受作业和传输文件。State值为OFFLINE表示服务器无法执行文件传输操作。

状态STARTING和STOPPING表示服务器处于中间状态，要么无法完全响应，要么未完全脱机。值START_FAILED或STOP_FAILED可以表示错误情况。

类型：字符串

有效值：OFFLINE | ONLINE | STARTING | STOPPING | START_FAILED | STOP_FAILED

必需：否

UserCount

指定分配给您在ServerId中指定的服务器的用户数。

类型：整数

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- AWS 适用于 C++ 的 SDK
- AWS 适用于 Go 的 SDK
- AWS 适用于 Java 的 SDK V2
- AWS 适用于 Ruby V3 的 SDK

ListedUser

返回您指定的用户的属性。

内容

Arn

为您想要了解的用户提供唯一 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最小长度为 20。长度上限为 1600。

模式：`arn:\S+`

必需：是

HomeDirectory

用户使用客户端登录服务器时的登录目录（文件夹）。

`HomeDirectory` 示例为 `/bucket_name/home/mydirectory`。

> **Note**
>
> `HomeDirectory` 参数仅在 `HomeDirectoryType` 设置为 `PATH` 时使用。

类型：字符串

长度限制：长度下限为 0。最大长度为 1024。

模式：`(|/.*)`

必需：否

HomeDirectoryType

您希望用户在登录服务器时，用户主目录的登录目录（文件夹）的类型。如果您将其设置为`PATH`，则用户将在其文件传输协议客户端中原样看到 Amazon S3 存储桶或 Amazon EFS 的绝对路径。如果您将其设置为`LOGICAL`，则需要针对您希望如何使 Amazon S3 或 Amazon EFS 路径对用户可见，在`HomeDirectoryMappings`中提供映射。

> **Note**
>
> 如果HomeDirectoryType是LOGICAL，则必须使用HomeDirectoryMappings参数提供映射。另一方面，如果HomeDirectoryType是PATH，则使用HomeDirectory参数提供绝对路径。您的模板中不能同时使用HomeDirectory和HomeDirectoryMappings。

类型：字符串

有效值：`PATH | LOGICAL`

必需：否

Role

控制用户对 Amazon S3 桶或 Amazon EFS 文件系统访问权限的 AWS Identity and Access Management（IAM）角色的 Amazon 资源名称（ARN）。附加到此角色的策略确定在将文件传入和传出 Amazon S3 桶或 Amazon EFS 文件系统时要为用户提供的访问权限级别。IAM 角色还应包含一个信任关系，从而允许服务器在为用户的传输请求提供服务时访问您的资源。

> **Note**
>
> 控制用户对 Amazon S3 存储桶访问权限的 IAM 角色Domain=S3，控制用户对 Amazon S3 桶访问权限的Domain=EFS。
> 附加到此角色的策略确定在将文件传入和传出 S3 桶或 EFS 文件系统时要为用户提供的访问权限级别。

类型：字符串

长度约束：最小长度为 20。最大长度为 2048。

模式：`arn:.*role/\S+`

必需：否

SshPublicKeyCount

指定为您指定的用户存储的 SSH 公有密钥的数量。

类型：整数

必需：否

UserName

指定已指定 ARN 的用户的名称。用户名用于身份验证。

类型：字符串

长度限制：长度下限为 3。最大长度为 100。

模式：`[\w][\w@.-]{2,99}`

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ListedWorkflow

包含工作流程的标识符、文本描述和 Amazon 资源名称 (ARN)。

内容

Arn

指定工作流程的唯一 Amazon 资源名称 (ARN)。

类型：字符串

长度约束：最小长度为 20。长度上限为 1600。

模式：`arn:\S+`

必需：否

Description

指定工作流的文本描述。

类型：字符串

长度约束：最小长度为 0。最大长度为 256。

模式：`[\w- ]*`

必需：否

WorkflowId

工作流的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`w-([a-z0-9]{17})`

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

LoggingConfiguration

由日志记录角色和日志组名称组成。

内容

LoggingRole

(IAM) 角色的亚马逊资源名称 (ARN)，允许服务器为亚马逊 S3 或 Amazon efSevents 开启亚马逊 CloudWatch 日志记录。AWS Identity and Access Management设置后，您可以在 CloudWatch 日志中查看用户活动。

类型：字符串

长度约束：最小长度为 20。最大长度为 2048。

模式：`arn:.*role/\S+`

必需：否

LogGroupName

此工作流所属AWS Transfer Family服务器的 CloudWatch 日志组的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 512。

模式：`[\.\-_/#A-Za-z0-9]*`

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

PosixProfile

完整的 POSIX 身份，包括用户 ID (`Uid`)、组 ID (`Gid`) 和任何辅助组 ID (`SecondaryGids`)，用于控制用户对 Amazon EFS 文件系统的访问。POSIX 权限针对文件系统中的文件和目录设置，用于确定用户在将文件传入和传出 Amazon EFS 文件系统时获得的访问权限级别。

目录

Gid

此用户用于所有 EFS 操作的 POSIX 组 ID。

类型：长整型

有效范围：最小值为 0。最大值为 4294967295。

必需：是

Uid

此用户用于所有 EFS 操作的 POSIX 用户 ID。

类型：长整型

有效范围：最小值为 0。最大值为 4294967295。

必需：是

SecondaryGids

此用户用于所有 EFS 操作的辅助 POSIX 组 ID。

类型：长数组

数组成员：最少 0 项。最多 16 项。

有效范围：最小值为 0。最大值为 4294967295。

必需：否

另请参阅

有关在特定语言的 AWS 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ProtocolDetails

为服务器配置的协议设置。

目录

As2Transports

指示 AS2 消息的传输方法。目前仅支持 HTTP。

类型：字符串数组

数组成员：固定数量为 1 项。

有效值：HTTP

必需：否

PassiveIp

指示 FTP 和 FTPS 协议的被动模式。输入一个 IPv4 地址，例如防火墙、路由器或负载均衡器的公有 IP 地址。例如：

```
aws transfer update-server --protocol-details PassiveIp=0.0.0.0
```

将上面示例中的 `0.0.0.0` 替换为要使用的实际 IP 地址。

> **Note**
>
> 如果更改 `PassiveIp` 值，则必须停止然后重新启动 Transfer Family 服务器，以使更改生效。有关在 NAT 环境中使用被动模式（PASV）的详细信息，请参阅使用 AWS Transfer Family 在防火墙或 NNAT 背后配置 FTPS 服务器。

特殊值

`AUTO` 和 `0.0.0.0` 是 `PassiveIp` 参数的特殊值。值 `PassiveIp=AUTO` 默认分配给 FTP 和 FTPS 类型的服务器。在这种情况下，服务器会自动使用 PASV 响应中的端点 IP 之一进行响应。`PassiveIp=0.0.0.0` 对其用法有更独特的应用。例如，如果您有一个高可用性（HA）网络负载均衡器（NLB）环境，其中有 3 个子网，则只能使用 `PassiveIp` 参数指定单个 IP 地址。这可降低高可用性的有效性。在这种情况下，可以指定 `PassiveIp=0.0.0.0`。这将告知客户端使用与控制连接相同的 IP 地址，并利用所有可用区进行连接。但请注意，并非所有 FTP 客户端都支持

PassiveIp=0.0.0.0 响应。FileZilla 和 WinSCP 支持此响应。如果您使用其他客户端，请检查您的客户端是否支持 PassiveIp=0.0.0.0 响应。

类型：字符串

长度约束：最小长度为 0。最大长度为 15。

必需：否

SetStatOption

使用 SetStatOption 忽略当客户端尝试对您正在上传到 S3 桶的文件使用 SETSTAT 时生成的错误。

一些 SFTP 文件传输客户端可以在上传文件时尝试使用命令（例如 SETSTAT）更改远程文件的属性，包括时间戳和权限。但是，这些命令与 Amazon S3 等对象存储系统不兼容。由于这种不兼容性，即使文件以其他方式成功上传，从这些客户端上传文件也可能导致错误。

将该值设置为 ENABLE_NO_OP 以使 Transfer Family 服务器忽略 SETSTAT 命令，并上传文件而无需对您的 SFTP 客户端进行任何更改。虽然 SetStatOption ENABLE_NO_OP 设置会忽略错误，但它确实在 Amazon CloudWatch Logs 中生成一个日志条目，因此您可以确定客户端何时进行 SETSTAT 调用。

> **Note**
>
> 如果要保留文件的原始时间戳，并使用 SETSTAT 修改其他文件属性，您可以将 Amazon EFS 用作具有 Transfer Family 的后端存储。

类型：字符串

有效值：DEFAULT | ENABLE_NO_OP

必需：否

TlsSessionResumptionMode

与使用 FTPS 协议的 Transfer Family 服务器一起使用的属性。TLS 会话恢复提供一种机制来恢复或共享 FTPS 会话的控制和数据连接之间协商的密钥。TlsSessionResumptionMode 确定服务器是否通过唯一的会话 ID 恢复最近协商的会话。此属性在 CreateServer 和 UpdateServer 调用期间可用。如果在 CreateServer 期间未指定 TlsSessionResumptionMode 值，则默认设置为 ENFORCED。

- DISABLED：服务器不处理 TLS 会话恢复客户端请求，并为每个请求创建一个新的 TLS 会话。
- ENABLED：服务器处理并接受正在执行 TLS 会话恢复的客户端。服务器不会拒绝不执行 TLS 会话恢复客户端处理的客户端数据连接。
- ENFORCED：服务器处理并接受正在执行 TLS 会话恢复的客户端。服务器拒绝不执行 TLS 会话恢复客户端处理的客户端数据连接。在将该值设置为 ENFORCED 之前，请测试您的客户端。

> **Note**
>
> 并非所有 FTPS 客户端都执行 TLS 会话恢复。因此，如果选择强制执行 TLS 会话恢复，您将阻止来自不执行协议协商的 FTPS 客户端的任何连接。要确定是否可以使用 ENFORCED 值，需要测试客户端。

类型：字符串

有效值：DISABLED | ENABLED | ENFORCED

必需：否

另请参阅

有关在特定语言的 AWS 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

S3FileLocation

为工作流程中使用的文件的位置详细信息。仅在使用 S3 存储时适用。

目录

Bucket

指定包含所有文件的 S3 存储桶。

类型：字符串

长度限制：最小长度为 3。长度上限为 63。

模式：`[a-z0-9][\.\-a-z0-9]{1,61}[a-z0-9]`

必需：否

Etag

实体标签是对象的哈希。ETag 仅反映对对象的内容的更改，而不反映对对象的元数据的更改。

类型：字符串

长度限制：最小长度为 1。最大长度为 65536。

模式：`.+`

必需：否

Key

作业创建时所分配的名称。您可以使用对象键检索该对象。

类型：字符串

长度约束：最小长度为 0。长度上限为 1024。

模式：`[\P{M}\p{M}]*`

必需：否

VersionId

指定文件版本。

类型：字符串

长度限制：最小长度为 1。长度上限为 1024。

模式：`.+`

必需：否

另请参阅

有关在特定语言的 AWS 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

S3InputFileLocation

指定客户所输入的 Amazon S3 文件位置。如果在内部使用 `copyStepDetails.DestinationFileLocation`，则它应该是 S3 复制目标。

您需要提供存储桶和密钥。密钥可以表示路径或文件。这取决于键值是否以正斜杠 (/) 的字符结尾。如果最后一个字符是 “/”，则您的文件将被复制到此文件夹，并且其名称不会更改。相反，如果最后一个字符是字母数字，则您上传的文件将被重命名为路径值。在这种情况下，如果已存在同名文件，则该文件将被覆盖。

例如，如果您的路径是 `shared-files/bob/`，则您上传的文件将被复制到 `shared-files/bob/`文件夹。如果您的路径是 `shared-files/today`，则每个上传的文件都将复制到 `shared-files` 文件夹并命名为`today`：每次上传都会覆盖先前版本的 bob 文件。

目录

Bucket

为客户输入文件指定 S3 存储桶。

类型：字符串

长度限制：最小长度为 3。长度上限为 63。

模式：`[a-z0-9][\.\-a-z0-9]{1,61}[a-z0-9]`

必需：否

Key

作业创建时所分配的名称。您可以使用对象键检索该对象。

类型：字符串

长度约束：最小长度为 0。长度上限为 1024。

模式：`[\P{M}\p{M}]*`

必需：否

另请参阅

有关在特定语言的 AWS 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

S3StorageOptions

为您的服务器配置的 Amazon S3 存储选项。

内容

DirectoryListingOptimization

指定是否对您的 Amazon S3 目录的性能进行了优化。默认情况下，将禁用该功能。

默认情况下，主目录映射TYPE的值为。DIRECTORY如果启用此选项，则需要将显式设置为，HomeDirectoryMapEntryType以FILE使映射具有文件目标。

类型：字符串

有效值：`ENABLED | DISABLED`

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

S3Tag

指定在执行标记步骤期间分配给文件的键值对。

目录

Key

已分配给服务的名称。

类型：字符串

长度限制：最小长度为 1。最大长度为 128。

模式：`([\p{L}\p{Z}\p{N}_.:/=+\-@]*)`

必需：是

Value

与密钥相对应的值。

类型：字符串

长度约束：最小长度为 0。长度上限为 256。

模式：`([\p{L}\p{Z}\p{N}_.:/=+\-@]*)`

必需：是

另请参阅

有关在特定语言的 AWS 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

ServiceMetadata

与工作流程关联的、会话详细信息的容器对象。

目录

UserDetails

服务器 ID (`ServerId`)、会话 ID (`SessionId`) 和用户 (`UserName`) 构成了 `UserDetails`。

类型：UserDetails 对象

必需：是

另请参阅

有关在特定语言的 AWS 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

SftpConnectorConfig

包含 SFTP 连接器对象详细信息。连接器对象用于在合作伙伴 SFTP 服务器之间传输文件。

> **Note**
>
> 由于SftpConnectorConfig数据类型用于创建和更新 SFTP 连接器，因此其参数TrustedHostKeys和UserSecretId被标记为非必填项。这有点误导，因为在更新现有 SFTP 连接器时它们不是必需的，而是在创建新的 SFTP 连接器时是必需的。

内容

TrustedHostKeys

一个或多个主机密钥的公共部分，用于识别您想要连接的外部服务器。您可以对 SFTP 服务器使用 ssh-keyscan 命令以检索必要的密钥。

三个标准的 SSH 公钥格式元素是<key type>、<body base64>和可选的<comment>，每个元素之间都有空格。仅指定 <key type> 和 <body base64>：请勿输入密钥的<comment>部分。

对于可信主机密钥，AWS Transfer Family 接受 RSA 和 ECDSA 密钥。

- 对于 RSA 密钥，<key type> 字符串为 ssh-rsa。
- 对于 ECDSA 密钥，<key type> 字符串为ecdsa-sha2-nistp256、ecdsa-sha2-nistp384或ecdsa-sha2-nistp521，具体取决于您生成的密钥的大小。

运行此命令以检索 SFTP 服务器主机密钥，即您的 SFTP 服务器名称所在的位置。ftp.host.com

```
ssh-keyscan ftp.host.com
```

这会将公共主机密钥打印到标准输出。

```
ftp.host.com ssh-rsa AAAAB3Nza...<long-string-for-public-key
```

将此字符串复制并粘贴到create-connector命令的TrustedHostKeys字段或控制台的 “可信主机密钥” 字段中。

类型：字符串数组

数组成员：最少 1 个物品。最多 10 项。

长度限制：最小长度为 0。最大长度为 2048。

必需：否

UserSecretId

包含 SFTP 用户的私钥、密码或两者的密钥 (在 AWS Secrets Manager 中) 的标识符。该值是密钥的 Amazon Resource Name (ARN)。

类型：字符串

长度限制：最小长度为 0。最大长度为 2048。

必需：否

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

SshPublicKey

提供有关与启用文件传输协议的特定服务器（由 `ServerId` 标识）的用户账户关联的公有 Secure Shell (SSH) 密钥的信息。返回的信息包括导入密钥的日期、公有密钥内容和公有密钥 ID。用户可以在特定的服务器上存储与其用户名关联的多个 SSH 公有密钥。

目录

DateImported

指定将公有密钥添加到用户账户的日期。

类型：Timestamp

必需：是

SshPublicKeyBody

指定 `PublicKeyId` 指定的 SSH 公有密钥的内容。

AWS Transfer Family 接受 RSA、ECDSA 和 ED25519 密钥。

类型：字符串

长度约束：最小长度为 0。最大长度为 2048。

必需：是

SshPublicKeyId

指定包含公有密钥标识符的`SshPublicKeyId`参数。

类型：字符串

长度限制：固定长度为 21。

模式：`key-[0-9a-f]{17}`

必需：是

另请参阅

有关在特定语言的 AWS 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

Tag

为特定资源创建键值对。标签是可用于搜索资源并将其分组以实现各种目的的元数据。您可以将标签应用于服务器、用户和角色。一个标记键可以取多个值。例如，要出于记账目的对服务器进行分组，您可以创建一个名为Group的标签，并将值Research和Accounting分配给该组。

目录

Key

指定给您创建的标记的名称。

类型：字符串

长度约束：最小长度为 0。最大长度为 128。

必需：是

Value

包含您为创建的密钥名称分配的一个或多个值。

类型：字符串

长度约束：最小长度为 0。长度上限为 256。

必需：是

另请参阅

有关在特定语言的 AWS 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

TagStepDetails

每种步骤类型都有自己的StepDetails结构。

在执行工作流程步骤期间用于标记文件的键/值对。

目录

Name

步骤的名称，用作标识符。

类型：字符串

长度约束：最小长度为 0。最大长度为 30。

模式：[\w-]*

必需：否

SourceFileLocation

指定将哪个文件用作工作流程步骤的输入：要么是上一步的输出，要么是为工作流程最初上传的文件。

- 要使用前一个文件作为输入，请输入${previous.file}。在这种情况下，此工作流程步骤使用上一个工作流程步骤的输出文件作为输入。这是默认值。
- 要使用最初上传的文件位置作为此步骤的输入，请输入${original.file}。

类型：字符串

长度约束：最小长度为 0。长度上限为 256。

模式：\$\{(\w+.)+\w+\}

必需：否

Tags

包含 1 到 10 个键/值对的数组。

类型：S3Tag 对象数组

数组成员：最少 1 项。最多 10 项。

必需：否

另请参阅

有关在特定语言的 AWS 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

UserDetails

指定工作流程的用户名、服务器 ID 和会话 ID。

目录

ServerId

系统为传输服务器实例分配的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`s-([0-9a-f]{17})`

必需：是

UserName

标识与服务器关联的 Transfer Family 的唯一字符串。

类型：字符串

长度限制：最小长度为 3。长度上限为 100。

模式：`[\w][\w@.-]{2,99}`

必需：是

SessionId

对应于工作流程的会话的系统分配的唯一标识符。

类型：字符串

长度限制：最小长度为 3。最大长度为 32。

模式：`[\w-]*`

必需：否

另请参阅

有关在特定语言的 AWS 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

WorkflowDetail

指定要分配的工作流的工作流 ID 以及用于执行工作流的执行角色。

除了要在文件完全上传时执行的工作流，`WorkflowDetails` 还可能包含在部分文件上传时执行的工作流的工作流 ID（和执行角色）。在文件仍在上传时，如果断开连接，则会发生部分上传。

内容

ExecutionRole

包括 Transfer 可以承担的 S3、EFS 和 Lambda 操作的必要权限，以便所有工作流步骤都可以在所需资源上运行

类型：字符串

长度约束：最小长度为 20。最大长度为 2048。

模式：`arn:.*role/\S+`

必需：是

WorkflowId

工作流的唯一标识符。

类型：字符串

长度限制：固定长度为 19。

模式：`w-([a-z0-9]{17})`

必需：是

另请参阅

有关在特定语言的 AWS SDK 中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

WorkflowDetails

`WorkflowDetail` 数据类型的容器。它由触发工作流开始执行的操作使用。

目录

OnPartialUpload

文件仅部分上传时启动工作流的触发器。您可以将工作流附加到服务器，以便在存在部分上传时执行。

当会话断开连接时，如果打开文件，则会发生部分上传。

类型：WorkflowDetail 对象数组

数组成员：最少 0 项。最多 1 项。

必需：否

OnUpload

启动工作流的触发器：上传文件后，工作流开始执行。

要从服务器中删除关联的工作流，您可以提供一个空的 `OnUpload` 对象，如下例所示。

```
aws transfer update-server --server-id s-01234567890abcdef --workflow-details '{"OnUpload":[]}'
```

类型：WorkflowDetail 对象数组

数组成员：最少 0 项。最多 1 项。

必需：否

另请参阅

有关在特定语言的 AWS 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

WorkflowStep

工作流的基本构建块。

目录

CopyStepDetails

执行文件复制的步骤的详细信息。

包含以下值：

- 一个描述
- 文件复制目标的 Amazon S3 位置。
- 指示是否覆盖现有同名文件的标志。默认为 FALSE。

类型：CopyStepDetails 对象

必需：否

CustomStepDetails

调用 AWS Lambda 函数的步骤的详细信息。

包含 Lambda 函数名称、目标和超时（以秒为单位）。

类型：CustomStepDetails 对象

必需：否

DecryptStepDetails

解密加密文件的步骤的详细信息。

包含以下值：

- 描述性名称
- 用于解密源文件的 Amazon S3 或 Amazon Elastic File System (Amazon EFS)。
- 文件解密目标的 S3 或 Amazon S3 位置。
- 指示是否覆盖现有同名文件的标志。默认为 FALSE。
- 用于加密的类型。目前仅支持 PGP。

类型：DecryptStepDetails 对象

必需：否

DeleteStepDetails

删除文件的步骤的详细信息。

类型：DeleteStepDetails 对象

必需：否

TagStepDetails

创建一个或多个标签的步骤的详细信息。

可以指定一个或多个标签。每个标签由一个键值对组成。

类型：TagStepDetails 对象

必需：否

Type

目前支持以下步骤类型。

- **COPY** – 将文件复制到另一个位置。
- **CUSTOM** – 使用 AWS Lambda 函数目标执行自定义步骤。
- **DECRYPT** – 解密上传前加密的文件。
- **DELETE** – 删除文件。
- **TAG** – 向文件添加标签。

类型：字符串

有效值：COPY | CUSTOM | TAG | DELETE | DECRYPT

必需：否

另请参阅

有关在特定语言的 AWS 软件开发工具包中使用此 API 的更多信息，请参阅以下内容：

- 适用于 C++ 的 AWS SDK
- 适用于 Go 的 AWS SDK
- 适用于 Java V2 的 AWS SDK
- 适用于 Ruby V3 的 AWS SDK

提出 API 请求

除使用控制台外，您还可以使用 AWS Transfer Family API，以编程方式配置并管理服务器。本部分描述 AWS Transfer Family 操作、为身份验证进行的请求签名和错误处理。有关 Transfer Family 可用的区域和端点的信息，请参阅 AWS 一般参考 中的 AWS Transfer Family 端点和配额

> **Note**
>
> 使用 Transfer Family 开发应用程序时，您可以使用 AWS 开发工具包。适用于 Java、.Net 和 PHP 的 AWS 开发工具包包含底层的 Transfer Family API，从而简化您的编程任务。有关下载开发工具包库的信息，请参阅示例代码库。

主题

- Transfer Family 必填请求标头
- Transfer Family 请求输入和签名
- 错误响应
- 可用的库

Transfer Family 必填请求标头

本部分描述您每次向 AWS Transfer Family发送 POST 请求时必须使用的标头。您将 HTTP 标头包含在内以识别有关请求的密钥信息，包括您希望调用的操作、请求的日期以及表示您拥有请求发送者授权的信息。标头区分大小写，其次序不重要。

下例展示在 ListServers 操作中使用的标头。

```
POST / HTTP/1.1
Host: transfer.us-east-1.amazonaws.com
x-amz-target: TransferService.ListServers
x-amz-date: 20220507T012034Z
```

```
Authorization: AWS4-HMAC-SHA256 Credential=AKIDEXAMPLE/20220507/us-east-1/transfer/
aws4_request,
    SignedHeaders=content-type;host;x-amz-date;x-amz-target,
    Signature=13550350a8681c84c861aac2e5b440161c2b33a3e4f302ac680ca5b686de48de
Content-Type: application/x-amz-json-1.1
Content-Length: 17

{"MaxResults":10}
```

以下是必须包含在向 Transfer Family 发送的 POST 请求中的标头。以下所示标头以“x-amz”为开头，是AWS专属的标头。列出的其他所有标头均为 HTTP 事务中使用的普通标头。

标头	描述
`Authorization`	授权标头为必填项。格式为标准的 Sigv4 请求签名，该签名记录在 签署 AWS API 请求中。
`Content-Type`	将 `application/x-amz-json-1.1` 用作向 Transfer Family 提出的所有请求的内容类型。 `Content-Type: application/x-amz-json-1.1`
`Host`	使用主机标头指定向其发送请求的 Transfer Family 端点。例如，`transfer.us-east-1.amazonaws.com` 指美国东部（俄亥俄州）区域端点。有关对适用 Transfer Family 端点的更多信息，请参阅AWS 一般参考中的AWS Transfer Family 端点和配额。 `Host: transfer.` *`region`*`.amazonaws.com`
`x-amz-date`	您必须在 HTTP `Date` 标头或AWS `x-amz-date` 标头中提供时间戳。(部分 HTTP 客户端库文件不允许您设置`Date`标头。) 当 `x-amz-date` 标头存在时，Transfer Family 会在请求验证期间忽略任何`Date` 标头。`x-amz-date` 格式必须为 YYYYMMDD'T'HHMMSS'Z' 格式的 ISO8601 Basic。 `x-amz-date:` *`YYYYMMDD'T'HHMMSS'Z'`*

标头	描述
x-amz-target	该标头指定 API 的版本以及您要请求的操作。目标标头值通过结合 API 版本和 API 名称而形成，其格式如下。 `x-amz-target: TransferService.  operationName` 操作名称值（例如 ListServers ）可以从 API 列表 ListServers 中找到。
x-amz-security-token	当用于签署请求的凭证是临时凭证或会话凭证时，必须使用此标头（有关详细信息，请参阅 IAM 用户指南中的将临时凭证与 AWS 资源配合使用）。有关更多信息，请参阅 Amazon Web Services 一般参考 中的向 HTTP 请求添加签名。

Transfer Family 请求输入和签名

所有请求输入都必须作为请求正文中的 JSON 负载的一部分发送。对于所有请求字段均为可选字段的操作（例如 ListServers），您仍然需要在请求正文中提供一个空的 JSON 对象，例如 {}。Transfer Family 有效负载请求/响应的结构记录在现有的 API 参考中，例如 DescribeServer。

Transfer Family 支持使用 AWS 签名版本 4 进行身份验证。有关详细信息，请参阅签署 AWS API 请求。

错误响应

当存在错误时，响应头信息会包含：

- Content-Type:application/x-amz-json-1.1
- 适当的 4xx 或 5xx HTTP 状态码

错误响应的正文会包含有关错误出现的信息。下列错误响应示例显示的是所有错误响应中常见的响应元素的输出语法。

```
{
    "__type": "String",
```

```
    "Message": "String", <!-- Message is lowercase in some instances -->
    "Resource": String,
    "ResourceType": String
    "RetryAfterSeconds": String
}
```

下表介绍了前一语法中显示的 JSON 错误响应字段。

__type

Transfer Family API 调用的例外情况之一。

类型：字符串

留言或消息

一个操作错误代码消息。

> **Note**
>
> 一些例外使用 message，而另一些则使用 Message。您可以检查接口的代码以确定正确的情况。或者，您可以测试每个选项，看看哪个有效。

类型：字符串

资源

调用错误的资源。例如，如果您尝试创建已存在的用户，则 Resource 为现有用户的用户名。

类型：字符串

ResourceType

调用错误的资源类型。例如，如果您尝试创建已存在的用户，则 ResourceType 为 User。

类型：字符串

RetryAfterSeconds

重试命令之前等待的秒数。

类型：字符串

错误响应示例

如果您调用 DescribeServer API 并指定不存在的服务器，则会返回以下 JSON 正文。

```
{
  "__type": "ResourceNotFoundException",
  "Message": "Unknown server",
  "Resource": "s-11112222333344444",
  "ResourceType": "Server"
}
```

如果执行 API 导致出现节流，则返回以下 JSON 正文。

```
{
   "__type":"ThrottlingException",
   "RetryAfterSeconds":"1"
}
```

如果您使用 CreateServer API 但没有足够的权限创建 Transfer Family 服务器，则会返回以下 JSON 正文。

```
{
  "__type": "AccessDeniedException",
  "Message": "You do not have sufficient access to perform this action."
}
```

如果您使用 CreateUser API 并指定已存在的用户，则会返回以下 JSON 正文。

```
{
  "__type": "ResourceExistsException",
  "Message": "User already exists",
  "Resource": "Alejandro-Rosalez",
  "ResourceType": "User"
}
```

可用的库

对于更喜欢使用语言特定的 API 操作而不是命令行工具和 Query API 构建应用程序的软件开发人员，AWS 现在为他们提供了库、示例代码、教程和其他资源。这些库提供了一些基本功能 (未包括

API 中)，比如请求身份验证、请求重试和错误处置，以便您轻松地开始工作。请查看在 AWS 上构建的工具

有关所有语言的库和示例代码，请参阅示例代码和库。

常见参数

以下列表包含所有操作用于使用查询字符串对 Signature Version 4 请求进行签名的参数。任何特定于操作的参数都列在该操作的主题中。有关 Signature Version 4 的更多信息，请参阅《IAM 用户指南》中的签署 AWS API 请求。

Action

要执行的操作。

类型：字符串

必需：是

Version

编写请求所针对的 API 版本，格式为 YYYY-MM-DD。

类型：字符串

必需：是

X-Amz-Algorithm

您用于创建请求签名的哈希算法。

条件：当您在查询字符串中而不是 HTTP 授权标头中包括身份验证信息时，请指定此参数。

类型：字符串

有效值：AWS4-HMAC-SHA256

必需：条件

X-Amz-Credential

凭证范围值，该值是一个字符串，其中包含您的访问密钥、日期、您要定位的区域、您请求的服务以及终止字符串（“aws4_request”）。值采用以下格式表示：access_key/YYYYMMDD/region/service/aws4_request。

有关更多信息，请参阅《IAM 用户指南》中的创建已签名的 AWS API 请求。

条件：当您在查询字符串中而不是 HTTP 授权标头中包括身份验证信息时，请指定此参数。

类型：字符串

必需：条件

X-Amz-Date

用于创建签名的日期。格式必须为 ISO 8601 基本格式 (YYYYMMDD'T'HHMMSS'Z')。例如，以下日期时间是有效的 X-Amz-Date 值：`20120325T120000Z`。

条件：X-Amz-Date 对于所有请求都是可选的；它可以用于覆盖对请求签名所使用的日期。如果以 ISO 8601 基本格式指定 Date 标头，则不需要 X-Amz-Date。使用 X-Amz-Date 时，它始终会覆盖 Date 标头的值。有关更多信息，请参阅《IAM 用户指南》中的 AWS API 请求签名的元素。

类型：字符串

必需：条件

X-Amz-Security-Token

通过调用 AWS Security Token Service（AWS STS）获得的临时安全令牌。有关支持来自 AWS STS 的临时安全凭证的服务列表，请参阅《IAM 用户指南》中的使用 IAM 的 AWS 服务。

条件：如果您使用来自 AWS STS 的临时安全凭证，则必须包含安全令牌。

类型：字符串

必需：条件

X-Amz-Signature

指定从要签名的字符串和派生的签名密钥计算的十六进制编码签名。

条件：当您在查询字符串中而不是 HTTP 授权标头中包括身份验证信息时，请指定此参数。

类型：字符串

必需：条件

X-Amz-SignedHeaders

指定作为规范请求的一部分包含的所有 HTTP 标头。有关指定已签名标头的更多信息，请参阅《IAM 用户指南》中的创建已签名的 AWS API 请求。

条件：当您在查询字符串中而不是 HTTP 授权标头中包括身份验证信息时，请指定此参数。

类型：字符串

必需：条件

常见错误

本部分列出了所有 AWS 服务的常见 API 操作错误。对于特定于此服务的 API 操作的错误，请参阅该 API 操作的主题。

AccessDeniedException

您没有足够的访问权限，无法执行该操作。

HTTP 状态代码：400

IncompleteSignature

请求签名不符合 AWS 标准。

HTTP 状态代码：400

InternalFailure

由于未知错误、异常或故障，请求处理失败。

HTTP 状态代码：500

InvalidAction

所请求的操作无效。验证操作是否已正确键入。

HTTP 状态代码：400

InvalidClientTokenId

在我们的记录中没有所提供的 X.509 证书或 AWS 访问密钥 ID。

HTTP 状态代码：403

NotAuthorized

您无权执行此操作。

HTTP 状态代码：400

OptInRequired

AWS 访问密钥 ID 需要订阅服务。

HTTP 状态代码：403

RequestExpired

请求到达服务的时间超过请求上的日期戳或请求到期日期 (如针对预签名 URL) 15 分钟，或者请求上的日期戳离到期还有 15 分钟以上。

HTTP 状态代码：400

ServiceUnavailable

由于服务器发生临时故障而导致请求失败。

HTTP 状态代码：503

ThrottlingException

由于请求限制而导致请求被拒绝。

HTTP 状态代码：400

ValidationError

输入未能满足 AWS 服务指定的约束。

HTTP 状态代码：400

的文档历史记录 AWS Transfer Family

下表描述了此版本的文档 AWS Transfer Family。

- API 版本：transfer-2018-11-05
- 最新文档更新：2024 年 4 月 12 日

更改	描述	日期
能够在 AS2 消息交换中使用贸易伙伴的自签名 TLS 证书	AWS Transfer Family 增加了导入和使用贸易伙伴的公开自签名 TLS 证书的选项，用于通过 HTTPS 向其服务器发送适用性声明 2 (AS2) 消息。	2024年4月12日
为 SFTP 连接器添加了安全策略	AWS Transfer Family 添加了用于 SFTP 连接器的安全策略。有关更多信息，请参阅 AWS Transfer Family SFTP 连接器的安全策略。	2024 年 4 月 5 日
与 Amazon 集成 EventBridge	AWS Transfer Family 现在会自动将所有文件传输操作的事件发布到 Amazon EventBridge 。有关更多信息，请参阅 使用管理 Transfer Family 事件 Amazon EventBridge。	2024 年 2 月 8 日
增加了新的安全策略	AWS Transfer Family 添加了新的 FIPS 和非 FIPS 安全策略。此外，分配给服务器的默认安全策略始终是最新的安全策略。有关更多信息，请参阅 AWS Transfer Family 服务器的安全策略。	2024年2月5日

更改	描述	日期
Support 支持 SFTP 连接器和 AS2 的静态 IP 地址	Transfer Family 现在为 SFTP 连接器和 AS2 提供静态 IP 地址。这样可以与受 IP 许可名单控制保护的远程 SFTP 服务器建立连接。对于 AS2，我们将为来自 AS2 服务器的异步 MDN 响应引入静态 IP 地址。	2024 年 1 月 16 日
对用户指南进行了重新编排，使其与的最新版本更加一致。AWS Transfer Family	自该指南问世以来，Transfer Family已经增加了多项功能，因此有必要对该指南进行重组。	2024 年 1 月 3 日
逻辑目录映射增强功能 亚马逊 S3 列表性能优化	Transfer Family 现在支持最大为 2.1 MB 的逻辑目录映射。现在，您还可以声明用户映射是否指向文件。有关更多信息，请参阅 使用逻辑目录的规则。 在创建或更新使用 Amazon S3 进行存储的服务器时，您现在可以优化列出 S3 目录（或文件夹）的性能。有关更多信息，请参阅 配置 SFTP、FTPS 或 FTP 服务器端点。	2023 年 11 月 17 日
带有虚拟私有云 (VPC) 端点的 SFTP 服务器的备用端口	现在，您可以为具有 VPC 终端节点的 SFTP Transfer Family 服务器启用备用非标准端口。有关更多信息，请参阅 在虚拟私有云中创建服务器。	2023 年 11 月 17 日

更改	描述	日期
支持 SFTP 连接器	SFTP 连接器扩展了 AWS Transfer Family 与云端和本地远程服务器通信的功能。有关更多信息，请参阅 使用 SFTP 连接器发送和检索文件。	2023 年 7 月 25 日
支持 AS2 基本身份验证	Transfer Family 现在支持对使用适用性声明 2 (AS2) 协议的服务器使用基本身份验证。有关更多信息，请参阅 AS2 连接器的基本身份验证。	2023 年 6 月 30 日
支持结构化 JSON 日志记录	Transfer Family 现在支持向亚马逊 CloudWatch传送结构化 JSON 日志、将日志流分组到自定义日志组以及跨协议执行常见日志查询。有关更多信息，请参阅 Amazon CloudWatch 正在登录 AWS Transfer Family。	2023 年 6 月 24 日
支持多种身份验证方法	Transfer Family 支持使用密码、公有密钥/私有密钥对或两者进行身份验证。这适用于使用 SFTP 协议和自定义身份提供商的服务器。有关更多信息，请参阅 创建启用 SFTP 的服务器。	2023 年 5 月 17 日

更改	描述	日期
支持对 Transfer Family 通过工作流程处理的文件进行 Pretty Good Privacy (PGP) 解密。	Transfer Family 内置了对 Prety Good Privacy (PGP) 解密的支持。您可以对通过 SFTP、FTPS 或 FTP 上传到 Amazon Simple Storage Service (Amazon S3) 或 Amazon Elastic File System (Amazon EFS) 的文件使用 PGP 解密。有关更多信息，请参阅 生成和管理 PGP 密钥 和 在工作流程中使用 PGP 解密。	2022 年 12 月 21 日
Transfer Family 服务器对适用性声明 2 (AS2) 文件传输协议提供完全托管支持	您可以创建使用 AS2 协议的服务器，用于向 AWS 环境内部或外部的贸易伙伴发送和接收信息。有关更多信息，请参阅 配置 AS2。	2022 年 7 月 25 日
支持在创建服务器时显示横幅	您可以在创建服务器时添加自定义消息。您可以显示预身份验证消息（所有协议）和身份验证后消息（适用于 FTP 和 FTPS 服务器）。有关更多信息，请参阅 创建启用 SFTP 的服务器、创建启用 FTPS 的服务器 或 创建启用 FTP 的服务器。	2022 年 2 月 17 日

更改	描述	日期
AWS Lambda 作为身份提供者 Support	现在，您可以使用他们的 Tr AWS Lambda ansfer Family 服务器连接到自定义身份提供商。以前，您必须提供 Amazon API Gateway URL 才能集成自定义身份提供商。有关更多信息，请参阅 AWS Lambda 用于整合您的身份提供商。	2021 年 11 月 16 日
支持托管文件传输工作流程	托管文件传输工作流程为您提供了当前手动执行的常见任务的上传后处理抽象。有关更多信息，请参阅 AWS Transfer Family 托管工作流程。	2021 年 9 月 2 日
Support AWS Directory Service for Microsoft Active Directory	除了服务托管和自定义身份提供商之外，您现在还可以使用管理用户访问权限 AWS Directory Service for Microsoft Active Directory 以进行身份验证和授权。有关更多信息，请参阅 使用 Di AWS rectory Service 身份提供商。	2021 年 5 月 24 日
全新 AWS 区域	AWS Transfer Family 现已在非洲（开普敦）地区推出。有关对 Transfer Family 端点的更多信息，请参阅AWS 一般参考中的AWS Transfer Family 端点和限额。	2021 年 2 月 24 日

更改	描述	日期
全新 AWS 区域	AWS Transfer Family 现已在亚太地区（香港）和中东（巴林）地区推出。有关对 Transfer Family 端点的更多信息，请参阅AWS 一般参考中的AWS Transfer Family 端点和限额。	2021 年 2 月 17 日
支持 Amazon EFS 作为数据存储	Transfer Family 现在支持文件传输进出 Amazon Elastic File System (Amazon EFS)。Amazon EFS 是一个简单、可扩展、完全托管的弹性 NFS 文件系统。有关更多信息，请参阅 配置 Amazon EFS 文件系统。	2021 年 1 月 6 日
Support AWS WAF	Transfer Family 现在支持 AWS WAF Web 应用程序防火墙，可帮助保护网络应用程序和 API 操作免受攻击。有关更多信息，请参阅 是一个 Web 应用程序防火墙。。	2020 年 11 月 24 日
支持 Virtual Private Cloud (VPC) 中的多个安全组	现在，您可以将多个安全组附加到 VPC 中的一台服务器。有关更多信息，请参阅 在虚拟私有云中创建服务器。	2020 年 10 月 15 日

更改	描述	日期
全新 AWS 区域	Transfer Family 现已在各 AWS GovCloud (US) 地区推出。有关 AWS GovCloud (US) 区域的 Transfer Family 终端节点的更多信息，请参阅中的AWS Transfer Family 终端节点和配额AWS 一般参考。有关在 AWS GovCloud (US) 各地区使用 Transfer Family 的信息，请参阅AWS GovCloud (US) 用户指南AWS Transfer Family中的。	2020 年 9 月 30 日
现在可以将支持加密算法的安全策略附加到您的服务器	现在，您可以将包含一组受支持的加密算法的安全策略附加到服务器。有关更多信息，请参阅 AWS Transfer Family 服务器的安全策略。	2020 年 8 月 12 日
支持联邦信息处理标准 (FIPS) 终端节点	启用 FIPS 的端点仅在北美 AWS 区域可用。有关可用区域，请参阅AWS 一般参考中的AWS Transfer Family 端点和限额。要为启用 SFTP 的服务器端点启用 FIPS，请参阅创建启用 SFTP 的服务器。要为启用 FTPS 的服务器端点启用 FIPS，请参阅创建启用 FTPS 的服务器。要为启用 FTP 的服务器端点启用 FIPS，请参阅创建启用 FTP 的服务器。	2020 年 8 月 12 日

更改	描述	日期
用户名字符长度增加和允许的其他字符	用户名现在可以包含 at 符号 (@) 和句点 (.)，并且最大长度可以为 100 个字符。要添加用户，请参阅管理服务器端点的用户。	2020 年 8 月 12 日
支持自动创建亚马逊 CloudWatch 日志 AWS Identity and Access Management (IAM) 角色	Transfer Family 现在支持自动创建 CloudWatch 日志 IAM 角色来查看最终用户活动。有关更多信息，请参阅 创建启用 SFTP 的服务器、创建启用 FTPS 的服务器 或 创建启用 FTP 的服务器。	2020 年 7 月 30 日
AWS Transfer Family 现在支持将源 IP 作为授权因素。	Transfer Family 增加了对使用最终用户的源 IP 地址作为授权因素的支持，使您在通过安全文件传输协议 (SFTP)、SSL (FTPS) 文件传输协议或文件传输协议 (FTP) 授权访问权限时，可以应用额外的安全层。有关更多信息，请参阅 使用自定义身份提供程序。	2020 年 6 月 9 日
AWS SFTP 的传输功能现已启用， AWS Transfer Family 并增加了对 FTP 和 FTPS 的支持。	现在，您可以使用另外两个协议来传输用户的文件：安全文件传输协议 (FTPS) 和文件传输协议 (FTP)。除了现有的安全文件传输协议 (SFTP) 支持外，用户还可以在其中移动、运行 AWS、保护和集成基于 SSL 的 FTP (FTPS) 和基于纯文本 FTP 的工作流程。	2020 年 4 月 23 日

更改	描述	日期
支持 虚拟私有云 (VPC) 安全组和弹性 IP 地址	现在，您可以使用安全组为传入 IP 地址创建许可名单，从而为服务器提供额外的安全保护。您还可以将弹性 IP 地址与服务器的端点相关联。通过这样做，您可以让防火墙后面的用户允许访问该端点。有关更多信息，请参阅 在虚拟私有云中创建服务器。	2020 年 1 月 10 日
支持在 VPC 中工作	您现在可在 VPC 中创建服务器。您可以使用您的服务器通过客户端与 Amazon S3 存储桶往返传输数据而不流经公共互联网。有关更多信息，请参阅 在虚拟私有云中创建服务器。	2019 年 3 月 27 日
AWS Transfer Family 已发布的第一个版本。	此初始版本包括设置指令、描述如何入门，并提供有关客户端配置、用户配置和监控活动的信息。	2018 年 11 月 25 日

AWS 术语表

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

本文属于机器翻译版本。若本译文内容与英语原文存在差异，则一律以英文原文为准。