사용자 가이드

AWS Transfer Family

AWS Transfer Family: 사용자 가이드

Table of Contents

이게 뭐야 AWS Transfer Family?

AWS Transfer Family 스토리지 서비스 간에 파일을 전송하고 스토리지 서비스에서 파일을 전송할 수 있는 보안 전송 서비스입니다. AWS Transfer Family는 AWS 클라우드 플랫폼의 일부입니다. AWS Transfer Family SFTP, AS2, FTPS 및 FTP를 통해 Amazon S3 또는 Amazon EFS에서 직접 파일을 주고 받을 수 있도록 완벽하게 관리되는 지원을 제공합니다. 인증, 액세스 및 방화벽을 위한 기존 클라이언트 측 구성을 유지 관리하여 파일 전송 워크플로를 원활하게 마이그레이션, 자동화 및 모니터링할 수 있으므로 고객, 파트너, 내부 팀 또는 애플리케이션이 변경되지 않습니다.

Amazon Web Services에서 자세히 알아보고 클라우드 애플리케이션 구축을 시작하려면 시작하기를 참조하십시오. AWS

AWS Transfer Family 다음 스토리지 서비스로부터 또는 다음 AWS 스토리지 서비스로의 데이터 전송을 지원합니다.

- Amazon Simple Storage Service(S3) 스토리지 Amazon S3에 관한 자세한 내용은 Amazon Simple Storage Service 시작하기를 참조하세요.
- Amazon Elastic File System(Amazon EFS) Network File System(NFS) 파일 시스템 Amazon EFS에 관한 자세한 내용은 Amazon Elastic File System이란?을 참조하세요.

AWS Transfer Family 다음 프로토콜을 통한 데이터 전송을 지원합니다.

- 보안 파일 전송 프로토콜 (SFTP): 버전 3

  공식 IETF 문서는 다음과 같습니다: SSH 파일 전송 프로토콜 -02.txt. draft-ietf-secsh-filexfer
- File Transfer 프로토콜 보안(FTPS)
- File Transfer 프로토콜(FTP)
- 적용성 보고서 2(AS2)

> **Note**
>
> FTP 및 FTPS 데이터 연결의 경우 Transfer Family가 데이터 채널을 설정하는 데 사용하는 포트 범위는 8192~8200입니다.

File Transfer 프로토콜은 금융 서비스, 의료, 광고, 소매를 비롯한 다양한 산업의 데이터 교환 워크플로에서 활용됩니다. Transfer Family를 사용하면 파일 전송 워크플로우를 로 쉽게 마이그레이션할 수 AWS있습니다.

다음은 Amazon S3에서 Transfer Family를 사용하는 몇 가지 일반적인 사용 사례입니다.

- 공급업체 및 파트너와 같은 타사의 업로드를 AWS 위해 데이터가 레이크됩니다.
- 고객과의 구독 기반 데이터 배포.
- 조직 내의 내부 전송.

다음은 Amazon EFS에서 Transfer Family를 사용하는 몇 가지 일반적인 사용 사례입니다.

- 데이터 분산
- 공급망
- 콘텐츠 관리
- 웹 서비스 애플리케이션

다음은 AS2에서 Transfer Family를 사용하는 몇 가지 일반적인 사용 사례입니다.

- 프로토콜에 내장된 데이터 보호 및 보안 기능을 필요로 하는 컴플라이언스 요구 사항이 있는 워크플로
- 공급망 물류
- 결제 워크플로
- B usiness-to-business (B2B) 거래
- 전사적 자원 관리(ERP) 및 고객 관계 관리(CRM) 시스템과의 통합

Transfer Family를 사용하면 서버 인프라를 실행할 필요 AWS 없이 파일 전송 프로토콜 지원 서버에 액세스할 수 있습니다. 이 서비스를 사용하여 최종 사용자의 클라이언트 및 구성을 그대로 AWS 유지하면서 파일 전송 기반 워크플로를 마이그레이션할 수 있습니다. 먼저 호스트 이름을 서버 엔드포인트에 연결한 다음, 사용자를 추가하고 적절한 수준의 액세스를 제공해야 합니다. 이 작업이 끝나면, 사용자의 전송 요청은 Transfer Family 서버 엔드포인트에서 바로 처리됩니다.

Transfer Family는 다음과 같은 이점이 있습니다.

- 사용자의 필요에 맞게 실시간으로 확장되는 종합 관리형 서비스입니다.

- 애플리케이션을 수정하거나 File Transfer 프로토콜 인프라를 실행하지 않아도 됩니다.
- 내구성이 뛰어난 Amazon S3 스토리지에 데이터를 보관하면 처리, 분석, 보고, 감사 및 보관 기능에 AWS 서비스 네이티브를 사용할 수 있습니다.
- Amazon EFS를 데이터 스토어로 사용하면 AWS 클라우드 서비스 및 온프레미스 리소스와 함께 사용할 수 있는 완전 관리형 탄력적 파일 시스템을 확보할 수 있습니다. Amazon EFS는 애플리케이션을 중단하지 않고 페타바이트까지 온디맨드 확장 및 축소되도록 구축되어 파일을 추가하고 제거할 때마다 파일을 자동 확장 및 축소합니다. 따라서 확장을 수용하기 위해 용량을 프로비저닝하고 관리할 필요가 없습니다.
- AWS Transfer Family를 사용하여 업로드한 파일의 처리를 쉽게 설정, 실행, 자동화 및 모니터링할 수 있게 해주는 완전 관리형 서버리스 File Transfer 워크플로 서비스입니다.
- 선결제 비용이 없으며, 서비스를 사용한 만큼만 지불하면 됩니다.

다음 섹션에서는 Transfer Family의 다양한 특성에 대한 설명, 시작하기 자습서, 다양한 프로토콜 지원 서버를 설정하는 방법에 대한 자세한 지침, 다양한 타입의 ID 제공자를 사용하는 방법, 서비스의 API 참조를 확인할 수 있습니다.

Transfer Family를 시작하려면 다음을 참조하세요.

- 작동 AWS Transfer Family 원리
- 필수 조건
- 서버 엔드포인트 시작하기 AWS Transfer Family

작동 AWS Transfer Family 원리

AWS Transfer Family 다음 프로토콜을 통해 Amazon Simple Storage AWS Service (Amazon S3) 스토리지 또는 Amazon Elastic File System (Amazon EFS) 파일 시스템으로 파일을 주고받는 데 사용할 수 있는 완전 관리형 서비스입니다.

- 보안 파일 전송 프로토콜 (SFTP): 버전 3

  공식 IETF 문서는 다음과 같습니다: SSH 파일 전송 프로토콜 -02.txt. draft-ietf-secsh-filexfer
- File Transfer 프로토콜 보안(FTPS)
- File Transfer 프로토콜(FTP)
- 적용성 보고서 2(AS2)

AWS Transfer Family 최대 3개의 가용 영역을 지원하며 연결 및 전송 요청을 위한 Auto Scaling, 중복 플릿으로 뒷받침됩니다. 지연 시간 기반 라우팅을 사용하여 중복성을 높이고 네트워크 지연 시간을 최소화하는 방법에 대한 예는 블로그 게시물 SFTP 서버 전송을 통해 네트워크 지연 최소화를 참조하십시오. AWS

Transfer Family 관리형 File Transfer 워크플로(MFTW)는 완전 관리형 서버리스 File Transfer 워크플로 서비스로, AWS Transfer Family를 사용하여 업로드한 파일의 처리를 쉽게 설정, 실행, 자동화 및 모니터링할 수 있습니다. 고객은 MFTW를 사용하여 Transfer Family를 사용하여 전송되는 데이터의 복사, 태깅, 스캔, 필터링, 압축/압축 해제, 암호화/암호 해독과 같은 다양한 처리 단계를 자동화할 수 있습니다. 이를 통해 추적 및 감사에 대한 엔드 투 엔드 가시성이 제공됩니다. 자세한 내용은 AWS Transfer Family 관리형 워크플로를 참조하세요.

AWS Transfer Family 모든 표준 파일 전송 프로토콜 클라이언트를 지원합니다. 다음은 대표적인 클라이언트입니다.

- OpenSSH – Macintosh 및 Linux 명령줄 유틸리티.
- WinSCP – Windows 전용 그래픽 클라이언트.
- Cyberduck – Linux, Macintosh, Microsoft Windows 그래픽 클라이언트.
- FileZilla— 리눅스, 매킨토시, 윈도우 그래픽 클라이언트.

AWS 는 다음과 같은 Transfer Family 워크샵을 제공합니다.

- 관리형 SFTP/FTPS 엔드포인트와 사용자 관리를 AWS Transfer Family 위한 Amazon Cognito 및 DynamoDB를 활용하는 파일 전송 솔루션을 구축하십시오. 이 워크숍의 세부 정보는 여기에서 확인할 수 있습니다.
- AS2가 활성화된 Transfer Family 엔드포인트와 Transfer Family AS2 커넥터를 구축하십시오. 이 워크숍에 대한 자세한 내용은 여기에서 확인할 수 있습니다.
- 기존 애플리케이션을 수정하거나 서버 인프라를 관리할 필요 AWS 없이 확장 가능하고 안전한 파일 전송 아키텍처를 구축할 수 있는 방법에 대한 지침과 실습을 제공하는 솔루션을 구축하십시오. 이 워크숍의 세부 정보는 여기에서 확인할 수 있습니다.

Transfer Family와 관련된 블로그 게시물

다음 표에는 Transfer Family 고객에게 유용한 정보가 들어 있는 블로그 게시물이 나와 있습니다. 표는 시간 역순으로 작성되어 가장 최근 게시물이 표의 맨 앞에 오도록 되어 있습니다.

블로그 게시물 제목 및 링크	날짜
AWS Transfer Family SFTP 커넥터 및 PGP 암호화를 사용하여 안전하고 규정을 준수하는 관리형 파일 전송 설계	2024년 5월 16일
Amazon Cognito를 Amazon AWS Transfer Family S3와 함께 자격 증명 공급자로 사용	2024년 5월 14일
Transfer Family를 통해 안전하고 규정을 준수하는 관리형 파일 전송 솔루션을 구축하는 방법	2024년 1월 3일
다음을 사용하여 멀웨어 위협을 탐지합니다. AWS Transfer Family	2023년 7월 20일
다음을 통해 SAP 워크로드 확장 AWS Transfer Family	2023년 7월 13일
PGP를 사용하여 파일을 암호화 및 복호화하고 AWS Transfer Family	2023년 6월 21일
Azure 액티브 디렉터리를 통한 인증 및 AWS Transfer FamilyAWS Lambda	2022년 12월 15일
관리형 워크플로를 사용하여 AWS Transfer Family 파일 전송 알림을 사용자 지정합니다.	2022년 10월 14일
AWS Transfer Family 워크플로를 사용하여 클라우드 네이티브 File Transfer 플랫폼 구축	2022년 1월 5일
A AWS Transfer Family 및 AWS Lambda D를 사용하여 사용자 셀프 서비스 키 관리를 지원합니다.	2021년 12월 17일
Amazon AWS Transfer Family S3를 사용하여 데이터 액세스 제어를 개선하십시오.	2021년 10월 5일

블로그 게시물 제목 및 링크	날짜
AWS Global Accelerator 및 AWS Transfer Family 서비스를 사용하여 인터넷 연결 파일 전송의 처리량을 개선합니다.	2021년 6월 7일
AWS 웹 애플리케이션 방화벽 및 Amazon API Gateway를 AWS Transfer Family 통한 보안	2021년 5월 5일
AWS 웹 애플리케이션 방화벽 및 Amazon API Gateway를 AWS Transfer Family 통한 보안	2021년 1월 15일
AWS Transfer Family 아마존 Elastic File System 지원	2021년 1월 7일
AWS Transfer Family 사용을 위한 암호 인증을 활성화합니다. AWS Secrets Manager	2020년 11월 5일
및 를 사용하여 AWS Transfer Family 데이터 액세스를 중앙 집중화하십시오. AWS Storage Gateway	2020년 6월 22일
서버리스 AWS Lambda 애플리케이션에서 Amazon EFS 사용	2020년 6월 18일
IP 허용 목록을 사용하여 서버를 보호하십시오. AWS Transfer Family	2020년 4월 8일
SFTP 서버 AWS 전송으로 네트워크 지연 시간을 최소화합니다.	2020년 2월 19일
SFTP 서버를 리프트 앤 시프트로 마이그레이션할 수 있습니다. AWS	2020년 2월 12일
chroot 및 논리적 디렉토리를 AWS 사용하여 SFTP 구조를 단순화하십시오.	2019년 9월 26일
Okta를 아이덴티티 공급자로 사용하기: AWS Transfer Family	2019년 5월 30일

필수 조건

다음 섹션에서는 서비스를 사용하는 데 필요한 사전 요구 사항을 설명합니다. AWS Transfer Family 최소한 Amazon Simple Storage 서비스 (Amazon S3) 버킷을 생성하고 (IAM) 역할을 통해 AWS Identity and Access Management 해당 버킷에 대한 액세스 권한을 제공해야 합니다. 또한 사용자는 신뢰 관계도 수립해야 합니다. 이러한 신뢰 관계가 있어야 Transfer Family가 IAM 역할을 수임해 버킷에 액세스하고, 사용자의 파일 전송 요청을 처리할 수 있습니다.

주제

- 지원되는 AWS 지역, 엔드포인트 및 할당량
- 가입하기 AWS
- 함께 사용할 스토리지를 구성합니다. AWS Transfer Family
- IAM 역할 및 정책 생성

지원되는 AWS 지역, 엔드포인트 및 할당량

프로그래밍 방식으로 AWS 서비스에 연결하려면 엔드포인트를 사용합니다. 예를 들어, 미국 동부 (오하이오) 지역 (us-east-2) 에 있는 고객의 엔드포인트는 입니다`transfer.us-east-2.amazonaws.com`. 서비스 할당량은 AWS 계정계정의 최대 서비스 리소스 또는 작업 수입니다. 이 가이드에서는 및 의 할당량을 확인할 수 있습니다. AS2 할당량 Transfer for SFTP 커넥터 할당량

지원되는 AWS 지역, 엔드포인트, 서비스 할당량에 대한 자세한 내용은 의 AWS Transfer Family 엔드포인트 및 할당량을 참조하십시오. Amazon Web Services 일반 참조

가입하기 AWS

Amazon Web Services (AWS) 에 가입하면 다음을 AWS포함한 모든 서비스에 AWS 계정이 자동으로 등록됩니다 AWS Transfer Family. 사용자에게는 사용한 서비스에 대해서만 요금이 청구됩니다.

이미 AWS 계정이 있다면 다음 작업으로 건너뛰십시오. AWS 계정이 없는 경우 다음 절차에 따라 계정을 생성합니다.

계정이 없는 경우 다음 단계를 완료하여 계정을 AWS 계정만드세요.

가입하려면 AWS 계정

1. https://portal.aws.amazon.com/billing/signup을 여세요.
2. 온라인 지시 사항을 따르세요.

   등록 절차 중에는 전화를 받고 키패드로 인증 코드를 입력하는 과정이 있습니다.

   에 AWS 계정가입하면 AWS 계정 루트 사용자a가 생성됩니다. 루트 사용자에게는 계정의 모든 AWS 서비스 및 리소스 액세스 권한이 있습니다. 보안 모범 사례는 사용자에게 관리 액세스 권한을 할당하고, 루트 사용자만 사용하여 루트 사용자 액세스 권한이 필요한 작업을 수행하는 것입니다.

요금 및 Transfer Family 사용 비용 추정에 대한 자세한 내용은 AWS Transfer Family 요금을 참조하십시오. AWS Pricing Calculator

AWS 지역 가용성에 대한 자세한 내용은 의 AWS Transfer Family 엔드포인트 및 할당량을 참조하십시오. AWS 일반 참조

함께 사용할 스토리지를 구성합니다. AWS Transfer Family

이 항목에서는 함께 사용할 수 있는 스토리지 옵션에 대해 설명합니다 AWS Transfer Family. Amazon S3 또는 Amazon EFS를 Transfer Family 서버의 스토리지로 사용할 수 있습니다.

목차

Amazon S3 버킷 구성

AWS Transfer Family Amazon S3 버킷에 액세스하여 사용자의 전송 요청을 처리하므로 파일 전송 프로토콜 지원 서버를 설정하는 과정에서 Amazon S3 버킷을 제공해야 합니다. 기존 버킷을 사용할 수도 있고, 새 버킷을 만들어도 됩니다.

> **Note**
>
> 동일한 AWS 리전에 있는 SFTP 서버 및 S3 버킷을 사용할 필요는 없지만 모범 사례로 권장됩니다.

사용자를 설정할 때, 각 사용자에게 IAM 역할을 할당해야 합니다. 이 역할은 Amazon S3 버킷에 대한 사용자의 액세스 수준을 결정합니다.

새 버킷 생성에 대한 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 S3 버킷을 생성하려면 어떻게 해야 합니까?를 참조하세요.

> **Note**
>
> Amazon S3 객체 잠금을 사용하면 일정한 시간 동안 또는 무기한으로 객체를 덮어쓰지 않도록 할 수 있습니다. 이는 다른 서비스와 마찬가지로 Transfer Family에서도 동일한 방식으로 작동합니다. 객체가 존재하고 보호되는 경우 해당 파일에 쓰거나 삭제할 수 없습니다. Amazon S3 객체 잠금에 대한 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 Amazon S3 객체 잠금 사용을 참조하세요.

Amazon S3 액세스 포인트

AWS Transfer Family 공유 데이터 세트에 대한 세분화된 액세스를 쉽게 관리할 수 있는 Amazon S3의 기능인 Amazon S3 액세스 포인트를 지원합니다. S3 버킷 이름을 사용하는 모든 곳에서 S3 액세스 포인트 별칭을 사용할 수 있습니다. Amazon S3 버킷의 공유 데이터에 액세스할 수 있는 서로 다른 권한을 가진 사용자를 위해 Amazon S3에 수백 개의 액세스 포인트를 생성할 수 있습니다.

예를 들어 액세스 포인트를 사용하여 서로 다른 세 팀이 동일한 공유 데이터 세트에 액세스하도록 허용할 수 있습니다. 한 팀은 S3에서 데이터를 읽고, 두 번째 팀은 S3에 데이터를 쓰고, 세 번째 팀은 S3에서 데이터를 읽고, 쓰고, 삭제할 수 있습니다. 위에서 언급한 대로 세분화된 액세스 제어를 구현하려면 여러 팀에 비대칭 액세스를 제공하는 정책이 포함된 S3 액세스 포인트를 생성할 수 있습니다. Transfer Family 서버와 함께 S3 액세스 포인트를 사용하면 수백 개의 사용 사례를 아우르는 복잡한 S3 버킷 정

책을 만들지 않고도 세밀한 액세스 제어를 달성할 수 있습니다. Transfer Family 서버에서 S3 액세스 포인트를 사용하는 방법에 대한 자세한 내용은 Amazon S3를 통한 AWS Transfer Family 데이터 액세스 제어 강화 블로그 게시물을 참조하십시오.

> **Note**
>
> AWS Transfer Family 현재 Amazon S3 다중 지역 액세스 포인트를 지원하지 않습니다.

아마존 S3 HeadObject 동작

> **Note**
>
> Transfer Family 서버를 생성하거나 업데이트할 때 Amazon S3 디렉터리의 성능을 최적화하여 `HeadObject` 호출을 없앨 수 있습니다.

Amazon S3에서 버킷과 객체는 기본 리소스이며 객체는 버킷에 저장됩니다. Amazon S3는 계층적 파일 시스템을 모방할 수 있지만 때로는 일반적인 파일 시스템과 다르게 동작할 수 있습니다. 예를 들어, Amazon S3에서는 디렉터리가 일류 개념이 아니라 객체 키를 기반으로 합니다. AWS Transfer Family 객체의 키를 슬래시 문자 (/) 로 분할하고 마지막 요소를 파일 이름으로 취급한 다음 접두사가 같은 파일 이름을 동일한 경로 아래에 그룹화하여 디렉터리 경로를 유추합니다. `mkdir`를 사용하거나 Amazon S3 콘솔을 사용하여 빈 디렉터리를 생성할 때 폴더 경로를 나타내는 0바이트 객체가 생성됩니다. 이러한 객체의 키는 후행 슬래시로 끝납니다. 이러한 0바이트 객체에 대한 설명은 Amazon S3 사용 설명서의 폴더를 사용하여 Amazon S3 콘솔에서 객체 구성에 설명되어 있습니다.

`ls`명령을 실행하고 일부 결과가 Amazon S3 0바이트 객체 (이러한 객체에는 슬래시 문자로 끝나는 키가 있음) 가 생성되면 Transfer Family는 각 객체에 대한 `HeadObject` 요청을 발행합니다 (자세한 내용은 Amazon Simple Storage Service API 참조 참조). HeadObject 이로 인해 Transfer Family를 통해 Amazon S3를 스토리지로 사용할 때 다음과 같은 문제가 발생할 수 있습니다.

파일을 쓰고 나열할 수 있는 권한만 부여

Amazon S3 객체에 대한 쓰기 권한만 제공하고 싶은 경우도 있습니다. 예를 들어, 버킷의 객체를 작성 (또는 업로드) 하고 나열할 수 있는 액세스 권한은 제공하고, 객체 읽기 (다운로드) 는 허용하지 않는 것이 좋습니다. 파일 전송 클라이언트를 사용하여 `mkdir` 명령을 `ls` 수행하려면 Amazon S3 `ListObjects` 및 `PutObject` 권한이 있어야 합니다. 그러나 Transfer Family에서 파일을 쓰거나 나열하기 위해 `HeadObject` 호출을 해야 하는 경우 이 호출에는 `GetObject` 권한이 필요하므로 Access Denied 오류가 발생하면서 통화가 실패합니다.

> **Note**
>
> Transfer Family 서버를 생성하거나 업데이트할 때 Amazon S3 디렉터리의 성능을 최적화하여 `HeadObject` 호출을 없앨 수 있습니다.

이 경우 슬래시 (/) 로 끝나는 객체에 대해서만 `GetObject` 권한을 추가하는 AWS Identity and Access Management (IAM) 정책 조건을 추가하여 액세스 권한을 부여할 수 있습니다. `/` 이 조건에서는 파일을 읽을 수 없도록 파일에 대한 `GetObject` 호출을 차단하지만 사용자가 폴더를 나열하고 탐색할 수는 있습니다. 다음 예제 정책은 Amazon S3 버킷에 대한 쓰기 및 나열 액세스만 제공합니다. 이 정책을 사용하려면 버킷 *DOC-EXAMPLE-BUCKET* 이름으로 바꾸십시오.

```
{
    "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**
>
> 이 정책은 사용자의 파일 추가를 허용하지 않습니다. 즉, 이 정책을 적용받는 사용자는 파일을 열어 콘텐츠에 콘텐츠를 추가하거나 수정할 수 없습니다. 또한 사용 사례에서 파일을 업로드하기 전에 `HeadObject`을 호출해야 하는 경우에는 이 정책이 적용되지 않습니다.

많은 수의 제로바이트 객체로 인해 지연 시간 문제가 발생합니다.

Amazon S3 버킷에 이러한 0바이트 객체가 많이 포함되어 있는 경우 Transfer Family는 많은 `HeadObject` 호출을 발생시켜 처리 지연을 초래할 수 있습니다. 이 문제에 대한 권장 해결 방법은 최적화 디렉터리를 활성화하여 지연 시간을 줄이는 것입니다.

예를 들어 홈 디렉터리에 10,000개의 하위 디렉터리가 있다고 가정해 보겠습니다. 즉, Amazon S3 버킷에는 10,000개의 폴더가 있습니다. 이 시나리오에서 `ls` (list) 명령을 실행하면 목록 작업에 6~8분이 소요됩니다. 하지만 디렉터리를 최적화하는 경우 이 작업은 몇 초밖에 걸리지 않습니다. 이 옵션은 서버 생성 또는 업데이트 절차 중에 추가 세부 정보 구성 화면에서 설정합니다. 이러한 절차는 SFTP, FTPS 또는 FTP 서버 엔드포인트 구성 항목에 자세히 설명되어 있습니다.

> **Note**
>
> GUI 클라이언트는 제어할 수 없는 `ls` 명령을 실행할 수 있으므로 가능하면 이 설정을 활성화하는 것이 중요합니다.

디렉터리를 최적화하지 않거나 최적화할 수 없는 경우 이 문제의 다른 해결 방법은 모든 0바이트 개체를 삭제하는 것입니다. 유의할 사항:

- 빈 디렉터리가 더 이상 존재하지 않습니다. 디렉터리는 객체 키에 이름이 포함되어 있기 때문에 존재할 수 있습니다.
- 다른 사용자가 `mkdir`을 호출하여 문제가 생기는 것을 막지는 못합니다. 디렉터리 생성을 방지하는 정책을 만들어 이 문제를 완화할 수 있습니다.
- 일부 시나리오에서는 이러한 0바이트 객체를 사용합니다. 예를 들어 /inboxes/customer1000과 같은 구조를 사용하면 받은 편지함 디렉터리가 매일 정리됩니다.

마지막으로 가능한 해결 방법 중 하나는 정책 조건을 통해 표시되는 개체 수를 제한하여 호출 수를 줄이는 것입니다. `HeadObject` 실행 가능한 솔루션이 되려면 모든 하위 디렉터리의 제한된 집합만 볼 수 있다는 점을 인정해야 합니다.

Amazon EFS 파일 시스템 구성

AWS Transfer Family Amazon Elastic File System (Amazon EFS) 에 액세스하여 사용자의 전송 요청을 처리합니다. 따라서 파일 전송 프로토콜 지원 서버 설정의 일환으로 Amazon EFS 파일 시스템을 제공해야 합니다. 기존 파일 시스템을 사용할 수도 있고, 새 버킷을 만들어도 됩니다.

유의할 사항:

- Transfer Family 서버와 Amazon EFS 파일 시스템을 사용하는 경우 서버와 파일 시스템이 동일해야 합니다 AWS 리전.
- 서버와 파일 시스템이 같은 계정에 속할 필요는 없습니다. 서버와 파일 시스템이 같은 계정에 속하지 않는 경우 파일 시스템 정책에서 사용자 역할에 명시적 권한을 부여해야 합니다.

  여러 계정을 설정하는 방법에 대한 자세한 내용은 AWS Organizations 사용 설명서의 조직 내 AWS 계정 관리를 참조하십시오.
- 사용자를 설정할 때, 각 사용자에게 IAM 역할을 할당해야 합니다. 이 역할은 Amazon EFS 파일 시스템에 대한 사용자의 액세스 수준을 결정합니다.
- Amazon EFS 파일 시스템 마운트에 대한 자세한 내용은 Amazon EFS 파일 시스템 마운트를 참조하십시오.

Amazon EFS가 함께 작동하는 방식에 AWS Transfer Family 대한 자세한 내용은 Amazon Elastic File System 사용 설명서의 Amazon EFS 파일 시스템에 있는 파일에 액세스하는 AWS Transfer Family 데 사용하는 방법을 참조하십시오.

Amazon EFS 파일 소유권

Amazon EFS는 Portable Operating System Interface(POSIX) 파일 권한 모델을 사용하여 파일 소유권을 나타냅니다.

POSIX에서 시스템의 사용자는 세 가지 권한 클래스로 분류됩니다. 사용자가 를 사용하여 AWS Transfer Family Amazon EFS 파일 시스템에 저장된 파일에 액세스하도록 허용할 때는 “POSIX 프로필”을 할당해야 합니다. 이 프로필은 Amazon EFS 파일 시스템의 파일 및 디렉터리에 대한 액세스 권한을 결정하는 데 사용됩니다.

- 사용자(u): 파일 또는 디렉터리의 소유자. 일반적으로 파일이나 디렉토리를 만든 사람이 소유자이기도 합니다.
- 그룹(g): 공유하는 파일 및 디렉터리에 대해 동일한 액세스 권한이 필요한 사용자 집합입니다.
- 기타(o): 소유자 및 그룹 구성원을 제외하고 시스템에 액세스할 수 있는 다른 모든 사용자. 이 권한 클래스를 “Public”이라고도 합니다.

POSIX 권한 모델에서는 모든 파일 시스템 개체(파일, 디렉터리, 심볼릭 링크, 명명된 파이프, 소켓)가 앞서 언급한 세 가지 권한 집합과 연결됩니다. Amazon EFS 객체에는 연결된 Unix 스타일 모드가 있습니다. 이 모드 값은 해당 객체에 대한 작업을 수행할 수 있는 권한을 정의합니다.

또한 Unix 스타일 시스템에서 사용자와 그룹은 Amazon EFS가 파일 소유권을 나타내는 데 사용하는 숫자 식별자에 매핑됩니다 Amazon EFS의 경우 단일 소유자 및 단일 그룹이 객체를 소유합니다. Amazon EFS는 사용자가 파일 시스템 객체에 액세스하려고 할 때 매핑된 숫자 ID를 사용하여 권한을 확인합니다.

Transfer Family를 위한 Amazon EFS 사용자 설정

Amazon EFS 사용자를 설정하기 전에 다음 중 하나를 수행할 수 있습니다.

- Amazon EFS에서 사용자를 생성하고 홈 폴더를 설정할 수 있습니다. 세부 정보는 Amazon EFS에서 Transfer Family 사용자 구성를 참조하세요.
- 루트 사용자를 추가하는 것이 편하다면 Amazon EFS 루트 사용자 생성를 할 수 있습니다.

> **Note**
>
> Transfer Family 서버는 POSIX 권한을 설정하는 Amazon EFS 액세스 포인트를 지원하지 않습니다. Transfer Family 사용자의 POSIX 프로필(이전 섹션에서 설명)은 POSIX 권한을 설정하는 기능을 제공합니다. 이러한 권한은 UID, GID 및 보조 GID를 기반으로 세분화된 액세스를 위해 사용자 수준에서 설정됩니다.

Amazon EFS에서 Transfer Family 사용자 구성

Transfer Family는 사용자를 지정한 UID/GID 및 디렉터리에 매핑합니다. UID/GID/디렉토리가 EFS에 아직 없는 경우 Transfer에서 사용자에게 할당하기 전에 먼저 UID/GID/디렉토리를 생성해야 합니다. Amazon EFS 사용자 생성에 대한 자세한 내용은 Amazon Elastic File System 사용 설명서의 NFS(Network File System) 수준에서의 사용자, 그룹 및 권한 작업을 참조하세요.

Transfer Family에서 Amazon EFS 사용자를 설정하는 단계

1. `PosixProfile` 필드를 사용하여 Transfer Family에서 사용자에 대한 EFS UID 및 GID를 매핑합니다.
2. 사용자가 로그인 시 특정 폴더에서 시작하도록 하려면 `HomeDirectory` 필드 아래에 EFS 디렉터리를 지정할 수 있습니다.

CloudWatch 규칙 및 Lambda 함수를 사용하여 프로세스를 자동화할 수 있습니다. EFS와 상호 작용하는 Lambda 함수의 예는 서버리스 애플리케이션에서 Amazon EFS 사용을 AWS Lambda 참조하십시오.

또한 Transfer Family 사용자를 위한 논리적 디렉터리를 구성할 수 있습니다. 자세한 내용은 논리적 디렉터리를 사용하여 Transfer Family 디렉터리 구조를 단순화합니다. 항목의 Amazon EFS를 위한 논리적 디렉터리 구성를 참조하세요.

Amazon EFS 루트 사용자 생성

조직에서 사용자 구성을 위해 SFTP/FTPS를 통한 루트 사용자 액세스를 허용하려는 경우 UID 및 GID가 0(루트 사용자) 인 사용자를 생성한 다음 해당 루트 사용자를 사용하여 폴더를 생성하고 나머지 사용자의 POSIX ID 소유자를 할당할 수 있습니다. 이 옵션의 장점은 Amazon EFS 파일 시스템을 탑재할 필요가 없다는 것입니다.

Amazon EFS 서비스 관리 사용자 추가에 설명된 단계를 수행하고, 사용자 ID와 그룹 ID 모두에 0을 입력합니다.

지원되는 Amazon EFS 명령

다음 명령은 AWS Transfer Family를 위한 Amazon EFS에서 지원됩니다.

- `cd`
- `ls/dir`
- `pwd`
- `put`
- `get`
- `rename`
- `chown`: 루트(즉, uid=0인 사용자)만 파일 및 디렉터리의 소유권과 권한을 변경할 수 있습니다.

- chmod: 루트(즉, uid=0인 사용자)만 파일 및 디렉터리의 소유권과 권한을 변경할 수 있습니다.
- chgrp: 루트 또는 파일 소유자만 파일 그룹을 보조 그룹 중 하나로 변경할 수 있는 파일 소유자만 지원됩니다.
- ln -s/symlink
- mkdir
- rm/delete
- rmdir
- chmtime

IAM 역할 및 정책 생성

이 주제에서는 함께 AWS Transfer Family사용할 수 있는 정책 및 역할 유형을 설명하고 사용자 역할을 생성하는 프로세스를 안내합니다. 또한 세션 정책의 작동 방식을 설명하고 사용자 역할 예를 제공합니다.

AWS Transfer Family 는 다음과 같은 유형의 역할을 사용합니다.

- 사용자 역할 — 서비스 관리 사용자가 필요한 Transfer Family 리소스에 액세스할 수 있습니다. AWS Transfer Family Transfer Family 사용자 ARN의 컨텍스트에서 이 역할을 맡습니다.
- 액세스 역할 – 전송 중인 Amazon S3 파일에만 액세스할 수 있습니다. 인바운드 AS2 전송의 경우 액세스 역할은 계약의 Amazon 리소스 이름(ARN)을 사용합니다. 아웃바운드 AS2 전송의 경우 액세스 역할은 커넥터의 ARN을 사용합니다.
- 간접 호출 역할 - Amazon API Gateway와 함께 서버의 사용자 지정 자격 증명 공급자로 사용합니다. Transfer Family는 Transfer Family 서버 ARN의 컨텍스트에서 이 역할을 담당합니다.
- 로깅 역할 — Amazon에 항목을 기록하는 데 사용됩니다 CloudWatch. Transfer Family는 이 역할을 사용하여 File Transfer에 대한 정보와 함께 성공 및 실패 세부 정보를 기록합니다. Transfer Family는 Transfer Family 서버 ARN의 컨텍스트에서 이 역할을 담당합니다. 아웃바운드 AS2 전송의 경우, 로깅 역할은 커넥터 ARN을 사용합니다.
- 실행 역할 – Transfer Family 사용자가 전화를 걸어 워크플로를 시작할 수 있습니다. Transfer Family는 Transfer Family 워크플로 ARN의 컨텍스트에서 이 역할을 담당합니다.

이러한 역할 외에도 세션 정책을 사용할 수 있습니다. 세션 정책은 필요한 경우 액세스를 제한하는 데 사용됩니다. 이러한 정책은 독립형이라는 점에 유의하세요. 즉, 이러한 정책은 역할에 추가할 수 없습니다. 대신 Transfer Family 사용자에게 직접 세션 정책을 추가할 수 있습니다.

> **Note**
>
> 서비스 관리 Transfer Family 사용자를 생성하는 경우 홈 폴더를 기반으로 정책 자동 생성을 선택할 수 있습니다. 사용자 액세스를 자신의 폴더에 제한하려는 경우 유용한 단축키입니다. 또한 세션 정책의 작동 방식에서 세션 정책 및 예에 대한 세부 정보를 볼 수 있습니다. 세션 정책에 대한 자세한 내용은 IAM 사용 설명서의 세션 정책에서도 찾아볼 수 있습니다.

주제

사용자 역할 생성

사용자를 만들 때, 사용자 액세스에 관한 결정을 여러 번 해야 합니다. 이러한 결정에는 사용자가 액세스할 수 있는 Amazon S3 버킷 또는 Amazon EFS 파일 시스템, 각 Amazon S3 버킷의 일부 및 파일 시스템의 어떤 파일에 액세스할 수 있는지, 사용자에게 어떤 권한(예: PUT 또는GET)이 포함됩니다.

액세스를 설정하려면 해당 액세스 정보를 제공하는 ID 기반 AWS Identity and Access Management (IAM) 정책 및 역할을 생성합니다. 이 과정의 일부에서, 사용자가 파일 작업의 대상 또는 원본인 Amazon S3 bucket 또는 Amazon EFS에 액세스할 수 있게 됩니다. 이렇게 하려면 나중에 자세하게 설명하는, 다음과 같은 상위 수준 단계를 거쳐야 합니다.

사용자 역할 생성

1. 에 대한 IAM 정책을 생성합니다. AWS Transfer Family이에 대한 설명은 다음에 대한 IAM 정책을 만들려면 AWS Transfer Family에 나와 있습니다.
2. IAM 역할을 생성하여 새 IAM 정책을 연결합니다. 예는 읽기/쓰기 액세스 정책의 예를 참조하세요.
3. AWS Transfer Family 와 IAM 역할 간의 신뢰 관계를 구축하십시오. 이에 대한 설명은 신뢰 관계를 구축하기 위해에 나와 있습니다.

다음 절차에서는 IAM 정책 및 역할을 생성하는 방법을 설명합니다.

다음에 대한 IAM 정책을 만들려면 AWS Transfer Family

1. https://console.aws.amazon.com/iam/에서 IAM 콘솔을 엽니다.

2. 탐색 창에서 정책을 선택한 후 정책 생성을 선택합니다.
3. 정책 생성 페이지에서 JSON 탭을 선택합니다.
4. 표시되는 편집기에서, 편집기 내용을 IAM 역할에 연결할 IAM 정책으로 바꿉니다.

   읽기/쓰기 액세스 권한을 부여하거나 사용자를 홈 디렉터리로 제한할 수 있습니다. 자세한 내용은 읽기/쓰기 액세스 정책의 예를 참조하세요.
5. 정책 검토를 선택하고 정책의 이름과 설명을 입력한 다음 정책 생성을 선택합니다.

그런 다음 IAM 역할을 만들고 새 IAM 정책을 역할에 연결합니다.

IAM 역할을 만들려면 AWS Transfer Family

1. 탐색 창에서 역할을 선택한 후 역할 생성을 선택합니다.

   역할 생성 페이지에서 AWS 서비스가 선택되어 있는지 확인합니다.
2. 서비스 목록에서 전송을 선택하고, 다음: 권한을 선택합니다. 이렇게 하면 과 사이에 AWS Transfer Family 신뢰 관계가 설정됩니다. AWS
3. 권한 정책 연결 섹션에서, 방금 생성한 정책을 찾아 선택한 다음 다음: 태그를 선택합니다.
4. (옵션) 태그의 키와 값을 입력하고 다음: 검토를 선택합니다.
5. 검토 페이지에 새 역할의 명칭과 설명을 입력한 다음 역할 생성을 선택합니다.

다음으로, AWS Transfer Family 와 AWS사이에 신뢰 관계를 설정합니다.

신뢰 관계를 구축하기 위해

> **Note**
>
> 이 예에서는 `ArnLike` 및 `ArnEquals` 모두를 사용합니다. 기능적으로 동일하므로 정책을 구성할 때 둘 중 하나를 사용할 수 있습니다. Transfer Family 설명서에서는 조건에 와일드카드 문자가 포함된 경우에는 `ArnLike`를 사용하고, 정확한 일치 조건을 나타내기 위해서는 `ArnEquals`를 사용합니다.

1. IAM 콘솔에서 방금 생성한 역할을 선택합니다.
2. 요약 페이지에서 신뢰 관계를 선택한 다음 신뢰 관계 편집을 선택합니다.

3. 신뢰 관계 편집 편집기에서 서비스가 "transfer.amazonaws.com"인지 확인합니다. 액세스 정책은 다음과 같습니다.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "transfer.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

대리인 혼동 문제로부터 스스로를 보호하려면 aws:SourceAccount 및 aws:SourceArn 조건 키를 사용할 것을 권장합니다. 소스 계정은 서버의 소유자이고 소스 ARN은 사용자의 ARN입니다. 예:

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

사용자 계정의 서버 대신 특정 서버로 제한하려는 경우에도 ArnLike 조건을 사용할 수 있습니다. 예:

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

> **Note**
>
> 상기 예에서 각 ### ## ## ###를 자신의 정보로 바꿉니다.

헷갈리는 대리인 문제에 대한 자세한 내용 및 기타 예는 교차 서비스 혼동된 대리인 방지 단원을 참조하세요.

4. 신뢰 정책 업데이트를 선택하여 액세스 정책을 업데이트합니다.

이제 사용자를 대신하여 AWS 서비스를 AWS Transfer Family 호출할 수 있는 IAM 역할을 생성했습니다. 생성한 IAM 정책과 역할을 연결해, 사용자에게 액세스를 부여했습니다. 서버 엔드포인트 시작하기 AWS Transfer Family 섹션에서, 이 역할과 정책은 사용자 또는 일반 사용자에게 할당됩니다.

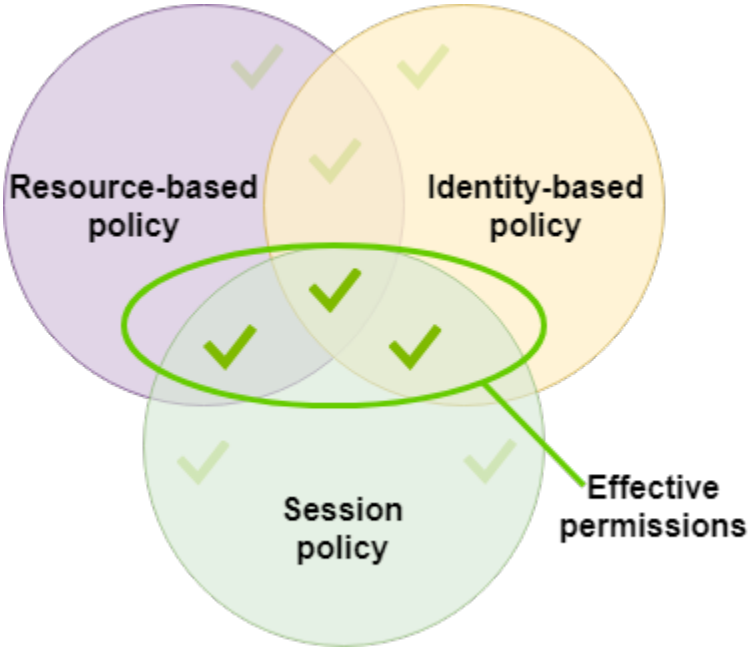
참고 항목

- IAM 역할에 대한 자세한 일반 정보는 IAM 사용 설명서의 AWS 서비스에 권한을 위임하기 위한 역할 생성을 참조하십시오.
- Amazon S3 리소스의 자격 증명 기반 정책에 대한 자세한 내용은 Amazon Simple Storage Service 사용 설명서의 Amazon S3의 자격 증명 및 액세스 관리를 참조하세요.
- Amazon EFS 리소스의 자격 증명 기반 정책에 대해 자세히 알아보려면 Amazon Elastic File System 사용 설명서의 IAM을 사용하여 파일 시스템 데이터 액세스 제어를 참조하세요.

세션 정책의 작동 방식

관리자가 역할을 생성하면 역할에 여러 사용 사례 또는 팀 구성원을 포괄할 수 있는 광범위한 권한이 포함되는 경우가 많습니다. 관리자가 콘솔 URL을 구성하면 세션 정책을 사용하여 결과 세션에 대한 권한을 줄일 수 있습니다. 예를 들어 읽기/쓰기 권한이 있는 역할을 생성하는 경우 사용자의 액세스를 홈 디렉터리로만 제한하는 URL을 설정할 수 있습니다.

세션 정책은 역할 또는 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 세션 정책은 객체 접두사에 사용자 이름이 포함된 버킷의 일부에만 액세스할 수 있도록 사용자를 잠그는 데 유용합니다. 다음 다이어그램은 세션 정책의 권한이 세션 정책과 리소스 기반 정책의 교집합과 세션 정책 및 자격 증명 기반 정책의 교집합임을 보여줍니다.



자세한 내용은 IAM 사용 설명서의 세션 정책을 참조하세요.

에서 AWS Transfer Family세션 정책은 Amazon S3로 또는 Amazon S3에서 전송하는 경우에만 지원됩니다. 다음 예 정책은 사용자가 자신의 home 디렉터리에만 액세스하게 하는 세션 정책입니다. 유의할 사항:

- `GetObjectACL` 및 `PutObjectACL` 명령문은 크로스 계정 액세스를 활성화할 필요가 있는 경우에만 요구됩니다. 예컨대, Transfer Family 서버가 다른 계정의 버킷에 액세스할 필요가 있는 경우입니다.
- 세션 정책의 최대 길이는 2,048자입니다. 자세한 내용은 API 참조의 `CreateUser` 작업에 대한 정책 요청 파라미터를 참조하세요.
- Amazon S3 버킷을 AWS Key Management Service (AWS KMS) 사용하여 암호화한 경우 정책에 추가 권한을 지정해야 합니다. 자세한 내용은 Amazon S3의 데이터 암호화를 참조하세요.

```
{
    "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**
>
> 위의 정책 예에서는 사용자의 홈 디렉터리가 디렉터리임을 나타내기 위해 후행 슬래시를 포함하도록 설정된 것으로 가정합니다. 그 반면에 후행 슬래시가 없이 사용자의 `HomeDirectory`를 설정하는 경우에는 이를 정책의 일부로 포함해야 합니다.

이전 예 정책에서, `transfer:HomeFolder`, `transfer:HomeBucket`, `transfer:HomeDirectory` 정책 파라미터의 사용을 기억해 두세요. 이러한 파라미터는 HomeDirectory및 에 설명된 대로 사용자를 위해 구성된 파라미터에 대해 설정됩니다API Gateway 메서드 구현. `HomeDirectory` 이러한 파라미터의 정의는 다음과 같습니다.

- `transfer:HomeBucket` 파라미터는 `HomeDirectory`의 첫 번째 구성 요소로 대체됩니다.

- `transfer:HomeFolder` 파라미터가 `HomeDirectory` 파라미터의 나머지 부분으로 대체됩니다.
- `transfer:HomeDirectory` 파라미터는 `Resource` 명령문에서 S3 Amazon 리소스 이름 (ARN)의 일부로 사용할 수 있도록 선행 슬래시(/)가 제거되었습니다.

> **Note**
>
> 귀하가 논리적 디렉터리를 사용하는 경우—즉 사용자의 `homeDirectoryType`가 `LOGICAL`인 경우—이러한 정책 파라미터(`HomeBucket`, `HomeDirectory`, 및 `HomeFolder`)는 지원되지 않습니다.

예를 들어, Transfer Family 사용자에 대해 구성된 `HomeDirectory` 파라미터가 `/home/bob/amazon/stuff/`(이)라고 가정해 보겠습니다.

- `transfer:HomeBucket`를 `/home`(으)로 설정합니다.
- `transfer:HomeFolder`를 `/bob/amazon/stuff/`(으)로 설정합니다.
- `transfer:HomeDirectory`는 `home/bob/amazon/stuff/`가 됩니다.

첫 번째 `"Sid"`은(는) 사용자가 `/home/bob/amazon/stuff/`부터 시작하여 모든 디렉터리를 나열할 수 있게 해 줍니다.

두 번째 `"Sid"`은(는) 동일한 경로인 `/home/bob/amazon/stuff/`에 대한 사용자 `put` 및 `get` 액세스를 제한합니다.

읽기/쓰기 액세스 정책의 예

Amazon S3 버킷에 대한 읽기/쓰기 권한 부여

다음 예제 정책은 Amazon S3 버킷의 객체에 대한 읽기/쓰기 액세스 권한을 AWS Transfer Family 부여합니다.

유의할 사항:

- *`DOC-EXAMPLE-BUCKET`*을 Amazon S3 버킷의 이름으로 바꿉니다.
- `GetObjectACL` 및 `PutObjectACL` 명령문은 크로스 계정 액세스를 활성화할 필요가 있는 경우에만 요구됩니다. 예컨대, Transfer Family 서버가 다른 계정의 버킷에 액세스할 필요가 있는 경우입니다.

- `GetObjectVersion` 및 `DeleteObjectVersion` 보고서는 액세스 중인 Amazon S3 버킷에서 버전 관리가 활성화된 경우에만 필요합니다.

> **Note**
>
> 버킷의 버전 관리를 활성화한 경우 Amazon S3에서만 버전 관리를 일시 중지할 수 있고 완전히 비활성화할 수는 없으므로 이러한 권한이 필요합니다. 자세한 내용은 버전 관리 없음, 버전 관리 지원 및 버전 관리 일시 중지 버킷을 참조하십시오.

```
{
    "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/*"
        }
    ]
}
```

Amazon EFS 파일 시스템의 파일에 대한 파일 시스템 액세스 권한 부여

> **Note**
>
> 정책 외에도 POSIX 파일 권한이 적절한 액세스를 허용하고 있는지도 확인해야 합니다. 자세한 내용은 Amazon Elastic File System 사용 설명서의 NFS(네트워크 파일 시스템) 수준에서 사용자, 그룹, 권한 작업을 참조하세요.

다음 예 정책은 루트 파일 시스템에 Amazon EFS 파일 시스템의 파일에 대한 액세스 권한을 부여합니다.

> **Note**
>
> 다음 예시에서는 *### ## ####*, *## ID#* 파일이 있는 계정으로, 그리고 Amazon Elastic File System (Amazon EFS) 의 *file-system-id*ID로 바꾸십시오.

```
{
    "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"
        }
    ]
}
```

다음 예 정책은 Amazon EFS 파일 시스템의 파일에 대한 사용자 파일 시스템 액세스 권한을 부여합니다.

```
{
```

```
    "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 자습서

AWS Transfer Family 사용 설명서는 여러 사용 사례에 대한 자세한 안내를 제공합니다.

- 서버 엔드포인트 시작하기 AWS Transfer Family: 이 자습서에서는 SFTP Transfer Family 서버 및 서비스 관리 사용자를 만드는 과정을 안내한 다음 클라이언트를 사용하여 파일을 전송하는 방법을 보여줍니다.
- SFTP 커넥터 설정 및 사용: 이 자습서에서는 SFTP 커넥터를 설정한 다음 Amazon S3 스토리지와 SFTP 서버 간에 파일을 전송하는 방법을 보여줍니다.
- Amazon API Gateway 메서드를 사용자 지정 ID 공급자로 설정 : 이 자습서에서는 Amazon API Gateway 메서드를 설정하고 이를 사용자 지정 ID 공급자로 사용하여 AWS Transfer Family 서버에 파일을 업로드하는 방법을 보여줍니다.
- 파일 암호 해독을 위한 관리형 워크플로우 설정: 이 자습서는 암호 해독 단계가 포함된 관리형 워크플로를 설정하는 방법과 암호화된 파일을 Amazon S3 버킷에 업로드한 다음 복호화된 파일을 보는 방법을 보여줍니다.
- AS2 구성 설정: 이 자습서에서는 AS2 Transfer Family 서버를 구성하는 데 필요한 단계를 안내합니다. 인증서 가져오기, 프로파일 및 계약 생성, 선택적으로 AS2 커넥터 생성, 구성 테스트에 대한 지침이 나와 있습니다.

주제

서버 엔드포인트 시작하기 AWS Transfer Family

이 튜토리얼을 사용하여 AWS Transfer Family (Transfer Family) 를 시작하십시오. Amazon S3 스토리지를 사용하여 공개적으로 액세스할 수 있는 엔드포인트가 있는 SFTP 지원 서버를 생성하고, 서비스 관리형 인증을 통해 사용자를 추가하고, Cyberduck으로 파일을 전송하는 방법을 알아봅니다.

주제

사전 조건

시작하기 전에 먼저 필수 조건 요건을 충족해야 합니다. 이 설정의 일환으로 Amazon Simple Storage 서비스 (Amazon S3) 버킷과 (AWS Identity and Access Management IAM) 사용자 역할을 생성합니다.

AWS Transfer Family 콘솔을 사용하는 데 필요한 권한도 있고, Transfer Family가 사용하는 다른 AWS 서비스 (예: Amazon Simple Storage Service, Amazon Elastic File System AWS Certificate Manager, Amazon Route 53) 를 구성하는 데 필요한 권한도 있습니다. 예를 들어, Transfer AWS Family를 사용하여 파일을 주고 받는 사용자의 경우 AmazonS3는 Amazon S3 버킷을 설정하고 사용할 권한을 FullAccess 부여합니다. Amazon S3 버킷을 생성하려면 이 정책의 일부 권한이 필요합니다.

Transfer Family 콘솔을 사용하려면 다음이 필요합니다.

- AWSTransferConsoleFullAccessSFTP 사용자에게 Transfer Family 리소스를 생성할 수 있는 권한을 부여합니다.
- IAM FullAccess (특히 IAM 역할 생성을 허용하는 정책) 은 Transfer Family가 Amazon CloudWatch Logs에서 서버의 로깅 역할을 자동으로 생성하거나 서버에 로그인하는 사용자의 사용자 역할을 자동으로 생성하도록 하려는 경우에만 필요합니다.
- VPC 서버 유형을 생성하고 삭제하려면 정책에 ec2: CreateVpc 엔드포인트 및 ec2: DeleteVpc 엔드포인트 작업을 추가해야 합니다.

> **Note**
>
> AmazonS3 FullAccess 및 IAM FullAccess 정책 자체는 일반적인 용도에는 필요하지 않습니다. AWS Transfer Family여기에는 필요한 모든 권한이 포함되는지 확인할 수 있는 간단한 방법이 나와 있습니다. 또한 이러한 정책은 모든 고객이 사용할 수 있는 표준 정책인 AWS 관리형 정책입니다. AWS 이러한 정책의 개별 권한을 보고 목적에 필요한 최소 권한을 결정할 수 있습니다.

1단계: AWS Transfer Family 콘솔에 로그인

Transfer Family에 로그인하려면

1. AWS Management Console 로그인하고 https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 계정 ID 또는 별칭에 AWS 계정사용자 ID를 입력합니다.
3. IAM 사용자 이름에 Transfer Family용으로 생성한 사용자 역할의 이름을 입력합니다.
4. 비밀번호에는 AWS 계정 비밀번호를 입력합니다.
5. 로그인을 선택합니다.

2단계: SFTP 지원 서버 생성

Secure Shell(SSH) File Transfer 프로토콜(SFTP)은 인터넷을 통한 안전한 데이터 전송에 사용되는 네트워크 프로토콜입니다. 이 프로토콜은 SSH의 전체 보안 및 인증 기능을 지원합니다. 금융 서비스, 의료, 소매 및 광고와 같은 다양한 산업의 사업 파트너 간의 민감한 정보를 비롯한 데이터를 교환하는 데 널리 사용됩니다.

SFTP 지원 서버를 생성하려면

1. 탐색 창에서 서버를 선택한 다음 서버 생성을 선택합니다.
2. 프로토콜 선택에서 SFTP를 선택하고 다음을 선택합니다.
3. 자격 증명 제공자 선택에서 Transfer Family에 사용자 ID 및 키를 저장하도록 관리 서비스를 선택한 후 다음을 선택합니다.
4. 엔드포인트 선택에서 다음을 수행합니다.
    a. 엔드포인트 타입으로는 공적으로 액세스 가능 엔드포인트 타입을 선택합니다.
    b. 사용자 지정 호스트 이름의 경우 없음을 선택합니다.
    c. 다음을 선택합니다.
5. 도메인 선택에서 Amazon S3를 선택합니다.
6. 추가 세부 정보 구성에서 암호화 알고리즘 옵션에 대해 서버에서 사용하도록 설정된 암호화 알고리즘이 포함된 보안 정책을 선택합니다. 최신 보안 정책이 기본값입니다. 자세한 내용은 을 참조하십시오. 서버 보안 정책 AWS Transfer Family

> **Note**
>
> 서버의 관리형 워크플로를 추가하는 경우에만 CloudWatch로깅을 위한 새 역할 만들기를 선택하세요. 서버 이벤트를 기록하기 위해 IAM 역할을 생성할 필요는 없습니다.

7. 검토 및 생성에서 서버 생성을 선택합니다. 서버 페이지로 이동합니다.

새 서버 상태가 온라인으로 변경되기까지 몇 분 정도 걸릴 수 있습니다. 이때 서버에서 파일 작업을 수행할 수 있지만 먼저 사용자를 만들어야 합니다. 사용자 생성에 대한 자세한 내용은 을 참조하십시오서버 엔드포인트의 사용자 관리.

3단계: 서비스 관리 사용자 추가

SFTP 지원 서버에 사용자를 추가하려면

1. 서버 페이지에서 사용자를 추가할 서버를 선택합니다.
2. 사용자 추가를 선택합니다.
3. 사용자 구성 섹션의 사용자 이름에 사용자 이름을 입력합니다. 이 사용자 이름은 3~100자여야 합니다. 사용자 이름에는 a—z, A-Z, 0—9, 밑줄 '_', 하이픈 '-', 마침표 '등의 문자를 사용할 수 있습니다. . ', 그리고 기호 (@). 사용자 이름은 하이픈, 마침표 및 @ 기호로 시작할 수 없습니다.
4. Access의 경우 에서 IAM 역할 및 정책 생성 생성한 IAM 역할을 선택합니다. 이 IAM 역할에는 Amazon S3 버킷에 액세스할 수 있는 권한과 서비스와의 신뢰 관계가 포함된 IAM 정책이 포함됩니다. AWS Transfer Family 에 설명된 절차는 적절한 신뢰 관계를 설정하는 방법을 신뢰 관계를 구축하기 위해 보여줍니다.
5. 정책에서 없음을 선택합니다.
6. 홈 디렉터리의 경우 전송하는 데 사용할 데이터를 저장할 Amazon S3 버킷을 선택합니다 AWS Transfer Family. home디렉터리 경로를 입력합니다. 이 디렉토리는 사용자가 클라이언트를 사용하여 로그인할 때 보게 되는 디렉토리입니다.

   세션 정책을 사용할 수 있도록 사용자 이름이 포함된 디렉터리 경로를 사용하는 것이 좋습니다. 세션 정책은 Amazon S3 버킷에서의 사용자 액세스를 해당 사용자의 home 디렉터리로 제한합니다. 세션 정책 사용에 대한 자세한 내용은 을 참조하십시오세션 정책의 작동 방식.

   원하는 경우 이 파라미터를 비워 두고 Amazon S3 버킷의 `root` 디렉터리를 사용할 수 있습니다. 이 옵션을 선택하는 경우 IAM 역할이 `root` 디렉터리에 대한 액세스를 제공하는지 확인하십시오.

7. Restricted 확인란을 선택하여 사용자가 home 디렉터리 외부에 액세스하는 것을 차단하십시오. 또한 사용자가 Amazon S3 버킷 이름 또는 폴더 이름을 볼 수 없게 됩니다.
8. SSH 공개 키의 경우, SSH 키 쌍의 공개 SSH 키 부분을 형식으로 입력합니다. `ssh-rsa <string>`

   새 사용자를 추가하려면 먼저 서비스에서 키를 검증해야 합니다. SSH 키 쌍을 생성하는 방법에 대한 자세한 내용은 을 참조하십시오서비스 관리 사용자를 위한 SSH 키 생성.
9. (옵션) 키와 값에 하나 이상의 태그를 키-값 쌍로 입력하고 태그 추가를 선택합니다.
10. 추가를 선택해 새 사용자를 원하는 서버에 추가합니다.

    새 사용자는 서버 세부 정보 페이지의 사용자 섹션에 나타납니다.

4단계: 클라이언트를 사용하여 파일 전송

클라이언트에서 전송 작업을 지정하여 AWS Transfer Family 서비스를 통해 파일을 전송합니다. AWS Transfer Family 여러 클라이언트를 지원합니다. 자세한 내용은 클라이언트를 사용하여 서버 엔드포인트를 통한 파일 전송을 참조하세요.

이 섹션에는 Cyberduck과 OpenSSH를 사용하기 위한 절차가 수록되어 있습니다.

주제

- Cyberduck 사용하기
- OpenSSH 사용

Cyberduck 사용하기

Cyberduck을 사용하여 파일을 AWS Transfer Family 전송하려면

1. Cyberduck 클라이언트를 엽니다.
2. 연결 열기를 선택합니다.
3. 연결 열기 대화 상자에서 SFTP(SSH File Transfer 프로토콜)를 선택합니다.
4. 서버에 서버 엔드포인트를 입력합니다. 서버 엔드포인트는 서버 세부 정보 페이지에 있습니다. SFTP, FTPS 및 FTP 서버 세부 정보 보기를 참조하세요.
5. 포트 번호에는 SFTP에 **22**를 입력합니다.
6. 사용자 이름에 서버 엔드포인트의 사용자 관리에서 생성한 사용자 이름을 입력합니다.
7. SSH 프라이빗 키에는 SSH 프라이빗 키를 선택 또는 입력합니다.

8. 연결을 선택합니다.
9. 파일 전송을 실행합니다.

   파일 위치에 따라, 다음 중 하나를 수행하세요.

   - 로컬 디렉터리(소스)에서 전송할 파일을 선택하고, Amazon S3 디렉터리(대상)로 끌어다 놓습니다.
   - Amazon S3 디렉터리(소스)에서 전송할 파일을 선택하고, 로컬 디렉터리(대상)로 끌어다 놓습니다.

OpenSSH 사용

아래의 지침을 이용해 OpenSSH를 이용하는 명령줄로 파일을 전송하세요.

> **Note**
>
> 이 클라이언트는 SFTP 지원 서버에서만 작동합니다.

OpenSSH 명령줄 유틸리티를 AWS Transfer Family 사용하여 파일을 전송하려면

1. Linux나 Macintosh에서 명령 터미널을 엽니다.
2. 프롬프트에서 다음 명령을 입력합니다. `% sftp -i transfer-key sftp_user@service_endpoint`

   앞의 명령에서 `sftp_user`는 사용자 이름이고 `transfer-key`는 SSH 프라이빗 키입니다. 다음은 선택한 서버의 AWS Transfer Family 콘솔에 표시된 서버 엔드포인트입니다. `service_endpoint`

   `sftp` 프롬프트가 나타날 것입니다.
3. (옵션) 사용자의 홈 디렉터리를 보려면 `sftp` 프롬프트에 다음 명령을 입력합니다. `sftp> pwd`
4. 다음 줄에 다음 텍스트를 입력합니다. `sftp> cd /mybucket/home/sftp_user`

   본 시작하기 실습에서는 이 Amazon S3 버킷이 파일 전송 대상입니다.
5. 다음 줄에 다음 명령을 입력합니다. `sftp> put filename.txt`

   `put` 명령은 파일을 Amazon S3 버킷으로 전송합니다.

파일 전송이 진행 중이거나 완료되었음을 나타내는, 다음과 비슷한 메시지가 표시됩니다.

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

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

파일 암호 해독을 위한 관리형 워크플로우 설정

이 자습서에서는 복호화 단계가 포함된 관리형 워크플로를 설정하는 방법을 설명합니다. 또한 자습서에서는 Amazon S3 버킷에 암호화된 파일을 업로드한 다음 동일한 버킷에서 복호화된 파일을 보는 방법을 보여줍니다.

> **Note**
>
> AWS 스토리지 블로그에는 Transfer Family Managed 워크플로를 사용하여 코드를 작성하지 않고 파일을 간단히 해독하는 방법, PGP를 사용한 파일 암호화 및 암호 해독 방법을 설명하는 게시물이 있습니다. AWS Transfer Family

주제

1단계: 실행 역할 구성

Transfer Family가 워크플로를 시작하는 데 사용할 수 있는 AWS Identity and Access Management (IAM) 실행 역할을 생성합니다. 실행 역할을 생성하는 프로세스는 워크플로에 대한 IAM 정책에 설명되어 있습니다.

> **Note**
>
> 신뢰 관계를 구축하기 위해에 설명된 대로 실행 역할을 생성하는 과정에서 실행 역할과 Transfer Family 사이에 신뢰 관계를 확실히 구축해야 합니다.

다음 실행 역할 정책에는 이 자습서에서 생성한 워크플로를 시작하는 데 필요한 모든 권한이 포함되어 있습니다. 이 정책 예를 사용하려면 *user input placeholders*를 실제 정보로 바꾸세요. 암호화된 파일을 업로드하는 Amazon S3 버킷의 이름으로 DOC-EXAMPLE-BUCKET 바꾸십시오.

> **Note**
>
> 모든 워크플로에 이 예에 나열된 모든 권한이 필요한 것은 아닙니다. 특정 워크플로의 단계 유형에 따라 권한을 제한할 수 있습니다. 사전 정의된 각 단계 타입에 필요한 권한은 사전 정의된 단계 사용에 설명되어 있습니다. 사용자 지정 단계에 필요한 권한은 사용자 지정 단계 IAM 권한에 설명되어 있습니다.

```
{
    "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/
*"
        }
    ]
}
```

2단계: 관리형 워크플로 생성

이제 복호화 단계가 포함된 워크플로를 만들어야 합니다.

복호화 단계가 포함된 워크플로를 만들려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 워크플로를 선택한 다음, 워크플로 생성을 선택합니다.
3. 다음 세부 정보를 입력합니다.
   - 설명을 입력합니다. 예를 들어, **`Decrypt workflow example`**.
   - 공칭 단계 섹션에서 단계 추가를 선택합니다.
4. 단계 타입 선택에서 파일 복호화를 선택한 후 다음을 선택합니다.
5. 파라미터 구성 대화 상자에서 다음 사항을 지정합니다.
   - 설명이 포함된 단계 이름을 입력합니다. 예: **`decrypt-step`**. 단계 이름에는 공백은 허용되지 않습니다.
   - 복호화된 파일의 대상으로 Amazon S3을 선택합니다.
   - 대상 버킷 이름으로는 1단계에서 생성한 IAM 정책에서 DOC-EXAMPLE-BUCKET로 지정한 것과 동일한 Amazon S3 버킷을 선택합니다.
   - 대상 키 접두사에는 대상 버킷에 복호화된 파일을 저장할 접두사(폴더)의 이름을 입력합니다. 예: **`decrypted-files/`**.

> **Note**
>
> 접두사에는 반드시 후행 슬래시(/)를 추가해야 합니다.

- 이 자습서에서는 기존 항목 덮어쓰기를 선택 해제해 두세요. 이 설정을 지우면 기존 파일과 이름이 같은 파일을 해독하려고 하면 워크플로 처리가 중지되고 새 파일은 처리되지 않습니다.

다음을 선택하여 검토 화면으로 이동합니다.

6. 해당 단계의 세부 정보를 검토합니다. 모든 내용이 정확하면 단계 생성을 선택합니다.
7. 워크플로에는 단일 복호화 단계만 필요하므로 추가로 구성할 단계가 없습니다. 워크플로 생성을 선택하여 새 워크플로를 생성합니다.

새 워크플로의 워크플로 ID를 기록해 둡니다. 다음 단계에서 이 ID가 필요합니다. 이 자습서에서는 *w-1234abcd5678efghi*를 예 워크플로 ID로 사용합니다.

3단계: 서버에 워크플로 추가 및 사용자 생성

이제 복호화 단계가 포함된 워크플로를 Transfer Family 서버에 연결해야 합니다. 이 자습서에서는 기존 Transfer Family 서버에 워크플로를 연결하는 방법을 보여줍니다. 또는 워크플로에 사용할 새 서버를 만들 수 있습니다.

워크플로를 서버에 연결한 후에는 SFTP를 통해 서버에 연결하고 워크플로가 실행되도록 트리거할 수 있는 사용자를 만들어야 합니다.

워크플로를 실행하도록 Transfer Family 서버를 구성하려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 서버를 선택한 다음 목록에서 서버 하나를 선택합니다. 이 서버가 SFTP 프로토콜을 지원하는지 확인하세요.
3. 서버의 세부 정보 페이지에서 추가 세부 정보 섹션까지 아래로 스크롤한 다음 편집을 선택합니다.
4. 추가 세부 정보 편집 페이지의 관리형 워크플로 섹션에서 워크플로를 선택하고 해당하는 실행 역할을 선택합니다.

   - 전체 파일 업로드를 위한 워크플로의 경우 2단계: 관리형 워크플로 생성에서 만든 워크플로를 선택합니다. 예: **w-1234abcd5678efghi**.
   - 관리형 워크플로 실행 역할의 경우 1단계: 실행 역할 구성에서 생성한 IAM 역할을 선택합니다.

5. 페이지 맨 아래로 스크롤하고 저장을 선택하여 변경 내용을 저장합니다.

사용 중인 서버의 ID를 기록해 둡니다. PGP 키를 저장하는 데 사용하는 AWS Secrets Manager 암호의 이름은 부분적으로 서버 ID를 기반으로 합니다.

워크플로를 트리거할 수 있는 사용자를 추가하려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 서버를 선택한 다음 복호화 워크플로에 사용할 서버를 선택합니다.
3. 서버 세부 정보 페이지에서 사용자 섹션으로 스크롤하여 사용자 추가를선택합니다.
4. 새 사용자의 경우 다음 세부 정보를 입력합니다.
   - 사용자 이름에 **`decrypt-user`**를 입력합니다.
   - 역할에서 서버에 액세스할 수 있는 사용자 역할을 선택합니다.
   - 홈 디렉터리의 경우 이전에 사용한 Amazon S3 버킷을 선택합니다. 예:`DOC-EXAMPLE-BUCKET`.
   - SSH 퍼블릭 키의 경우 보유한 프라이빗 키에 해당하는 퍼블릭 키를 붙여넣습니다. 자세한 내용은 서비스 관리 사용자를 위한 SSH 키 생성를 참조하세요.
5. 추가를 선택하여 새 사용자를 저장합니다.

이 서버의 Transfer Family 사용자 이름을 기록해 둡니다. 암호는 부분적으로 사용자 이름을 기반으로 합니다. 단순화를 위해 이 자습서에서는 서버의 모든 사용자가 사용할 수 있는 기본 암호를 사용합니다.

4단계: PGP 키 쌍 생성

지원되는 PGP 클라이언트 중 하나를 사용하여 PGP키 쌍을 생성합니다. 이 과정은 PGP 키 생성에 자세히 설명되어 있습니다.

PGP 키 쌍 생성

1. 이 자습서에서는 gpg(GnuPG) 버전 2.0.22 클라이언트를 사용하여 RSA를 암호화 알고리즘으로 사용하는 PGP 키 쌍을 생성할 수 있습니다. 이 클라이언트의 경우 다음 명령을 실행하고 이메일 주소와 암호를 입력합니다. 원하는 이름이나 이메일 주소를 사용할 수 있습니다. 자습서 뒷부분에서 값을 입력해야 하므로 사용하는 값을 기억해 두세요.

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

> **Note**
>
> GnuPG 버전 2.3.0 이상을 사용하는 경우 `gpg --full-gen-key`를 실행해야 합니다. 생성할 키 타입을 묻는 메시지가 표시되면 RSA 또는 ECC를 선택합니다. 하지만 ECC를 선택할 경우 타원 곡선에서 NIST 또는 BrainPool을 선택해야 합니다. Curve 25519를 선택하지 마세요.

2. 다음 명령을 실행하여 프라이빗 키를 내보냅니다. *user@example.com*을 키를 생성할 때 사용한 이메일 주소로 바꿉니다.

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

이 명령은 프라이빗 키를 **workflow-tutorial-key.pgp** 파일로 내보냅니다. 출력 파일의 이름을 원하는 대로 지정할 수 있습니다. 프라이빗 키를 AWS Secrets Manager에 추가한 후 프라이빗 키 파일을 삭제할 수도 있습니다.

5단계: PGP 프라이빗 키를 AWS Secrets Manager에 저장

매우 구체적인 방식으로 Secrets Manager에 프라이빗 키를 저장해야 워크플로가 업로드된 파일에 대한 복호화 단계를 실행할 때 워크플로가 프라이빗 키를 찾을 수 있습니다.

> **Note**
>
> Secrets Manager에 비밀을 저장하면 AWS 계정 요금이 발생합니다. 요금에 대한 자세한 내용은 AWS Secrets Manager 요금을 참조하세요.

Secrets Manager에 PGP 프라이빗 키를 저장하려면

1. https://console.aws.amazon.com/secretsmanager/ 에서 AWS Management Console 로그인하고 AWS Secrets Manager 콘솔을 여십시오.
2. 왼쪽 탐색 창에서 암호를 선택합니다.
3. 암호 페이지에서 새 암호 저장을 선택합니다.
4. 암호 선택 페이지의 암호 타입에서 다른 타입의 암호를 선택합니다.

5. 키/값 쌍 섹션에서 키/값 탭을 선택합니다.
   - 키 — **PGPPrivateKey**를 입력합니다.
   - 값 — 프라이빗 키의 텍스트를 값 필드에 붙여넣습니다.
6. 행 추가를 선택하고 키/값 쌍 섹션에서 키/값 탭을 선택합니다.
   - 키 — **PGPPassphrase**를 입력합니다.
   - 값 — 4단계: PGP 키 쌍 생성에서 PGP 키 쌍을 생성할 때 사용한 암호를 입력합니다.
7. 다음을 선택합니다.
8. 암호 구성 페이지에 암호를 위한 명칭과 설명을 입력합니다. 특정 사용자를 위한 비밀번호나 모든 사용자가 사용할 수 있는 비밀번호를 생성할 수 있습니다. 서버 ID가 *s-11112222333344445*인 경우 다음과 같이 암호 이름을 지정합니다.
   - 모든 사용자에 대한 기본 암호를 만들려면 암호의 이름을 **aws/transfer/*s-11112222333344445*/@pgp-default** 지정하십시오.
   - 이전에 생성한 사용자에 대한 암호만 생성하려면 암호 이름을 **aws/transfer/*s-11112222333344445*/decrypt-user**로 지정합니다.
9. 다음을 선택한 후 교체 구성 페이지의 기본값을 그대로 사용합니다. 그 다음 다음을 선택합니다.
10. 검토 페이지에서 저장을 선택하여 암호를 만들고 저장합니다.

Secrets Manager에 PGP 개인 키를 추가하는 방법에 대한 자세한 내용은 PGP 키 AWS Secrets Manager 저장에 사용을 참조하십시오.

6단계: 파일 암호화

gpg 프로그램을 사용하여 워크플로에 사용할 파일을 암호화합니다. 파일을 암호화하려면 다음 명령을 실행합니다.

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

이 자동화를 실행하기 전에 다음 사항에 유의하세요.

- `-r` 인수의 경우 *marymajor@example.com*을 PGP 키 쌍을 만들 때 사용한 이메일 주소로 바꿉니다.
- `--openpgp` 플래그는 옵션입니다. 이 플래그는 암호화된 파일이 OpenPGP RFC4880 표준을 준수하도록 합니다.

- 이 명령은 **testfile.txt**와 같은 위치에 이름이 **testfile.txt.gpg**라고 지정된 파일을 만듭니다.

7단계: 워크플로 실행 및 결과 보기

워크플로를 실행하려면 3단계에서 만든 사용자를 사용하여 Transfer Family 서버에 연결합니다. 그런 다음 2.5단계에서 지정한 Amazon S3 버킷을 살펴보고 복호화된 파일을 볼 수 있도록 대상 파라미터를 구성할 수 있습니다.

복호화 워크플로를 실행하려면

1. 명령 터미널을 엽니다.
2. 실제 엔드포인트를 *your-endpoint*로 대체하고 *transfer-key*를 사용자의 SSH 프라이빗 키로 대체하여 다음 명령을 실행합니다.

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

   예를 들어 프라이빗 키가 ~/.ssh/decrypt-user에 저장되어 있고 엔드포인트가 s-11112222333344445.server.transfer.us-east-2.amazonaws.com인 경우, 명령은 다음과 같습니다.

   ```
   sftp -i  ~/.ssh/decrypt-user decrypt-user@s-11112222333344445.server.transfer.us-east-2.amazonaws.com
   ```
3. pwd 명령을 실행합니다. 이 명령이 제대로 실행되면 다음과 같은 응답을 반환합니다.

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

   디렉터리에는 Amazon S3 버킷 이름이 반영됩니다.
4. 다음 명령을 실행하여 파일을 업로드하고 워크플로를 실행하도록 트리거합니다.

   ```
   put testfile.txt.gpg
   ```
5. 복호화된 파일의 대상 폴더는 워크플로를 만들 때 지정한 decrypted-files/ 폴더입니다. 이제 해당 폴더로 이동하여 내용을 나열할 수 있습니다.

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

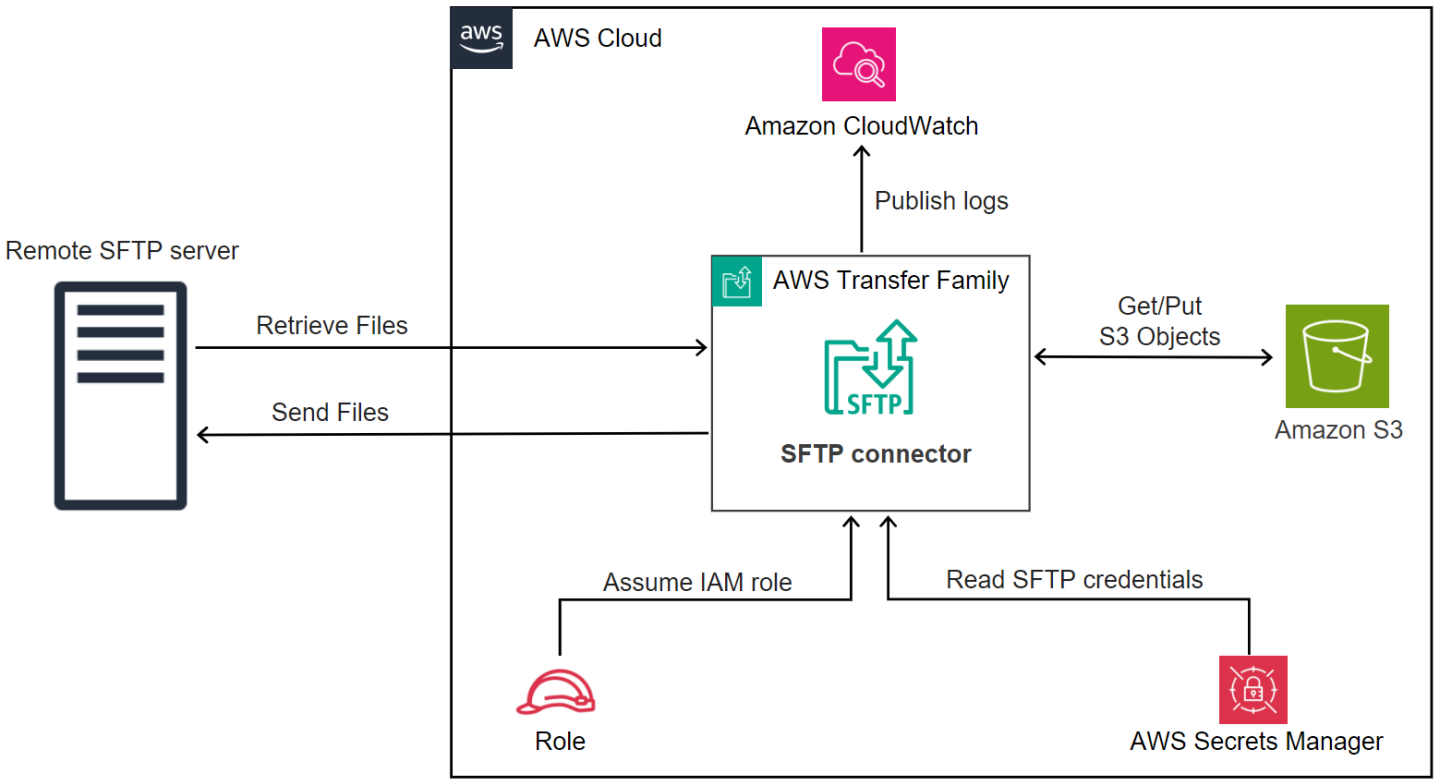
성공하면 `ls` 명령이 `testfile.txt` 파일을 나열합니다. 이 파일을 다운로드하여 이전에 암호화한 원본 파일과 동일한지 확인할 수 있습니다.

SFTP 커넥터 설정 및 사용

커넥터의 목적은 AWS 스토리지와 파트너의 SFTP 서버 간의 관계를 설정하는 것입니다. Amazon S3에서 파트너 소유의 외부 대상으로 파일을 보낼 수 있습니다. SFTP 커넥터를 사용하여 파트너의 SFTP 서버에서 파일을 검색할 수도 있습니다.

이 자습서는 SFTP 커넥터를 설정한 다음 Amazon S3 스토리지와 SFTP 서버 간에 파일을 전송하는 방법을 보여줍니다.

SFTP 커넥터는 원격 SFTP 서버에 인증하고 연결을 AWS Secrets Manager 설정하기 위해 SFTP 자격 증명을 검색합니다. 커넥터는 원격 서버로 파일을 보내거나 원격 서버에서 파일을 검색하고 Amazon S3에 파일을 저장합니다. IAM 역할은 Amazon S3 버킷 및 Secrets Manager에 저장된 자격 증명에 대한 액세스를 허용하는 데 사용됩니다. 그리고 Amazon에 로그인할 수 CloudWatch 있습니다.



다음 블로그 게시물은 SFTP 커넥터를 사용하여 원격 SFTP 서버로 파일을 전송하기 전에 PGP를 사용하여 파일을 암호화하는 것을 포함하여 SFTP 커넥터를 사용하여 MFT 워크플로를 구축하는 참조 아키

텍처를 제공합니다. SFTP 커넥터 및 PGP 암호화로 안전하고 규정을 준수하는 관리형 파일 전송 설계 AWS Transfer Family

주제

- 1단계: 필요한 지원 리소스 만들기
- 2단계: SFTP 커넥터 생성 및 테스트
- 3단계: SFTP 커넥터를 사용하여 파일 전송 및 검색
- 원격 SFTP 서버로 사용할 Transfer Family 서버를 만드는 절차

1단계: 필요한 지원 리소스 만들기

SFTP 커넥터를 사용하여 Amazon S3와 모든 원격 SFTP 서버 간에 파일을 복사할 수 있습니다. 이 자습서에서는 서버를 원격 SFTP AWS Transfer Family 서버로 사용합니다. 다음 리소스를 만들고 구성해야 합니다.

- Amazon S3 버킷을 생성하여 사용자 AWS 환경에 파일을 저장하고 원격 SFTP 서버에서 파일을 전송 및 검색하십시오. Amazon S3 버킷 생성
- Secrets Manager:에서 Amazon S3 스토리지에 액세스하기 위한 AWS Identity and Access Management 역할과 보안 정보를 생성합니다필요한 권한을 가진 IAM 역할을 생성합니다..
- SFTP 프로토콜을 사용하는 Transfer Family 서버와 SFTP 커넥터를 사용하여 SFTP 서버와 파일을 주고받는 서비스 관리 사용자를 생성합니다. Transfer Family SFTP 서버 및 사용자 생성
- SFTP 커넥터가 원격 SFTP 서버에 로그인할 때 사용하는 자격 증명을 저장하는 AWS Secrets Manager 암호를 생성하십시오. 시크릿 생성 및 저장 AWS Secrets Manager

Amazon S3 버킷 생성

Amazon S3 버킷을 생성하려면

1. https://console.aws.amazon.com/s3/ 에서 AWS Transfer Family 콘솔에 로그인합니다.
2. 지역을 선택하고 이름을 입력합니다.

   이 자습서에서는 버킷이 **US East (N. Virginia) us-east-1** 들어 있으며 이름은 입니다**sftp-server-storage-east**.
3. 기본값을 그대로 사용하고 버킷 생성을 선택합니다.

Amazon S3 버킷 생성에 대한 자세한 내용은 S3 버킷을 생성하려면 어떻게 해야 합니까? 를 참조하십시오. Amazon 심플 스토리지 서비스 사용 설명서에서 확인할 수 있습니다.

필요한 권한을 가진 IAM 역할을 생성합니다.

액세스 역할의 경우 다음 권한이 포함된 정책을 생성합니다.

다음 예제는 Amazon S3의 *DOC-EXAMPLE-BUCKET#* 액세스하는 데 필요한 권한을 부여하고 Secrets Manager에 저장된 지정된 시크릿을 부여합니다.

```
{
  "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"
    }
  ]
}
```

다음과 같이 항목을 교체하십시오.

- *DOC-EXAMPLE-BUCKET# ## ###### ### #####*. **s3-storage-east**
- 자습서에서는 *### ## #####*. **us-east-1**
- *## ID# ## ID#* 사용하세요. AWS 계정
- *SecretNameRandomCharacters-6#* **using sftp-connector1** 경우 이름을 입력합니다 (비밀번호는 무작위로 6자 입력해야 합니다).

또한 사용자의 전송 요청을 처리할 때 커넥터가 리소스에 액세스할 수 있도록 하는 신뢰 관계가 이 역할에 포함되어 있는지 확인해야 합니다. 신뢰 관계 설정에 대한 자세한 내용은 신뢰 관계를 구축하기 위해을 참조하세요.

> **Note**
>
> 자습서에서 사용하는 역할의 세부 정보를 보려면 을 참조하십시오. 사용자 및 액세스 역할 통합

시크릿 생성 및 저장 AWS Secrets Manager

SFTP 커넥터의 사용자 자격 증명을 저장하려면 Secrets Manager에 시크릿을 저장해야 합니다. 암호, SSH 개인 키 또는 둘 다를 사용할 수 있습니다. 자습서에서는 개인 키를 사용하고 있습니다.

> **Note**
>
> Secrets Manager에 비밀을 저장하면 AWS 계정 요금이 발생합니다. 요금에 대한 자세한 내용은 AWS Secrets Manager 요금을 참조하세요.

비밀 저장 절차를 시작하기 전에 개인 키를 검색하고 형식을 지정하십시오. 개인 키는 원격 SFTP 서버의 사용자에 대해 구성된 공개 키와 일치해야 합니다. 자습서에서 개인 키는 원격 서버로 사용 중인 Transfer Family SFTP 서버에 테스트 사용자를 위해 저장된 공개 키와 일치해야 합니다.

이렇게 하려면 다음 명령을 실행합니다.

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

예를 들어 프라이빗 키 파일이 에 `~/.ssh/sftp-testuser-privatekey` 있는 경우 명령은 다음과 같습니다.

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

그러면 키가 올바른 형식 (포함된 줄 바꿈 문자 포함) 으로 표준 출력에 출력됩니다. 다음 절차 (6단계) 에 따라 붙여넣어야 하므로 이 텍스트를 어딘가에 복사하십시오.

SFTP 커넥터의 사용자 자격 증명을 Secrets Manager에 저장하려면

1. 에 AWS Management Console 로그인하고 https://console.aws.amazon.com/secretsmanager/ 에서 AWS Secrets Manager 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 암호를 선택합니다.
3. 암호 페이지에서 새 암호 저장을 선택합니다.
4. 암호 선택 페이지의 암호 타입에서 다른 타입의 암호를 선택합니다.
5. 키/값 쌍 섹션에서 키/값 탭을 선택합니다.
   - 키 - **`Username`**를 입력합니다.
   - 값 — 사용자 이름, 을 입력합니다**`sftp-testuser`**.
6. 키를 입력하려면 일반 텍스트 탭을 사용하는 것이 좋습니다.
   a. 행 추가를 선택한 다음 입력하십시오. **`PrivateKey`**
   b. 일반 텍스트 탭을 선택합니다. 이제 필드에 다음 텍스트가 포함됩니다.

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

   c. 이전에 저장한 개인 키 텍스트를 빈 큰따옴표 (“”) 사이에 붙여넣습니다.

      화면은 다음과 같아야 합니다 (키 데이터는 회색으로 표시됨).

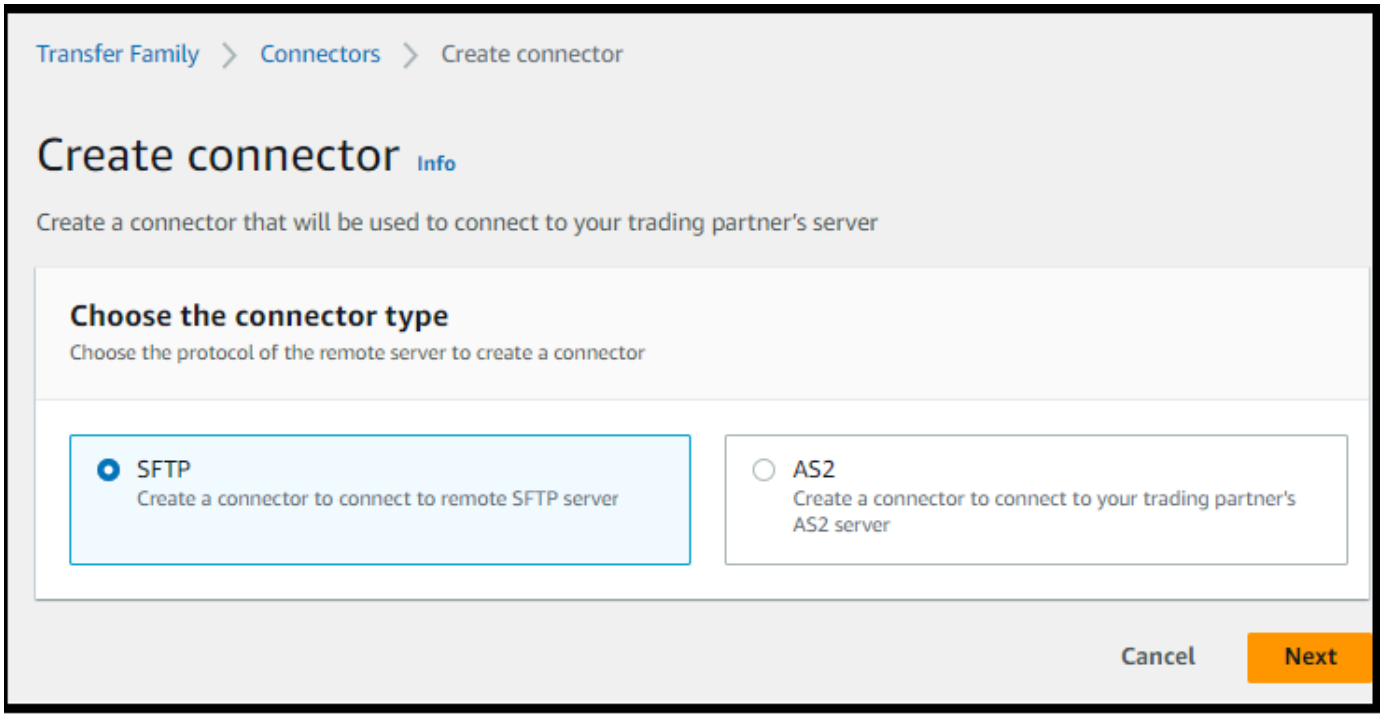
7. 다음을 선택합니다.
8. 암호 구성 페이지에서 암호 이름을 입력합니다. 이 자습서에서는 암호의 이름을 지정합니다**`aws/transfer/sftp-connector1`**.
9. 다음을 선택한 후 교체 구성 페이지의 기본값을 그대로 사용합니다. 그 다음 다음을 선택합니다.
10. 검토 페이지에서 저장을 선택하여 암호를 만들고 저장합니다.

2단계: SFTP 커넥터 생성 및 테스트

이 섹션에서는 앞서 만든 모든 리소스를 사용하는 SFTP 커넥터를 만듭니다. 자세한 내용은 SFTP 커넥터 구성를 참조하세요.

SFTP 커넥터를 생성하려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 커넥터를 선택한 다음 커넥터 생성을 선택합니다.
3. 커넥터 타입으로 SFTP를 선택하여 SFTP 커넥터를 생성한 후 다음을 선택합니다.



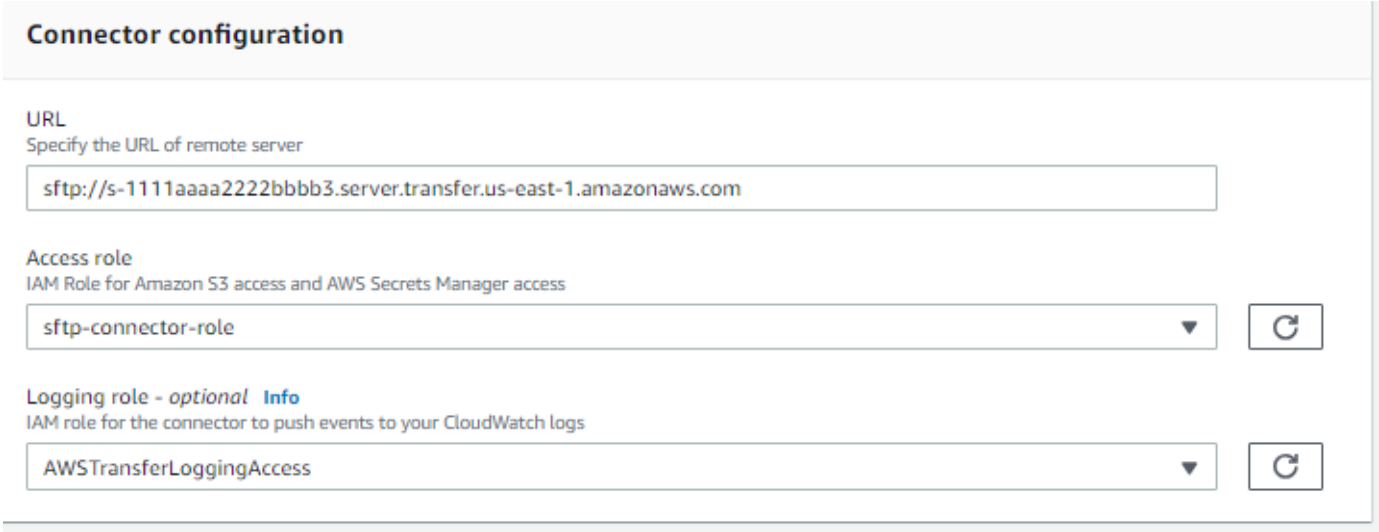
4. 커넥터 구성 섹션에서 다음 정보를 제공합니다:

   - URL에는 원격 SFTP 서버의 URL을 입력합니다. 자습서에서는 원격 SFTP 서버로 사용하고 있는 Transfer Family 서버의 URL을 입력합니다.

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

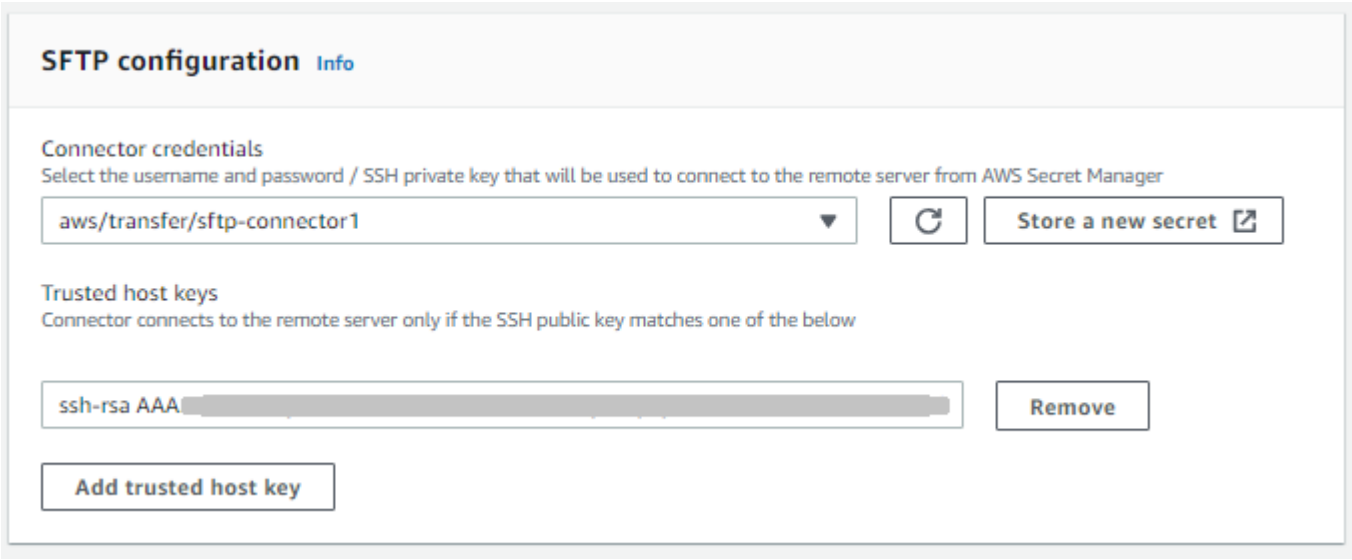
     1111aaaa2222bbbb3# Transfer Family 서버 ID로 바꾸십시오.
   - 액세스 역할에는 앞서 생성한 역할을 입력합니다. **sftp-connector-role**
   - 로깅 역할에서 선택합니다**AWSTransferLoggingAccess**.

     > **Note**
     >
     > AWSTransferLoggingAccess AWS 관리형 정책입니다. 이 정책은 에 자세히 설명되어 AWS 관리형 정책: AWSTransferLoggingAccess 있습니다.



5. SFTP 구성 섹션에 다음 정보를 제공합니다.

   - 커넥터 자격 증명의 경우 SFTP 자격 증명이 포함된 Secrets Manager 리소스의 이름을 선택합니다. 자습서를 보려면 선택하십시오**`aws/transfer/sftp-connector1`**.
   - 신뢰할 수 있는 호스트 키의 경우 호스트 키의 공개 부분을 붙여넣습니다. SFTP 서버에서 실행하여 이 키를 `ssh-keyscan` 검색할 수 있습니다. 신뢰할 수 있는 호스트 키를 포맷하고 저장하는 방법에 대한 자세한 내용은 SftpConnectorConfig데이터 유형 설명서를 참조하십시오.



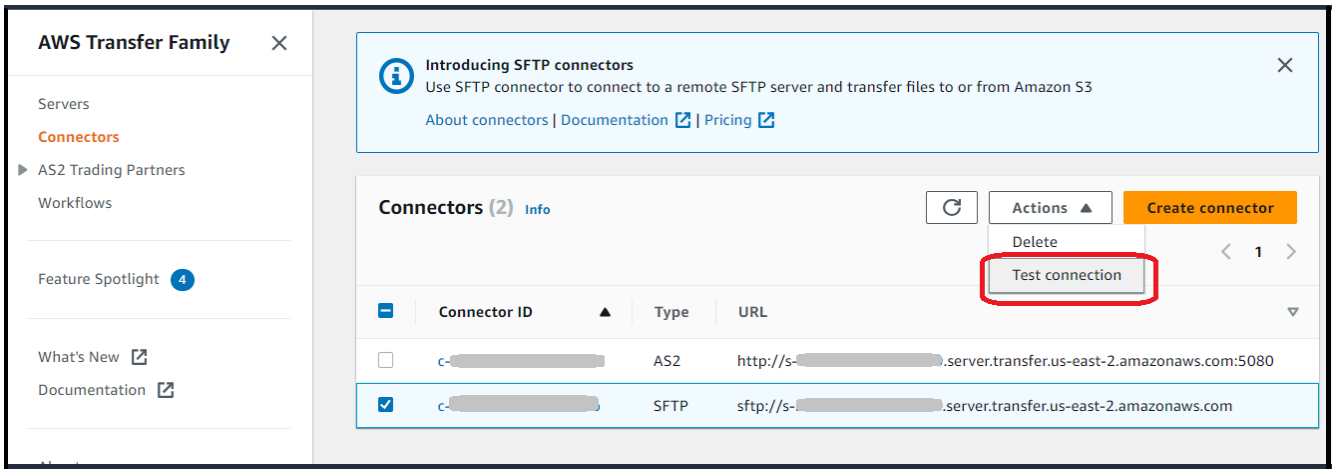
6. 모든 설정을 확인한 후 커넥터 생성을 선택하여 SFTP 커넥터를 생성합니다.

SFTP 커넥터를 만든 후에는 새 커넥터를 사용하여 파일을 전송하기 전에 테스트해 보는 것이 좋습니다.

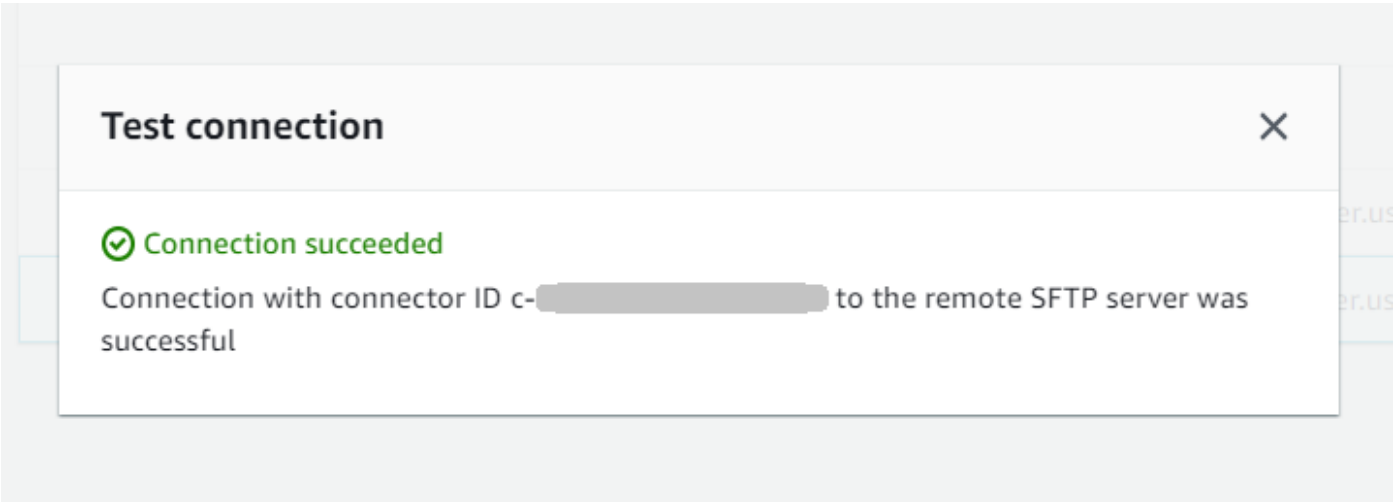
Test a connector using the console

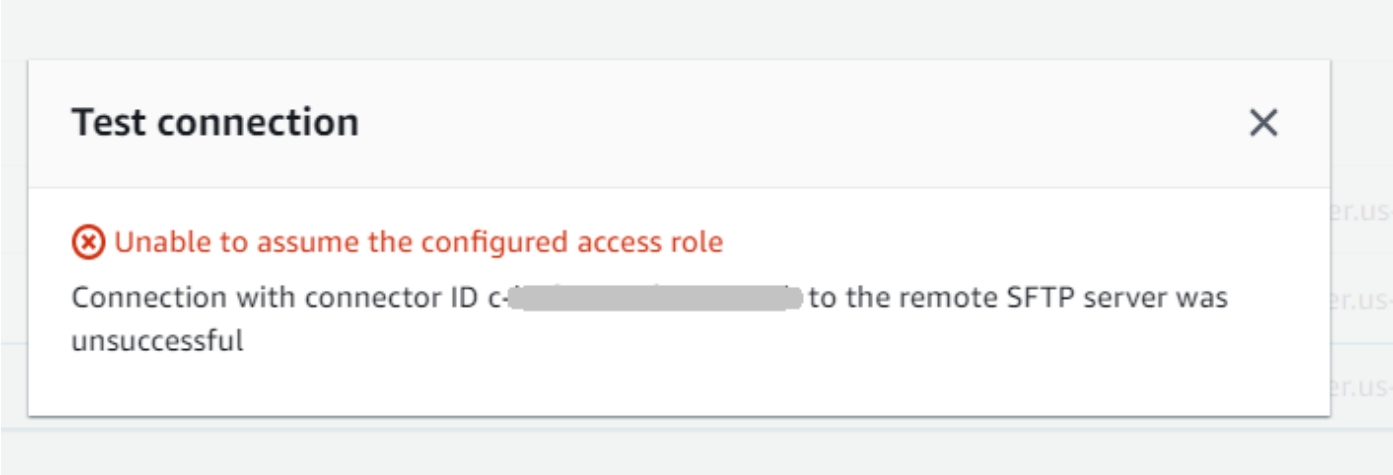
SFTP 커넥터를 테스트하려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 커넥터를 선택하고 커넥터를 선택합니다.
3. 작업 메뉴에서 테스트 연결을 선택합니다.



시스템은 테스트의 통과 또는 실패 여부를 나타내는 메시지를 반환합니다. 테스트가 실패하면 시스템은 테스트 실패 이유를 기반으로 오류 메시지를 제공합니다.





Test a connector using the CLI

를 사용하여 커넥터를 테스트하려면 명령 프롬프트에서 다음 명령을 실행합니다 (*connector-id # ## ### ID#* 대체). AWS Command Line Interface

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

테스트가 성공하면 다음 줄이 반환됩니다.

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

테스트가 실패하면 설명이 포함된 오류 메시지가 나타납니다. 예를 들면 다음과 같습니다.

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

3단계: SFTP 커넥터를 사용하여 파일 전송 및 검색

단순화를 위해 Amazon S3 버킷에 이미 파일이 있다고 가정합니다.

> **Note**
>
> 이 자습서에서는 원본 및 대상 스토리지 위치 모두에 Amazon S3 버킷을 사용합니다. SFTP 서버가 Amazon S3 스토리지를 사용하지 않는 경우, 다음 `sftp-server-storage-east` 명령에서 볼 수 있는 모든 위치에서 경로를 SFTP 서버에서 액세스할 수 있는 파일 위치의 경로로 바꿀 수 있습니다.

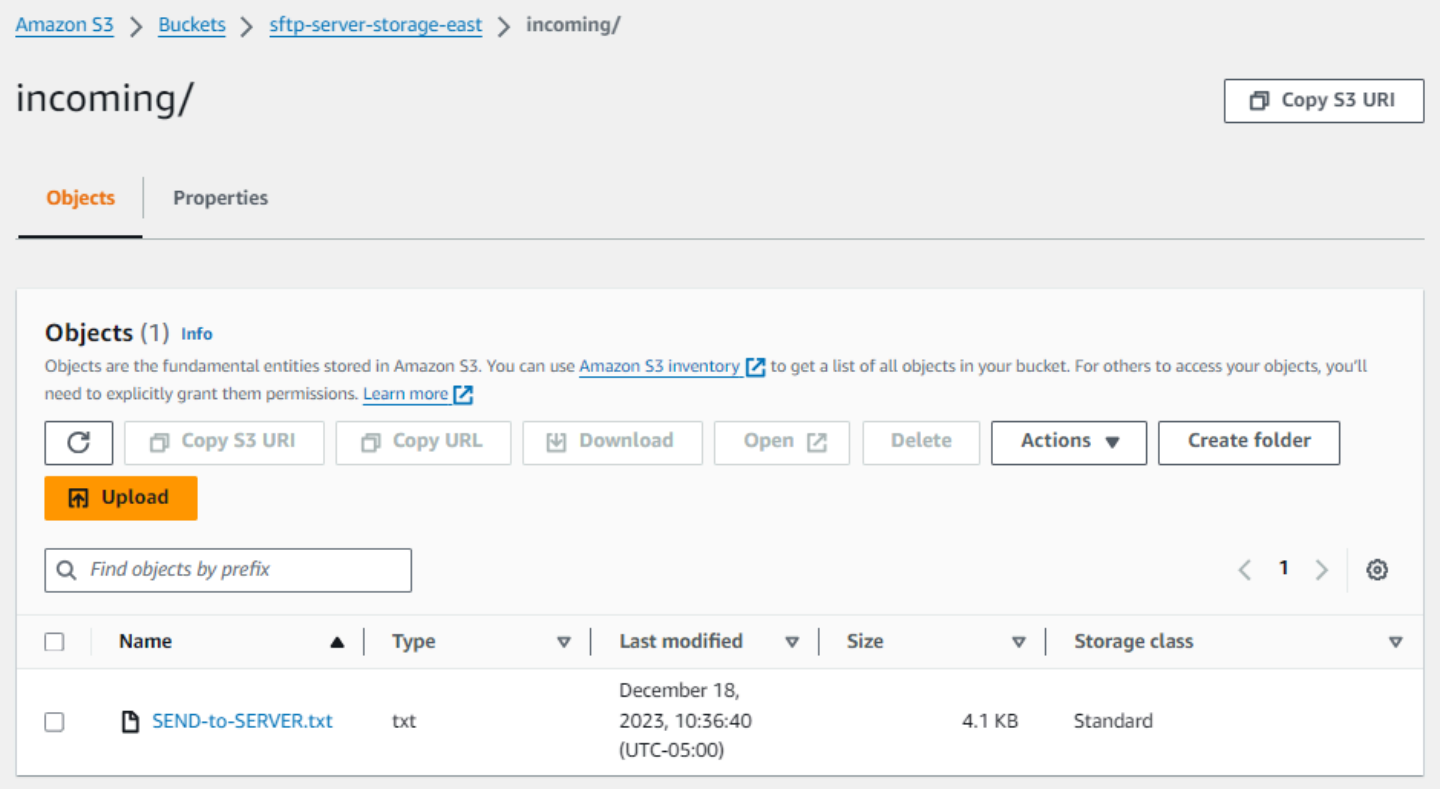
- Amazon S3 `SEND-to-SERVER.txt` 스토리지에서 이름이 지정된 파일을 SFTP 서버로 전송합니다.
- SFTP 서버에서 Amazon S3 스토리지로 이름이 지정된 `RETRIEVE-to-S3.txt` 파일을 검색합니다.

> **Note**
>
> 다음 명령에서 *`###-ID# ### ID#`* 대체하십시오.

먼저 Amazon S3 버킷에서 원격 SFTP 서버로 파일을 전송합니다. 명령 프롬프트에서 다음 명령을 실행합니다.

```
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"
```

이제 `sftp-server-storage-east` 버킷이 다음과 같이 보일 것입니다.



파일이 예상대로 표시되지 않으면 CloudWatch 로그를 확인하세요.

CloudWatch 로그를 확인하려면

1. https://console.aws.amazon.com/cloudwatch/ 에서 아마존 CloudWatch 콘솔을 엽니다
2. 왼쪽 탐색 메뉴에서 로그 그룹을 선택합니다.
3. 로그를 찾으려면 검색 창에 커넥터 ID를 입력합니다.
4. 검색에서 반환된 로그 스트림을 선택합니다.
5. 가장 최근 로그 항목을 확장합니다.

성공하면 로그 항목이 다음과 같이 표시됩니다.

```
{
    "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",
```

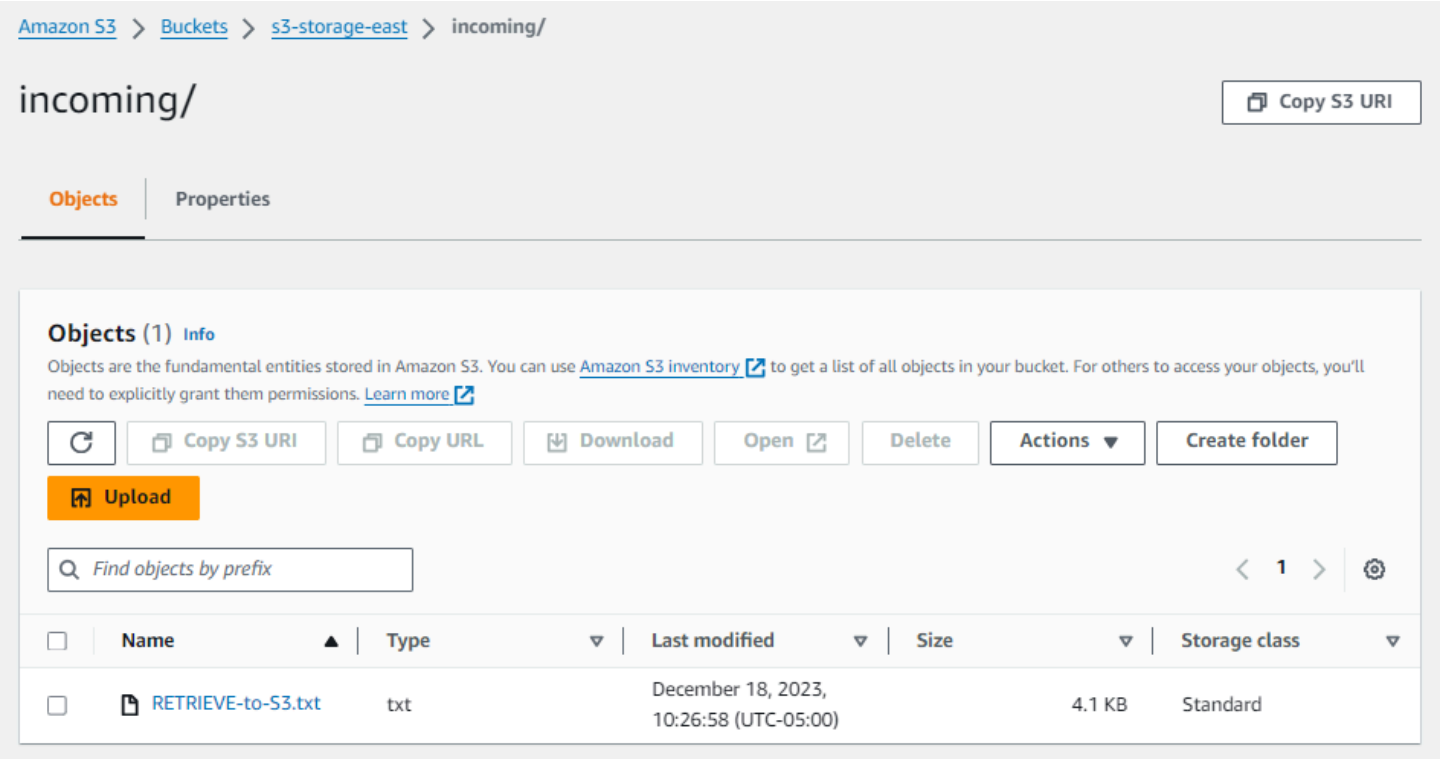
```
    "status-code": "COMPLETED",
    "start-time": "2023-12-18T15:26:56.915864Z",
    "end-time": "2023-12-18T15:26:57.298122Z",
    "account-id": "500655546075",
    "connector-arn": "arn:aws:transfer:us-east-1:500655546075:connector/connector-id",
    "remote-directory-path": "/sftp-server-storage-east/incoming"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

파일 전송이 실패한 경우 로그 항목에 문제를 지정하는 오류 메시지가 포함됩니다. 오류의 일반적인 원인은 IAM 권한 문제와 잘못된 파일 경로입니다.

다음으로 SFTP 서버에서 Amazon S3 버킷으로 파일을 검색합니다. 명령 프롬프트에서 다음 명령을 실행합니다.

```
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"
```

전송이 성공하면 아래와 같이 Amazon S3 버킷에 전송된 파일이 포함됩니다.



성공하면 로그 항목이 다음과 같이 표시됩니다.

```
{
```

```
    "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"
}
```

원격 SFTP 서버로 사용할 Transfer Family 서버를 만드는 절차

다음은 이 자습서에서는 원격 SFTP 서버 역할을 하는 Transfer Family 서버를 만드는 단계를 간략하게 설명합니다. 유의할 사항:

- Transfer Family 서버를 사용하여 원격 SFTP 서버를 나타냅니다. 일반적인 SFTP 커넥터 사용자는 자체 원격 SFTP 서버를 가지고 있습니다. Transfer Family SFTP 서버 및 사용자 생성를 참조하세요.
- Transfer Family 서버를 사용하고 있기 때문에 서비스 관리 SFTP 사용자도 사용하고 있습니다. 또한 단순화를 위해 이 사용자가 Transfer Family 서버에 액세스하는 데 필요한 권한과 커넥터를 사용하는 데 필요한 권한을 결합했습니다. 다시 말하지만, 대부분의 SFTP 커넥터 사용 사례에는 Transfer Family 서버와 연결되지 않은 별도의 SFTP 사용자가 있습니다. Transfer Family SFTP 서버 및 사용자 생성를 참조하세요.
- 자습서에서는 원격 SFTP 서버에 Amazon S3 스토리지를 사용하고 있으므로 한 버킷에서 다른 버킷으로 파일을 전송할 수 있도록 두 번째 버킷을 만들어야 합니다. **`s3-storage-east`**

Transfer Family SFTP 서버 및 사용자 생성

이미 사용자와 함께 SFTP 서버를 보유하고 있으므로 대부분의 사용자는 Transfer Family SFTP 서버와 사용자를 만들 필요가 없으며 이 서버를 사용하여 파일을 주고 받을 수 있습니다. 그러나 이 자습서에서는 단순화를 위해 Transfer Family 서버를 사용하여 원격 SFTP 서버로 작동합니다.

SFTP 지원 서버 생성에 설명된 절차에 따라 서버를 만들고 사용자를 3단계: 서비스 관리 사용자 추가 추가합니다. 자습서에서 사용하는 사용자 세부 정보는 다음과 같습니다.

- 서비스 관리 사용자 생성,. `sftp-testuser`
  - 홈 디렉터리를 다음으로 설정합니다. `/sftp-server-storage-east/sftp-testuser`
  - 사용자를 생성할 때 공개 키를 저장합니다. 나중에 Secrets Manager에서 시크릿을 생성할 때 해당 프라이빗 키를 제공해야 합니다.
- 역할:`sftp-connector-role`. 이 자습서에서는 SFTP 사용자와 SFTP 커넥터 액세스에 모두 동일한 IAM 역할을 사용하고 있습니다. 조직을 위한 커넥터를 생성할 때 별도의 사용자 및 액세스 역할을 가질 수 있습니다.
- 서버 호스트 키: 커넥터를 만들 때는 서버 호스트 키를 사용해야 합니다. 서버에 `ssh-keyscan` 대해 실행하여 이 키를 검색할 수 있습니다. 예를 들어, 서버 ID가 이고 `s-1111aaaa2222bbbb3` 엔드포인트가 인 `us-east-1` 경우 다음 명령은 서버 호스트 키를 검색합니다.

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

  이 텍스트를 2단계: SFTP 커넥터 생성 및 테스트 프로시저에 붙여넣어야 하므로 어딘가에 복사하십시오.

사용자 및 액세스 역할 통합

이 자습서에서는 하나의 결합된 역할을 사용하고 있습니다. 이 역할은 SFTP 사용자와 커넥터 액세스 모두에 사용됩니다. 다음 예제에는 자습서의 작업을 수행하려는 경우를 대비하여 이 역할에 대한 세부 정보가 포함되어 있습니다.

다음 예제는 Amazon S3에 있는 두 개의 버킷에 액세스하는 데 필요한 권한을 부여하고 Secrets Manager에 `aws/transfer/sftp-connector1` 저장된 암호라는 이름을 부여합니다. 자습서에서는 이 역할의 이름을 `sftp-connector-role` 지정했습니다.

```
{
    "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"
        }
    ]
}
```

Transfer Family의 역할 생성에 대한 자세한 내용을 사용자 역할 생성 보려면 역할 생성에 설명된 절차를 따르십시오.

Amazon API Gateway 메서드를 사용자 지정 ID 공급자로 설정

이 자습서에서는 Amazon API Gateway 메서드를 설정하고 이를 사용자 지정 ID 공급자로 사용하여 AWS Transfer Family 서버에 파일을 업로드하는 방법을 보여줍니다. 이 자습서에서는 기본 스택 템플릿과 기타 기본 기능을 예로만 사용합니다.

주제

필수 조건

에서 AWS CloudFormation Transfer Family 리소스를 생성하기 전에 스토리지와 사용자 역할을 생성하십시오.

스토리지를 지정하고 사용자 역할을 생성하려면

1. 사용 중인 스토리지에 따라 다음 설명서를 참조하세요.
   - Amazon S3 버킷을 생성하려면 Amazon Simple Storage Service 사용 설명서의 S3 버킷을 생성하려면 어떻게 해야 합니까? 단원을 참조하세요.
   - Amazon EFS 파일 시스템을 생성하려면 을 참조하십시오Amazon EFS 파일 시스템 구성.
2. 사용자 역할을 생성하려면 IAM 역할 및 정책 생성 단원을 참조하세요.

다음 섹션에서 AWS CloudFormation 스택을 생성할 때 스토리지의 세부 정보와 사용자 역할을 입력합니다.

1단계: CloudFormation 스택 생성

제공된 템플릿으로 AWS CloudFormation 스택을 만들려면

1. https://console.aws.amazon.com/cloudformation 에서 AWS CloudFormation 콘솔을 엽니다.
2. 스택 생성을 선택한 다음 새 리소스 사용(표준)을 선택합니다.
3. 사전 조건 - 템플릿 준비 창에서 템플릿 준비 완료를 선택합니다.
4. 기본 스택 템플릿인 이 링크를 복사하여 Amazon S3 URL 필드에 붙여넣습니다.
5. 다음을 클릭합니다.

6. 스택 이름을 비롯한 파라미터를 지정합니다. 다음 작업을 수행합니다.

   - UserName및 의 기본값을 바꿉니다 UserPassword.
   - 에 UserHomeDirectory앞서 생성한 스토리지 (Amazon S3 버킷 또는 Amazon EFS 파일 시스템)의 세부 정보를 입력합니다.
   - 기본값을 이전에 UserRoleArn생성한 사용자 역할로 바꾸십시오. AWS Identity and Access Management (IAM) 역할에는 적절한 권한이 있어야 합니다. IAM 역할 및 버킷 정책 예는 6단계: 버킷에 대한 액세스 제한 단원을 참조하세요.
   - 비밀번호 대신 퍼블릭 키를 사용하여 인증하려면 UserPublicKey1 필드에 퍼블릭 키를 입력합니다. SFTP를 사용하여 서버에 처음 연결할 때 암호 대신 프라이빗 키를 입력합니다.
7. 다음을 선택한 후 스택 옵션 구성 페이지에서 다시 다음을 선택합니다.
8. 생성 중인 스택의 세부 정보를 검토한 다음 스택 생성을 선택합니다.

> **Note**
>
> 페이지 하단의 기능에서 AWS CloudFormation 이 IAM 리소스를 생성할 수도 있음을 인정해야 합니다.

2단계: 서버의 API Gateway 메서드 구성을 확인합니다.

> **Note**
>
> 보안을 강화하기 위해 웹 애플리케이션 방화벽을 구성할 수 있습니다. AWS WAF 은(는) Amazon API Gateway에 전달되는 HTTP 및 HTTPS 요청을 모니터링할 수 있게 해주는 웹 애플리케이션 방화벽입니다. 자세한 내용은 웹 애플리케이션 방화벽 추가를 참조하세요.

서버의 API Gateway 메서드 구성을 확인하고 배포하려면

1. https://console.aws.amazon.com/apigateway/에서 Amazon API Gateway 콘솔을 엽니다.
2. 템플릿에서 생성한 사용자 지정 ID 공급자 이전 기본 템플릿 API를 선택합니다. AWS CloudFormation
3. 리소스 창에서 GET을 선택한 후 메서드 요청을 선택합니다.
4. 작업 및 API 배포를 선택합니다. 배포 단계에서 prod를 선택한 다음 배포를 선택합니다.

API Gateway 메서드가 성공적으로 배포되면 단계 편집기 섹션에서 성능을 확인합니다.

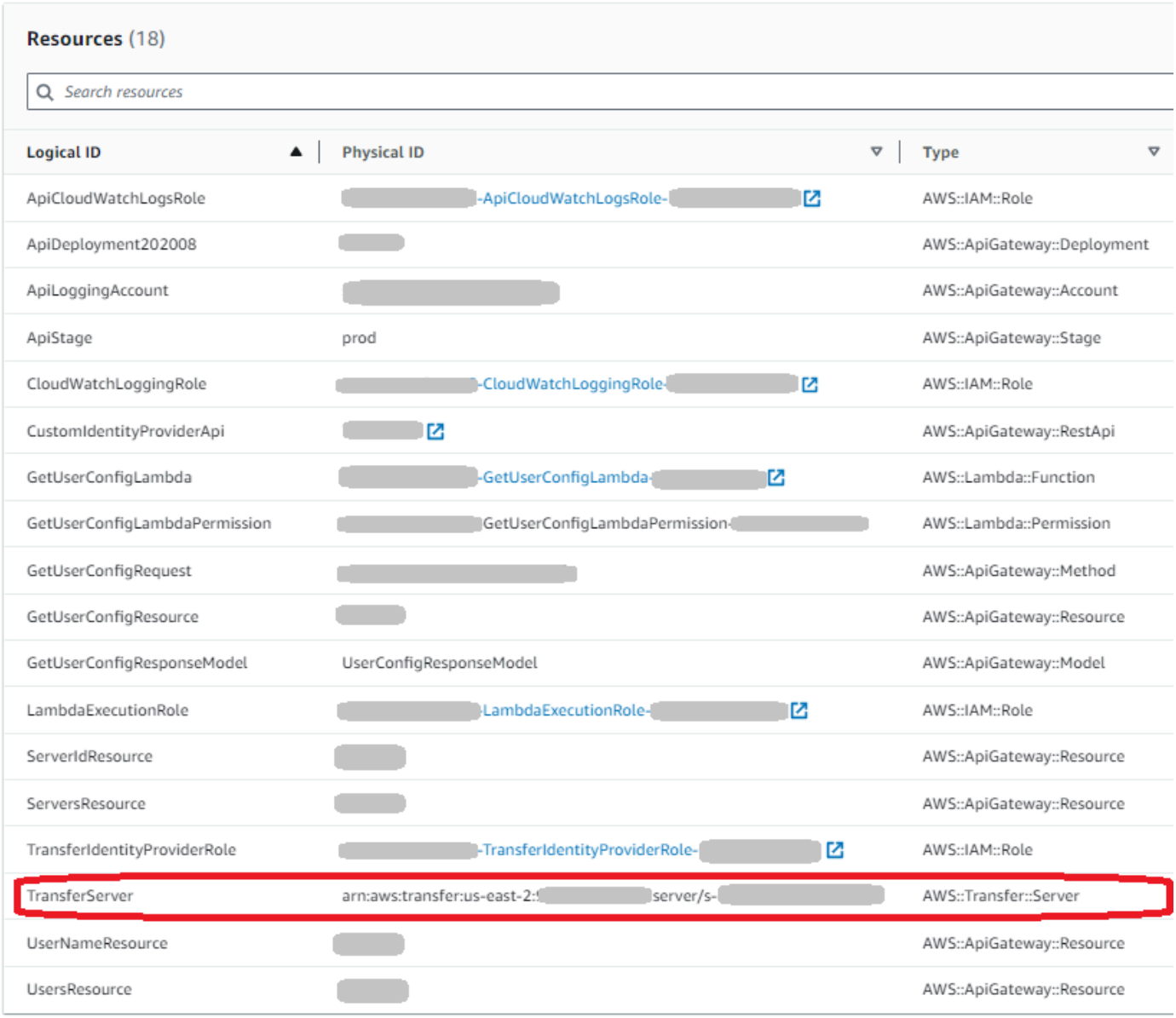
> **Note**
>
> 페이지 상단에 표시되는 URL 호출 주소를 복사합니다. 이 정보는 다음 단계에 필요합니다.

3단계: Transfer Family 서버 세부 정보 보기

템플릿을 사용하여 AWS CloudFormation 스택을 생성하면 Transfer Family 서버가 자동으로 생성됩니다.

Transfer Family 서버 세부 정보 보기

1. https://console.aws.amazon.com/cloudformation 에서 AWS CloudFormation 콘솔을 엽니다.
2. 생성한 스택을 선택합니다.
3. 리소스 탭을 선택합니다.



서버 ARN은 해당 행의 물리적 ID 열에 표시됩니다. TransferServer 서버 ID는 ARN에 포함되어 있습니다(예: s-11112222333344445).

4. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 열고 서버 페이지에서 새 서버를 선택합니다.

   서버 ID는 의 TransferServer리소스에 대해 표시된 ID와 AWS CloudFormation일치합니다.

4단계: 사용자가 서버에 연결할 수 있는지 테스트

Transfer Family 콘솔을 사용하여 사용자가 서버에 연결할 수 있는지 테스트하려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 서버 페이지에서 새 서버를 선택하고 작업을 선택한 다음 테스트를 선택합니다.
3. 사용자 이름 필드와 암호 필드에 로그인 자격 증명의 텍스트를 입력합니다. AWS CloudFormation 스택을 배포할 때 설정한 값입니다.
4. 서버 프로토콜에 대해 SFTP를 선택하고 소스 IP에 **`127.0.0.1`**을 입력합니다.
5. 테스트를 선택합니다.

   사용자 인증에 성공하면 테스트에서 `StatusCode: 200` HTML 응답과 사용자 역할 및 권한의 세부 정보가 포함된 JSON 개체가 반환됩니다. 예:

   ```
   {
       "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"
   }
   ```

   테스트가 실패할 경우 API Gateway AWS 관리형 정책 중 하나를 API에 사용 중인 역할에 추가하십시오.

5단계: SFTP 연결 및 파일 전송 테스트

SFTP 연결을 테스트하려면

1. Linux 또는 macOS 장치에서 명령 터미널을 엽니다.
2. 인증에 암호를 사용하는지 또는 키 쌍을 사용하는지 여부에 따라 다음 명령 중 하나를 입력합니다.

   - 암호를 사용하는 경우 이 명령 입력:

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

메시지가 표시되면 암호를 입력합니다.

- 키 쌍을 사용하는 경우 이 명령 입력:

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

> **Note**
>
> 이러한 `sftp` 명령에 Transfer Family 서버가 위치해 있는 AWS 리전 에 해당하는 코드를 입력합니다. 예를 들어 서버가 미국 동부(오하이오)에 있는 경우 **us-east-2**를 입력합니다.

3. `sftp>` 프롬프트에서 디렉터리와 파일(`pwd` 및 `ls`)을 업로드(`put`)하고 다운로드(`get`)하고 볼 수 있는지 확인합니다.

6단계: 버킷에 대한 액세스 제한

특정 Amazon S3 버킷에 액세스할 수 있는 사용자를 제한할 수 있습니다. 다음 예는 CloudFormation 스택과 사용자에 대해 선택한 정책에서 사용할 설정을 보여줍니다.

이 예시에서는 AWS CloudFormation 스택에 다음과 같은 파라미터를 설정합니다.

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

> **Important**
>
> 다음은 암호의 예시입니다. API Gateway 메서드를 구성할 때는 강력한 암호를 입력해야 합니다.

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

UserPublicKey1은 공개/개인 키 쌍의 일부로 생성한 공개 키입니다.

*role-id*은(는) 생성한 사용자 역할에 따라 다릅니다. myuser-api-gateway-role에 연결된 정책은 다음과 같습니다.

```
{
    "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/*"
        }
    ]
}
```

SFTP를 사용하여 서버에 연결하려면 프롬프트에 다음 명령 중 하나를 입력합니다.

- 암호를 사용하여 인증하는 경우 실행할 명령:

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

  메시지가 표시되면 암호를 입력합니다.
- 키 쌍을 사용하여 인증하는 경우 실행할 명령:

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

> **Note**
>
> 이러한 `sftp` 명령에는 Transfer Family 서버가 위치한 AWS 리전 곳의 ID를 사용하십시오. 예를 들어 서버가 미국 동부(오하이오)에 있는 경우 `us-east-2`를 사용하세요.

`sftp` 프롬프트가 표시되면 pwd 명령을 실행하여 홈 디렉터리를 볼 수 있습니다. 예:

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

사용자는 홈 디렉터리 상위 디렉터리는 일체 볼 수 없습니다. 예:

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

Amazon EFS를 사용하는 경우 Lambda 업데이트

Transfer Family 서버의 스토리지 옵션으로 Amazon EFS를 선택한 경우 스택의 람다 함수를 편집해야 합니다.

Lambda 함수에 Posix 프로필을 추가하려면

1. https://console.aws.amazon.com/lambda/에서 Lambda 콘솔을 엽니다.
2. 이전에 생성한 Lambda 함수를 선택합니다. *Lambda ### ## **## GetUserConfigLambda - - ## ### ### #####. ### ## ### ## #### ## #### ### ######. CloudFormation*
3. 코드 탭에서 index.js를 선택하여 함수에 대한 코드를 표시합니다.
4. `response`에서 `Policy`과(와) `HomeDirectory` 사이에 다음 줄을 추가합니다.

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

   여기서 *uid-value*와 *gid-value*는 각각 사용자 ID와 그룹 ID를 나타내는 0 이상의 정수입니다.

   예를 들어, POSIX 프로필을 추가한 후 응답 필드는 다음과 같을 수 있습니다.

```
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 '/'
   };
```

AS2 구성 설정

이 자습서에서는 AS2 (적용 가능성 설명 2) 구성을 사용하여 설정하는 방법을 안내합니다. AWS Transfer Family여기에 설명된 단계를 완료하면 샘플 거래 파트너(sample trading partner)의 AS2 메시지를 수락할 준비가 된 AS2 지원 서버를 갖게 됩니다. 샘플 거래 파트너에게 AS2 메시지를 보내는 데 사용할 수 있는 커넥터도 있습니다.

> **Note**
>
> 예제 설정의 일부 부분에서는 () 를 사용합니다. AWS Command Line Interface AWS CLI를 아직 설치하지 않은 경우 AWS Command Line Interface 사용 설명서의 최신 버전 설치 또는 업데이트를 참조하십시오. AWS CLI AWS CLI

1. 사용자 자신과 사용자의 거래 파트너를 위한 인증서를 만듭니다. 사용할 수 있는 기존 인증서가 있다면 이 단원을 건너뛰어도 됩니다.

   이 과정은 1단계: AS2용 인증서 생성에 설명되어 있습니다.
2. AS2 프로토콜을 사용하는 AWS Transfer Family 서버를 생성하십시오. 선택적으로 서버에 탄력적 IP 주소를 추가하여 인터넷에 연결되도록 할 수 있습니다.

   이 과정은 2단계: AS2 프로토콜을 사용하는 Transfer Family 서버 생성에 설명되어 있습니다.

   > **Note**
   >
   > 인바운드 전송에만 사용할 Transfer Family 서버를 생성해야 합니다. 아웃바운드 전송만 수행하는 경우에는 Transfer Family 서버가 필요하지 않습니다.
3. 1단계에서 생성한 인증서를 가져옵니다.

이 과정은 3단계: Transfer Family 인증서 리소스로 인증서 가져오기에 설명되어 있습니다.

4. 거래 파트너를 설정하려면 로컬 프로필과 파트너 프로필을 생성하세요.

   이 과정은 4단계: 사용자와 사용자의 거래 파트너를 위한 프로필 생성에 설명되어 있습니다.

5. 거래 파트너와 계약을 체결합니다.

   이 과정은 5단계: 사용자와 사용자의 파트너 간의 계약서 작성에 설명되어 있습니다.

   > **Note**
   >
   > 인바운드 전송에 대한 계약만 작성해야 합니다. 아웃바운드 전송만 수행하는 경우에는 계약이 필요하지 않습니다.

6. 사용자와 사용자의 거래 파트너와 계약을 체결합니다.

   이 과정은 6단계: 사용자와 사용자의 거래 파트너 간의 커넥터를 생성합니다.에 설명되어 있습니다.

   > **Note**
   >
   > 아웃바운드 전송 전용 커넥터를 생성해야 합니다. 인바운드 전송만 수행하는 경우 커넥터가 필요하지 않습니다.

7. AS2 파일 교환을 테스트해 보세요.

   이 과정은 7단계: Transfer Family를 사용하여 AS2를 통한 파일 교환 테스트에 설명되어 있습니다.

이 단계를 완료한 후 다음을 수행할 수 있습니다.

- Transfer Family `start-file-transfer` AWS Command Line Interface (AWS CLI) 명령을 사용하여 원격 AS2 지원 파트너 서버로 파일을 보냅니다.
- Virtual Private Cloud(VPC) 엔드포인트를 통해 포트 5080에서 원격 AS2 지원 파트너 서버로부터 파일을 수신합니다.

1단계: AS2용 인증서 생성

AS2 거래소의 양 당사자는 모두 X.509 인증서가 필요합니다. 원하는 방식으로 이러한 인증서를 생성할 수 있습니다. 이 주제에서는 명령줄에서 OpenSSL을 사용하여 루트 인증서를 만든 다음 하위 인증서에 서명하는 방법을 설명합니다. 양 당사자는 자체 인증서를 생성해야 합니다.

> **Note**
>
> AS2 인증서의 키 길이는 최소 2,048비트, 최대 4,096비트여야 합니다.

파트너와 파일을 전송하려면 다음 사항에 유의하세요.

- 프로필에 인증서를 첨부할 수 있습니다. 인증서에는 퍼블릭 키 또는 프라이빗 키가 포함됩니다.
- 사용자의 거래 파트너가 사용자에게 퍼블릭 키를 보내면, 사용자는 그에게 사용자의 퍼블릭 키를 보냅니다.
- 거래 파트너는 퍼블릭 키로 메시지를 암호화하고 프라이빗 키로 서명합니다. 정반대로, 사용자는 거래 파트너의 퍼블릭 키로 메시지를 암호화하고 그 메시지에 사용자 프라이빗 키로 서명합니다.

> **Note**
>
> GUI로 키를 관리하려는 경우, 사용할 수 있는 옵션 중 하나가 Portecle입니다.

예 인증서를 생성하려면

> **Important**
>
> 파트너에게 사용자 프라이빗 키를 보내지 마세요. 이 예시에서, 사용자는 한 당사자를 위해 자체 서명된 퍼블릭 키와 프라이빗 키 세트를 생성합니다. 테스트 목적으로 두 거래 파트너 역할을 모두 수행하려는 경우 이 지침을 반복하여 거래 파트너당 하나씩 총 두 세트의 키를 생성할 수 있습니다. 이 경우 두 개의 루트 인증 기관(CA)을 생성할 필요가 없습니다.

1. 다음 명령을 실행하여 2,048비트 길이의 모듈러스가 포함된 RSA 프라이빗 키를 생성합니다.

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

2. 다음 명령을 실행하여 사용자의 `root-ca-key.pem`파일로 자체 서명된 인증서를 생성합니다.

```
/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` 아규먼트는 다음 두 값으로 구성됩니다.

	명칭	설명
C	국가 코드	사용자 조직이 위치한 국가를 나타내는 2자리 코드입니다.
ST	주, 지역 또는 지방	해당 조직이 위치한 주 또는 지방의 이름. (이 경우 지역은 사용자의 AWS 리전을 의미하지 않습니다.)
L	시 이름	해당 조직이 위치한 시의 이름.
O	조직 이름	LLC, Corp 등과 같은 접미사를 포함한 사용자 조직의 전체 법적 이름.
OU	조직 단위 이름	조직 내에서 이 인증서를 다루는 부서.
CN	일반적인 이름 또는 완전한 자격을 갖춘 도메인 이름 (FQDN)	이 예에서는 루트 인증서를 생성하므로 값은 ROOTCA입니다. 이 예에서는 인증서의 용도를 설명하는데 CN를 사용합니다.

3. 로컬 프로필을 위한 서명 키와 암호화 키를 생성하세요.

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

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

> **Note**
>
> OpenAS2와 같은 일부 AS2 지원 서버에서는 서명과 암호화 모두에 동일한 인증서를 사용해야 합니다. 이 경우 두 가지 용도로 동일한 프라이빗 키와 인증서를 가져올 수 있습니다. 이렇게 하려면 이전 명령 두 개 대신 다음 명령을 실행합니다.
>
> ```
> /usr/bin/openssl genrsa -out signing-and-encryption-key.pem 2048
> ```

4. 다음 명령을 실행하여 루크 키를 위한 인증서 서명 요청(CSR)을 생성하여 서명합니다.

```
/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. 다음으로, `signing-cert.conf` 파일과 `encryption-cert.conf` 파일을 만들어야 합니다.

   - 텍스트 편집기를 사용하여 다음 내용을 포함하는 `signing-cert.conf` 파일을 생성합니다.

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

   - 텍스트 편집기를 사용하여 다음 내용을 포함하는 `encryption-cert.conf` 파일을 생성합니다.

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

6. 마지막으로 다음 명령을 실행하여 서명된 인증서를 생성합니다.

```
/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
```

2단계: AS2 프로토콜을 사용하는 Transfer Family 서버 생성

이 절차에서는 Transfer Family AWS CLI를 사용하여 AS2 지원 서버를 생성하는 방법을 설명합니다.

> **Note**
>
> 많은 예 단계에서 파일에서 파라미터를 로드하는 명령을 사용합니다. 파일을 사용하여 파라미터를 로드하는 방법에 대한 자세한 내용은 파일에서 파라미터를 로드하는 방법을 참조하세요.

이 방법 대신 콘솔을 사용하려면 Transfer Family 콘솔을 사용하여 AS2 서버 생성 섹션을 참조하세요.

SFTP 또는 FTPS AWS Transfer Family 서버를 생성하는 방법과 마찬가지로 명령의 매개 변수를 사용하여 AS2 지원 서버를 생성합니다. `--protocols AS2` `create-server` AWS CLI 현재 Transfer Family는 VPC 엔드포인트 타입과 AS2 프로토콜을 사용하는 Amazon S3 스토리지만 지원합니다.

`create-server` 명령을 사용하여 Transfer Family용 AS2 지원 서버를 생성하면 VPC 엔드포인트가 자동으로 생성됩니다. 이 엔드포인트는 AS2 메시지를 받아들일 수 있도록 TCP 포트 5080을 노출합니다.

VPC 엔드포인트를 인터넷에 공개하려는 경우 탄력적 IP 주소를 VPC 엔드포인트와 연결할 수 있습니다.

이러한 지침을 사용하려면 다음이 필요합니다.

- 사용자 VPC의 ID (예: vpc-abcdef01).
- 사용자 VPC 서브넷의 ID (예: 서브넷-abcdef01, 서브넷-서브넷-abcdef01, 서브넷-021345ab).
- 거래 파트너로부터 TCP 포트 5080을 통해 들어오는 트래픽을 허용하는 보안 그룹의 ID 하나 이상 (예: sg-1234567890abcdef0 및 sg-abcdef01234567890)
- (옵션) VPC 엔드포인트와 연결하려는 탄력적 IP 주소.
- 거래 파트너가 VPN을 통해 VPC에 연결되지 않은 경우 인터넷 게이트웨이가 필요합니다. 자세한 내용은 Amazon VPC 사용 설명서의 인터넷 게이트웨이를 사용한 인터넷 연결을 참조하세요.

AS2 지원 서버를 만들려면

1. 다음 명령을 실행합니다. 각 *user input placeholder*를 사용자의 정보로 바꿉니다.

```
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. (옵션) VPC 엔드포인트를 퍼블릭으로 만들 수 있습니다. update-server작업을 통해서만 Transfer Family 서버에 탄력적 IP 주소를 연결할 수 있습니다. 다음 명령은 서버를 중지하고 탄력적 IP 주소로 업데이트한 다음 다시 시작합니다.

```
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
```

   이 start-server 명령은 사용자 서버의 퍼블릭 IP 주소가 포함된 DNS 레코드를 자동으로 생성합니다. 거래 파트너에게 서버 액세스 권한을 부여하려면 거래 파트너에게 다음 정보를 제공해야 합니다. 이 경우에는 *your-region*는 귀하의 AWS 리전를 참조합니다.

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

   거래 파트너에게 제공하는 전체 URL은 다음과 같습니다.

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

3. AS2 지원 서버의 액세스 가능 여부를 테스트하려면 다음 명령을 사용하세요. VPC 엔드포인트의 프라이빗 DNS 주소 또는 퍼블릭 엔드포인트(탄력적 IP 주소를 엔드포인트와 연결한 경우)를 통해 서버에 액세스할 수 있는지 확인하세요.

   서버가 올바르게 구성되어 있으면 연결이 성공합니다. 하지만 유효한 AS2 메시지를 보내고 있지 않기 때문에 HTTP 상태 코드 400(잘못된 요청) 응답을 받게 됩니다.

- 퍼블릭 엔드포인트의 경우 (이전 단계에서 탄력적 IP 주소를 연결한 경우) 서버 ID와 지역을 대체하여 다음 명령을 실행합니다.

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

- VPC 내에서 연결하는 경우 다음 명령을 실행하여 VPC 엔드포인트의 프라이빗 DNS 이름을 조회합니다.

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

이 `describe-server` 명령은 `VpcEndpointId` 파라미터에 VPC 엔드포인트 ID를 반환합니다. 이 값을 사용하여 다음 명령을 실행합니다.

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

이 `describe-vpc-endpoints` 명령은 `DNSEntries` 배열을 반환하며, 여기에는 여러 `DnsName` 파라미터가 포함됩니다. 다음 명령에서 지역 DNS 이름(가용 영역을 포함하지 않는 이름)을 사용합니다.

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

예를 들어, 다음 명령은 이전 명령의 자리 표시자에 대한 샘플 값을 보여줍니다.

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

4. (옵션) 로깅 역할을 구성합니다. Transfer Family는 구조화된 JSON 형식으로 보내고 받은 메시지의 상태를 Amazon CloudWatch 로그에 기록합니다. Transfer Family에 계정의 CloudWatch 로그에 대한 액세스 권한을 제공하려면 서버에서 로깅 역할을 구성해야 합니다.

   신뢰할 `transfer.amazonaws.com` 수 있는 AWS Identity and Access Management (IAM) 역할을 만들고 `AWSTransferLoggingAccess` 관리형 정책을 연결합니다. 자세한 내용은 IAM 역할 및 정책 생성를 참조하세요. 방금 생성한 IAM 역할의 Amazon 리소스 이름 (ARN) 을 메모하고 다음 `update-server` 명령을 실행하여 서버에 연결합니다.

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

> **Note**
>
> 로깅 역할은 옵션이지만 메시지 상태를 확인하고 구성 문제를 해결할 수 있도록 로깅 역할을 설정하는 것이 좋습니다.

3단계: Transfer Family 인증서 리소스로 인증서 가져오기

이 절차에서는 AWS CLI를 사용하여 인증서를 가져오는 방법을 설명합니다. 이 방법 대신 Transfer Family 콘솔을 사용하려면 the section called “AS2 인증서 가져오기” 섹션을 참조하세요.

1단계에서 만든 서명 및 암호화 인증서를 가져오려면 다음 `import-certificate` 명령을 실행합니다. 암호화와 서명에 동일한 인증서를 사용하는 경우 동일한 인증서를 두 번 가져오세요 (한 번은 `SIGNING` 사용량과 함께, 다른 한 번은 `ENCRYPTION` 사용량과 함께).

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

이 명령은 사용자 서명 `CertificateId`을 반환합니다. 다음 섹션에서는 이 인증서 ID를 *my-signing-cert-id*라고 합니다.

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

이 명령은 암호화 `CertificateId`를 반환합니다. 다음 섹션에서는 이 인증서 ID를 *my-encrypt-cert-id*라고 합니다.

그런 다음, 다음 명령을 실행하여 파트너의 암호화 및 서명 인증서를 가져옵니다.

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

이 명령은 파트너의 암호화 CertificateId를 반환합니다. 다음 섹션에서는 이 인증서 ID를 *partner-encrypt-cert-id*라고 합니다.

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

이 명령은 파트너의 서명 CertificateId를 반환합니다. 다음 섹션에서는 이 인증서 ID를 *partner-signing-cert-id*라고 합니다.

4단계: 사용자와 사용자의 거래 파트너를 위한 프로필 생성

이 절차에서는 를 사용하여 AS2 프로필을 생성하는 방법을 설명합니다. AWS CLI이 방법 대신 Transfer Family 콘솔을 사용하려면 the section called “AS2 프로필 생성” 섹션을 참조하세요.

다음 명령을 실행하여 로컬 AS2 프로필을 생성합니다. 이 명령은 퍼블릭 키와 프라이빗 키가 포함된 인증서를 참조합니다.

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

이 명령은 프로필 ID를 반환합니다. 다음 섹션에서는 이 ID를 *my-profile-id*라고 합니다.

이제 다음 명령을 실행하여 파트너 프로필을 생성합니다. 이 명령어는 파트너의 퍼블릭 키 인증서만 사용합니다. 이 명령을 사용하려면 *user input placeholders*를 사용자 자신(예: 파트너의 AS2 이름 및 인증서 ID)의 정보로 바꾸세요.

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

이 명령은 파트너의 프로필 ID를 반환합니다. 다음 섹션에서는 이 ID를 *partner-profile-id*라고 합니다.

> **Note**
>
> 이전 명령에서 *MYCORP*를 조직 이름으로 바꾸고 *PARTNER-COMPANY*를 사용자의 거래 파트너의 조직 이름으로 바꾸세요.

5단계: 사용자와 사용자의 파트너 간의 계약서 작성

이 절차에서는 AWS CLI를 사용하여 AS2 계약서를 생성하는 방법에 대해 설명합니다. 이 방법 대신 Transfer Family 콘솔을 사용하려면 the section called “AS2 계약 생성” 섹션을 참조하세요.

계약에는 두 프로필 (로컬 및 파트너), 인증서, 그리고 두 당사자 간의 인바운드 AS2 전송을 허용하는 서버 구성이 함께 제공됩니다. 다음 명령을 실행하여 항목을 나열할 수 있습니다.

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

이 단계에는 버킷에 대한 읽기/쓰기 액세스 권한이 있는 Amazon S3 버킷 및 IAM 역할이 필요합니다. 이 역할을 생성하는 지침은 Transfer Family SFTP, FTP 및 FTPS 프로토콜의 지침과 동일하며 IAM 역할 및 정책 생성에서 확인할 수 있습니다.

계약을 생성하려면 다음 항목이 필요합니다.

- Amazon S3 버킷 이름(지정된 경우 객체 접두사 포함)
- 버킷에 액세스할 수 있는 IAM 역할의 ARN입니다.
- 사용자의 Transfer Family 서버 ID
- 사용자의 프로필 ID 및 사용자 파트너의 프로필 ID

다음 명령을 실행하여 계약을 생성합니다.

```
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
```

성공한 경우 계약의 ID가 반환됩니다. 그리고 다음 명령을 사용하여 계약의 세부 정보를 볼 수 있습니다.

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

6단계: 사용자와 사용자의 거래 파트너 간의 커넥터를 생성합니다.

이 절차에서는 AWS CLI를 사용하여 AS2 커넥터를 생성하는 방법에 대해 설명합니다. 이 방법 대신 Transfer Family 콘솔을 사용하려면 the section called “AS2 커넥터 구성” 섹션을 참조하세요.

`StartFileTransfer` API 작업을 사용하면 커넥터를 사용하여 Amazon S3에 저장된 파일을 거래 파트너의 AS2 엔드포인트로 전송할 수 있습니다. 다음 명령을 실행하여 이전에 생성한 프로필을 찾을 수 있습니다.

```
aws transfer list-profiles
```

커넥터를 생성할 때 파트너의 AS2 서버 URL을 제공해야 합니다. 다음 텍스트를 `testAS2Config.json`라는 파일에 복사합니다.

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

> **Note**
>
> 의 경우`EncryptionAlgorithm`, `DES_EDE3_CBC` 알고리즘을 필요로 하는 레거시 클라이언트를 지원해야 하는 경우가 아니면 알고리즘을 지정하지 마십시오. 이는 취약한 암호화 알고리즘이기 때문입니다.

이제 다음 명령을 실행하여 커넥터를 생성합니다.

```
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
```

7단계: Transfer Family를 사용하여 AS2를 통한 파일 교환 테스트

거래 파트너로부터 파일 받기

퍼블릭 탄력적 IP 주소를 VPC 엔드포인트와 연결한 경우 Transfer Family는 퍼블릭 IP 주소를 포함하는 DNS 이름을 자동으로 생성합니다. 하위 도메인은 `s-1234567890abcdef0` 해당 형식의 AWS Transfer Family 서버 ID입니다. 거래 파트너에게 다음과 같은 형식으로 서버 URL을 입력합니다.

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

퍼블릭 탄력적 IP 주소를 VPC 엔드포인트와 연결하지 않은 경우, 포트 5080에서 거래 파트너가 보내는 HTTP POST를 통해 AS2 메시지를 받아들일 수 있는 VPC 엔드포인트의 호스트 이름을 찾아보세요. VPC 엔드포인트 세부 정보를 검색하려면 다음 명령을 사용합니다.

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

예를 들어 위의 명령이 `vpce-1234abcd5678efghi`의 VPC 엔드포인트 ID를 반환한다고 가정해 보겠습니다. 그런 다음, 다음 명령을 사용하여 DNS 이름을 검색합니다.

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

이 명령은 다음 명령을 실행하는 데 필요한 VPC 엔드포인트의 모든 세부 정보를 반환합니다.

DNS 이름은 `DnsEntries` 배열에 나열됩니다. 거래 파트너가 VPC 내에 있어야 VPC 엔드포인트에 액세스할 수 있습니다(예: AWS PrivateLink 또는 VPN을 통해). 거래 파트너에게 다음과 같은 형식으로 서버 VPC 엔드포인트 URL을 제공합니다.

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

예를 들어, 다음 URL은 이전 명령의 자리 표시자에 대한 샘플 값을 보여줍니다.

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

이 예에서 성공적인 전송은 지정한 `base-directory` 파라미터에 5단계: 사용자와 사용자의 파트너 간의 계약서 작성의 지정된 위치에 저장됩니다. 이름이 `myfile1.txt` 및 `myfile2.txt` 인 파일이 성공적으로 수신되면 파일은 */path-defined-in-the-agreement/processed/original_filename.messageId.original_extension*로 저장됩니

다. 여기서 파일은 /DOC-EXAMPLE-DESTINATION-BUCKET/AS2-inbox/processed/myfile1.*messageId*.txt 및 로 /DOC-EXAMPLE-DESTINATION-BUCKET/AS2-inbox/processed/myfile2.*messageId*.txt저장됩니다.

Transfer Family 서버를 생성할 때 로깅 역할을 구성한 경우 CloudWatch 로그에서 AS2 메시지 상태를 확인할 수도 있습니다.

거래 파트너에게 파일을 보내세요.

Transfer Family를 사용하면 다음 `start-file-transfer` AWS Command Line Interface ()AWS CLI 명령에 표시된 대로 커넥터 ID 및 파일 경로를 참조하여 AS2 메시지를 보낼 수 있습니다.

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

커넥터의 세부 정보를 가져오려면 다음 명령을 실행합니다.

```
aws transfer list-connectors
```

이 `list-connectors` 명령은 커넥터의 커넥터 ID, URL 및 Amazon 리소스 이름(ARN)을 반환합니다.

특정 커넥터의 속성을 반환하려면 사용하려는 ID로 다음 명령을 실행합니다.

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

이 `describe-connector` 명령은 커넥터의 URL, 역할, 프로필, mDNS (메시지 처리 알림), 태그 및 모니터링 메트릭을 포함하여 커넥터의 모든 속성을 반환합니다.

JSON 및 MDN 파일을 보면 파트너가 파일을 성공적으로 수신했는지 확인할 수 있습니다. 이러한 파일 이름은 파일 이름 및 위치에 설명된 규칙에 따라 지정됩니다. 커넥터를 생성할 때 로깅 역할을 구성한 경우 CloudWatch 로그에서 AS2 메시지 상태를 확인할 수도 있습니다.

SFTP, FTPS 또는 FTP 서버 엔드포인트 구성

이 항목에서는 하나 이상의 SFTP, FTPS 및 FTP 프로토콜을 사용하는 AWS Transfer Family 서버 엔드포인트를 만들고 사용하는 방법에 대한 세부 정보를 제공합니다.

주제

자격 증명 공급자 옵션

AWS Transfer Family 사용자를 인증하고 관리하는 여러 가지 방법을 제공합니다. 다음 표에서는 Transfer Family와 함께 사용할 수 있는 ID 공급자를 비교합니다.

작업	AWS Transfer Family 서비스 관리형	AWS Managed Microsoft AD	Amazon API Gateway	AWS Lambda
지원되는 프로토콜	SFTP	SFTP, FTPS, FTP	SFTP, FTPS, FTP	SFTP, FTPS, FTP
키 기반 인증	예	아니요	예	예
암호 인증	아니요	예	예	예
AWS Identity and Access Management (IAM) 및 POSIX	예	예	예	예
논리적 홈 디렉터리	예	예	예	예

작업	AWS Transfer Family 서비스 관리형	AWS Managed Microsoft AD	Amazon API Gateway	AWS Lambda
파라미터화된 액세스(사용자 이름 기반)	예	예	예	예
임시 액세스 구조	예	아니요	예	예
AWS WAF	아니요	아니요	예	아니요

참고:

- IAM은 Amazon S3 지원 스토리지에 대한 액세스를 통제하는 데 사용되고 POSIX는 Amazon EFS에 사용됩니다.
- 임시는 런타임에 사용자 프로필을 전송할 수 있는 기능을 의미합니다. 예컨대, 사용자 이름을 변수로 전달하여 사용자를 홈 디렉터리에 연결할 수 있습니다.
- 에 대한 AWS WAF자세한 내용은 을 참조하십시오. 웹 애플리케이션 방화벽 추가
- Microsoft Azure AD와 통합된 Lambda 함수를 Transfer Family ID 공급자로 사용하는 방법을 설명하는 블로그 게시물이 있습니다. 자세한 내용은 Azure Active Directory를 AWS Transfer Family 사용한 인증 및 을 참조하십시오. AWS Lambda
- 사용자 지정 ID 공급자를 사용하는 Transfer Family 서버를 신속하게 배포하는 데 도움이 되는 몇 가지 AWS CloudFormation 템플릿을 제공합니다. 자세한 내용은 Lambda 함수 템플릿를 참조하세요.

다음 절차에서 SFTP 지원 서버, FTPS 지원 서버, FTP 지원 서버 또는 AS2 지원 서버를 생성할 수 있습니다.

다음 단계

- SFTP 지원 서버 생성
- FTPS 지원 서버 생성
- FTP 지원 서버 생성
- AS2 구성

AWS Transfer Family 엔드포인트 유형 매트릭스

Transfer Family 서버를 생성하는 경우 사용할 엔드포인트 타입을 선택합니다. 다음 표에서는 각 엔드포인트 타입의 특성에 대해 설명합니다.

엔드포인트 타입 매트릭스

기능	퍼블릭	VPC - 인터넷	VPC - 내부형	VPC_엔드포인트 (폐지됨)
지원되는 프로토콜	SFTP	SFTP, FTPS, AS2	SFTP, FTP, FTPS, AS2	SFTP
액세스	인터넷을 통해 이 엔드포인트 타입에는 VPC에 특별한 구성이 필요하지 않습니다.	인터넷을 통해, 그리고 VPC와 VPC로 연결된 환경 (예: 온프레미스 데이터 센터 또는 VPN) 내에서 가능합니다. AWS Direct Connect	VPC 및 VPC로 연결된 환경 (예: 온프레미스 데이터 센터 오버 또는 VPN) 내에서 AWS Direct Connect	VPC 및 VPC로 연결된 환경 (예: 온프레미스 데이터 센터 오버 또는 VPN) 내에서 AWS Direct Connect
고정 IP 주소	고정 IP 주소는 연결할 수 없습니다. AWS 변경될 수 있는 IP 주소를 제공합니다.	탄력적 IP 주소를 엔드포인트에 연결할 수 있습니다. 이는 AWS에서 소유한 IP 주소일 수도 있고 자체 IP 주소일 수도 있습니다(자체 IP 주소 사용). 엔드포인트에 연결된 탄력적 IP 주소는 변경되지 않습니다.	엔드포인트에 연결된 사설 IP 주소는 변경되지 않습니다.	엔드포인트에 연결된 사설 IP 주소는 변경되지 않습니다.

기능	퍼블릭	VPC - 인터넷	VPC - 내부형	VPC_엔드포인트 (폐지됨)
		서버에 연결된 사설 IP 주소도 변경되지 않습니다.		

기능	퍼블릭	VPC - 인터넷	VPC - 내부형	VPC_엔드포인트 (폐지됨)
소스 IP 허용 목록	이 엔드포인트 타입은 소스 IP 주소별 허용 목록을 지원하지 않습니다. 엔드포인트는 공개적으로 액세스할 수 있으며 포트 22를 통해 트래픽을 수신합니다. **Note** VPC 호스팅 엔드포인트의 경우 SFTP Transfer Family 서버는 포트 22(기본값), 포트 2222 또는 포트 22000을 통해 작동할 수 있습니다.	소스 IP 주소별 액세스를 허용하려면 서버 엔드포인트에 연결된 보안 그룹과 엔드포인트가 있는 서브넷에 연결된 네트워크 ACL을 사용할 수 있습니다.	소스 IP 주소별 액세스를 허용하려면 서버 엔드포인트에 연결된 보안 그룹과 엔드포인트가 있는 서브넷에 연결된 네트워크 ACL(액세스 통제 목록)을 사용할 수 있습니다.	소스 IP 주소별 액세스를 허용하려면 서버 엔드포인트에 연결된 보안 그룹과 엔드포인트가 있는 서브넷에 연결된 네트워크 ACL을 사용할 수 있습니다.

기능	퍼블릭	VPC - 인터넷	VPC - 내부형	VPC_엔드포인트 (폐지됨)
클라이언트 방화벽 허용 목록	서버의 DNS 이름을 허용해야 합니다. IP 주소는 변경될 수 있으므로 클라이언트 방화벽 허용 목록에 IP 주소를 사용하지 마세요.	서버의 DNS 이름 또는 서버에 연결된 탄력적 IP 주소를 허용할 수 있습니다.	엔드포인트의 프라이빗 IP 주소 또는 DNS 이름을 허용할 수 있습니다.	엔드포인트의 프라이빗 IP 주소 또는 DNS 이름을 허용할 수 있습니다.

> **Note**
>
> VPC_ENDPOINT 엔드포인트 타입은 이제 더 이상 사용되지 않으며 새 서버를 만드는 데 사용할 수 없습니다. EndpointType=VPC_ENDPOINT를 사용하는 대신 이전 표에 설명된 대로 내부 또는 인터넷 연결로 사용할 수 있는 새 VPC 엔드포인트 타입(EndpointType=VPC)을 사용하세요. 자세한 내용은 VPC_ENDPOINT 사용 중단를 참조하세요.

AWS Transfer Family 서버의 보안 상태를 강화하려면 다음 옵션을 고려해 보세요.

- 내부 액세스가 가능한 VPC 엔드포인트를 사용하여 VPC 또는 VPC로 연결된 환경 (예: 온프레미스 데이터 센터 over) 또는 VPN과 같은 VPC 연결 환경 내의 클라이언트만 서버에 액세스할 수 있도록 합니다. AWS Direct Connect
- 클라이언트가 인터넷을 통해 엔드포인트에 액세스하도록 허용하고 서버를 보호하려면 인터넷 연결 액세스가 가능한 VPC 엔드포인트를 사용하세요. 그런 다음 사용자 클라이언트를 호스팅하는 특정 IP 주소로부터의 트래픽만 허용하도록 VPC의 보안 그룹을 수정하세요.
- 암호 기반 인증이 필요하고 서버에서 맞춤 ID 공급자를 사용하는 경우 암호 정책을 통해 사용자가 취약한 암호를 만들지 못하도록 하고 로그인 시도 실패 횟수를 제한하는 것이 좋습니다.
- AWS Transfer Family 관리형 서비스이므로 셸 액세스를 제공하지 않습니다. Transfer Family 서버에서는 기본 SFTP 서버에 직접 액세스하여 OS 기본 명령을 실행할 수 없습니다.
- 내부 액세스가 가능한 VPC 엔드포인트 앞에서 Network Load Balancer를 사용하세요. 로드 밸런서의 리스너 포트를 포트 22에서 다른 포트로 변경합니다. 이렇게 하면 포트 스캐너와 봇이 서버를 탐

색할 위험을 줄일 수 있지만 제거하지는 못합니다. 포트 22가 스캔에 가장 많이 사용되기 때문입니다. 자세한 내용은 블로그 게시물을 참조하십시오. 이제 네트워크 로드 밸런서가 보안 그룹을 지원합니다.

> **Note**
>
> Network Load Balancer를 사용하는 경우 AWS Transfer Family CloudWatch 로그에는 실제 클라이언트 IP 주소가 아닌 NLB의 IP 주소가 표시됩니다.

SFTP, FTPS 또는 FTP 서버 엔드포인트 구성

서비스를 사용하여 파일 전송 서버를 생성할 수 있습니다. AWS Transfer Family 다음과 같은 파일 전송 프로토콜을 사용할 수 있습니다.

- Secure Shell(SSH) 파일 전송 프로토콜(SFTP) - SSH를 통한 파일 전송. 자세한 내용은 the section called “SFTP 지원 서버 생성”를 참조하세요.

  > **Note**
  >
  > SFTP Transfer Family 서버를 만드는 AWS CDK 예제를 제공합니다. 이 예제에서는 를 TypeScript 사용하며 GitHub 여기에서 확인할 수 있습니다.

- FTPS(파일 전송 프로토콜 보안) - TLS 암호화를 사용한 파일 전송. 자세한 내용은 the section called “FTPS 지원 서버 생성”를 참조하세요.
- FTP(파일 전송 프로토콜) - 암호화되지 않은 파일 전송. 자세한 내용은 the section called “FTP 지원 서버 생성”를 참조하세요.
- 적용 가능성 선언문 2 (AS2) — 정형 데이터 전송을 위한 파일 전송. business-to-business 자세한 내용은 the section called “AS2를 구성하십시오”단원을 참조하세요. AS2의 경우 데모용 AWS CloudFormation 스택을 빠르게 생성할 수 있습니다. 이 절차는 템플릿을 사용하여 데모 Transfer Family AS2 스택을 생성하세요.에 설명되어 있습니다.

여러 프로토콜로 서버를 만들 수 있습니다.

> **Note**
>
> 같은 서버 엔드포인트에 여러 프로토콜을 사용하도록 설정한 상태에서 여러 프로토콜에서 동일한 사용자 이름을 사용하여 액세스를 제공하려는 경우 ID 공급자에 각 프로토콜별 보안 인증이 설정되어 있으면 그렇게 할 수 있습니다. FTP의 경우 SFTP 및 FTPS와 별도의 자격 증명을 유지하는 것이 좋습니다. 이는 SFTP 및 FTPS와 달리 FTP는 자격 증명을 일반 텍스트로 전송하기 때문입니다. FTP 자격 증명을 SFTP 또는 FTPS에서 분리하면 FTP 자격 증명이 공유되거나 노출되더라도 SFTP 또는 FTPS를 사용하는 워크로드의 보안을 유지할 수 있습니다.

서버를 생성할 때 해당 서버에 할당된 사용자의 파일 작업 요청을 수행할 특정 AWS 리전 서버를 선택합니다. 서버에 하나 이상의 프로토콜을 할당하는 동시에 다음 ID 공급자 타입 중 하나를 할당합니다:

- SSH 키를 사용한 서비스 관리. 자세한 내용은 서비스 관리형과 작업를 참조하세요.
- AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD). 이 방법을 사용하면 Microsoft Active Directory 그룹을 통합하여 Transfer Family 서버에 대한 액세스를 제공할 수 있습니다. 자세한 내용은 AWS 디렉터리 서비스 ID 제공자 사용를 참조하세요.
- 맞춤 메서드. 맞춤 자격 증명 공급자는 AWS Lambda 또는 Amazon API Gateway를 이용하며, 사용자는 디렉터리 서비스를 통합해 사용자를 인증하고 승인할 수 있습니다. 이 서비스는 사용자의 서버만 식별하는 식별자를 자동으로 할당합니다. 자세한 내용은 사용자 지정 자격 증명 공급자와 작업를 참조하세요. Transfer Family는 사용자 지정 ID 공급자를 사용하는 서버를 빠르게 배포하는 데 사용할 수 있는 AWS CloudFormation 템플릿을 제공합니다.
  - 인증을 위한 Lambda 함수인증을 위해 Lambda 함수를 사용하는 CloudFormation 템플릿에 대해 설명합니다.
  - API Gateway 메서드를 사용하여 인증인증에 Amazon API Gateway 방법을 사용하는 CloudFormation 템플릿에 대해 설명합니다.

또한 기본 서버 엔드포인트를 사용하여 엔드포인트 타입(공개적으로 액세스 가능 또는 VPC 호스팅)과 호스트 이름을 서버에 할당하거나, Amazon Route 53 서비스를 사용하거나 선택한 도메인 이름 시스템(DNS) 서비스를 사용하여 맞춤 호스트 이름을 할당합니다. 서버 호스트 이름은 생성된 AWS 리전 위치에서 고유해야 합니다.

또한 Amazon CloudWatch 로깅 역할을 할당하여 이벤트를 CloudWatch 로그로 푸시하고, 서버에서 사용할 수 있는 암호화 알고리즘이 포함된 보안 정책을 선택하고, 키-값 쌍인 태그의 형태로 서버에 메타데이터를 추가할 수 있습니다.

> **Important**
>
> 인스턴스화한 서버와 데이터 전송에 대한 비용이 발생합니다. 요금 및 Transfer Family 사용 비용 추정에 대한 자세한 내용은 AWS Transfer Family 요금을 참조하십시오. AWS Pricing Calculator

SFTP 지원 서버 생성

Secure Shell(SSH) File Transfer 프로토콜(SFTP)은 인터넷을 통한 안전한 데이터 전송에 사용되는 네트워크 프로토콜입니다. 이 프로토콜은 SSH의 전체 보안 및 인증 기능을 지원합니다. 금융 서비스, 의료, 소매 및 광고와 같은 다양한 산업의 비즈니스 파트너들 사이의 민감한 정보를 비롯한 데이터를 교환하는 데 널리 사용됩니다.

> **Note**
>
> Transfer Family용 SFTP 서버는 포트 22를 통해 작동합니다. VPC 호스팅 엔드포인트의 경우 SFTP Transfer Family 서버는 포트 2222 또는 포트 22000을 통해 작동할 수도 있습니다. 자세한 내용은 Virtual Private Cloud(VPC)에 서버 생성단원을 참조하세요.

참고 항목

- SFTP Transfer Family 서버를 만드는 AWS CDK 예제를 제공합니다. 이 예제에서는 를 TypeScript 사용하며 GitHub 여기에서 확인할 수 있습니다.
- VPC 내에 Transfer Family 서버를 배포하는 방법에 대한 자세한 내용은 IP 허용 목록을 사용하여 서버 보안을 참조하십시오. AWS Transfer Family

SFTP 지원 서버를 생성하려면

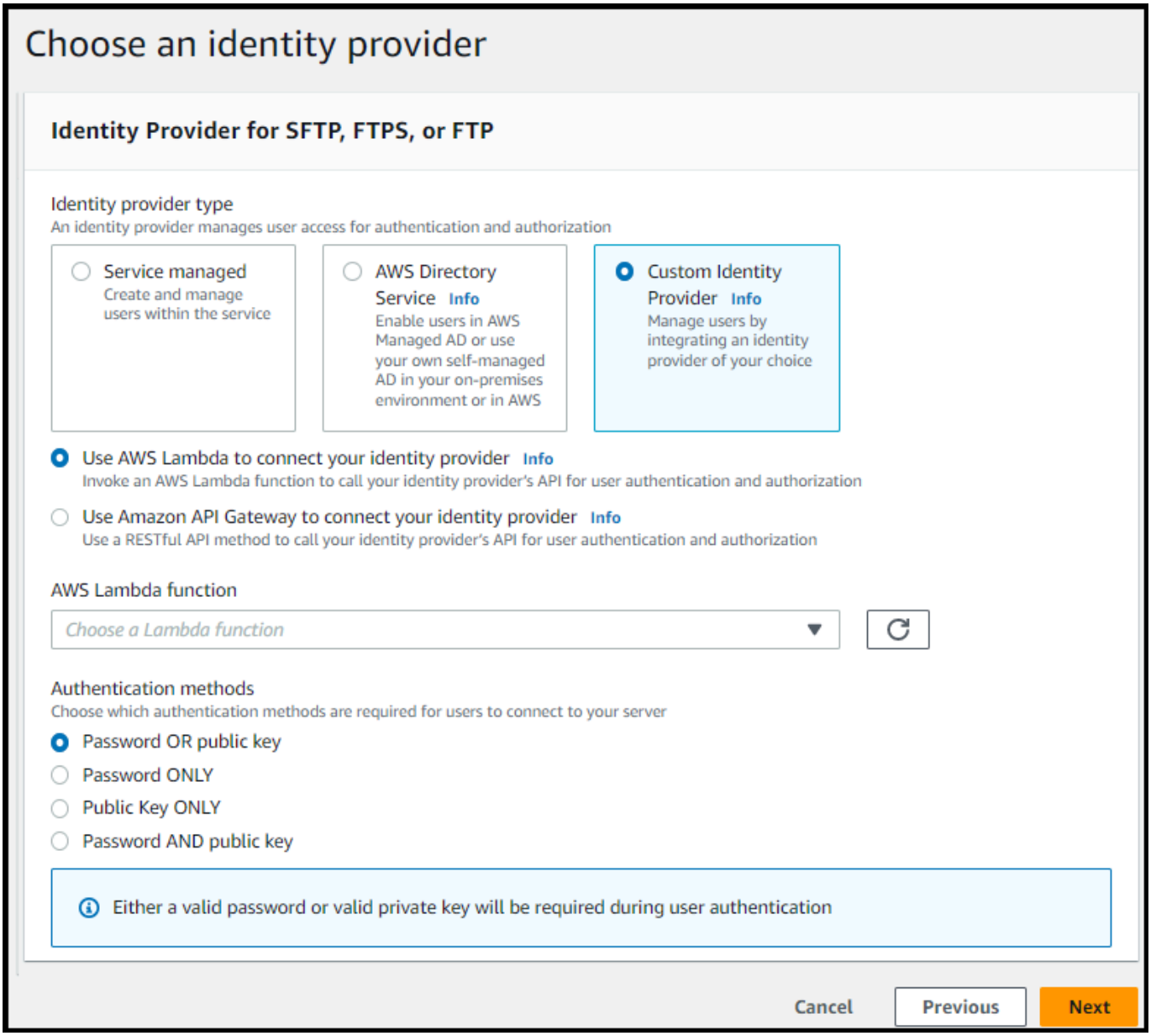
1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 열고 탐색 창에서 서버를 선택한 다음 서버 생성을 선택합니다.
2. 프로토콜 선택에서 SFTP를 선택하고 다음을 선택합니다.
3. 자격 증명 공급자 선택에서 사용자 액세스를 관리하는 데 사용할 자격 증명 공급자를 선택합니다. 다음과 같은 옵션이 있습니다:
   - 서비스 관리형 - 사용자 ID와 키를 저장합니다. AWS Transfer Family

- AWS Directory Service for Microsoft Active Directory— 엔드포인트에 액세스할 수 있는 AWS Directory Service 디렉터리를 제공합니다. 그러면 Active Directory에 저장된 자격 증명을 사용하여 사용자를 인증할 수 있습니다. AWS Managed Microsoft AD ID 공급자와의 작업에 대한 자세한 내용은 을 참조하십시오AWS 디렉터리 서비스 ID 제공자 사용.

  > **Note**
  >
  > - 교차 계정 및 공유 디렉터리는 지원되지 않습니다. AWS Managed Microsoft AD
  > - Directory Service를 ID 제공자로 사용하여 서버를 설정하려면 몇 가지 AWS Directory Service 권한을 추가해야 합니다. 자세한 내용은 사용을 시작하기 전 AWS Directory Service for Microsoft Active Directory단원을 참조하세요.

- 맞춤 ID 공급자 – 다음 옵션 중 하나를 선택합니다.
  - ID 공급자 AWS Lambda 연결에 사용 - Lambda 함수가 지원하는 기존 ID 공급자를 사용할 수 있습니다. 귀하는 Lambda 함수의 명칭을 제공합니다. 자세한 내용은 ID AWS Lambda 제공자를 통합하는 데 사용를 참조하세요.
  - Amazon API Gateway를 사용하는 자격 증명 공급자 연결 - Lambda 함수가 지원하는 API Gateway 방법을 생성하여 자격 증명 공급자로 사용할 수 있습니다. 사용자는 Amazon API Gateway URL과 호출 역할을 제공합니다. 자세한 내용은 Amazon API Gateway를 ID 제공자 통합에 사용를 참조하세요.

  어느 옵션이든 귀하가 인증 방법도 지정할 수 있습니다.
  - 암호 또는 키 — 사용자는 암호 또는 키로 인증할 수 있습니다. 이것이 기본값입니다.
  - 암호만 해당 — 연결하려면 사용자가 암호를 입력해야 합니다.
  - 키만 해당 - 연결하려면 사용자가 개인 키를 제공해야 합니다.
  - 암호 및 키 — 연결하려면 사용자가 개인 키와 암호를 모두 제공해야 합니다. 서버가 먼저 키를 확인한 다음 키가 유효하면 암호를 입력하라는 메시지가 표시됩니다. 제공된 프라이빗 키가 저장된 퍼블릭 키와 일치하지 않는 경우 인증이 실패합니다.



4. 다음을 선택합니다.
5. 엔드포인트 선택에서 다음을 수행합니다.

   a. 엔드포인트 타입으로는 공적으로 액세스 가능 엔드포인트 타입을 선택합니다. VPC 호스팅 엔드포인트에 대한 내용은 Virtual Private Cloud(VPC)에 서버 생성 섹션을 참조하세요.

   b. (옵션) 맞춤 호스트 이름의 경우 없음을 선택합니다.

      에서 서버 호스트 이름을 제공합니다. AWS Transfer Family서버 호스트 이름은 *serverId*.server.transfer.*regionId*.amazonaws.com 형식을 취합니다.

맞춤 호스트 이름의 경우 서버 엔드포인트의 맞춤 별칭을 지정합니다. 사용자 지정 호스트 이름 사용에 대한 자세한 내용은 사용자 지정 호스트 이름으로 작업를 참조하세요.

c. (옵션) FIPS 지원의 경우 FIPS 지원 엔드포인트 확인란을 선택하여 엔드포인트가 미연방 정보 처리 표준(FIPS)를 준수하는지 확인합니다.

> **Note**
>
> FIPS 지원 엔드포인트는 북미 AWS 리전에서만 사용할 수 있습니다. 사용할 수 있는 리전은 AWS 일반 참조의 AWS Transfer Family 엔드포인트 및 할당량을 참조하세요. FIPS에 대한 자세한 설명은 미연방 정보 처리 표준(FIPS) 140-2를 참조하세요.

d. 다음을 선택합니다.

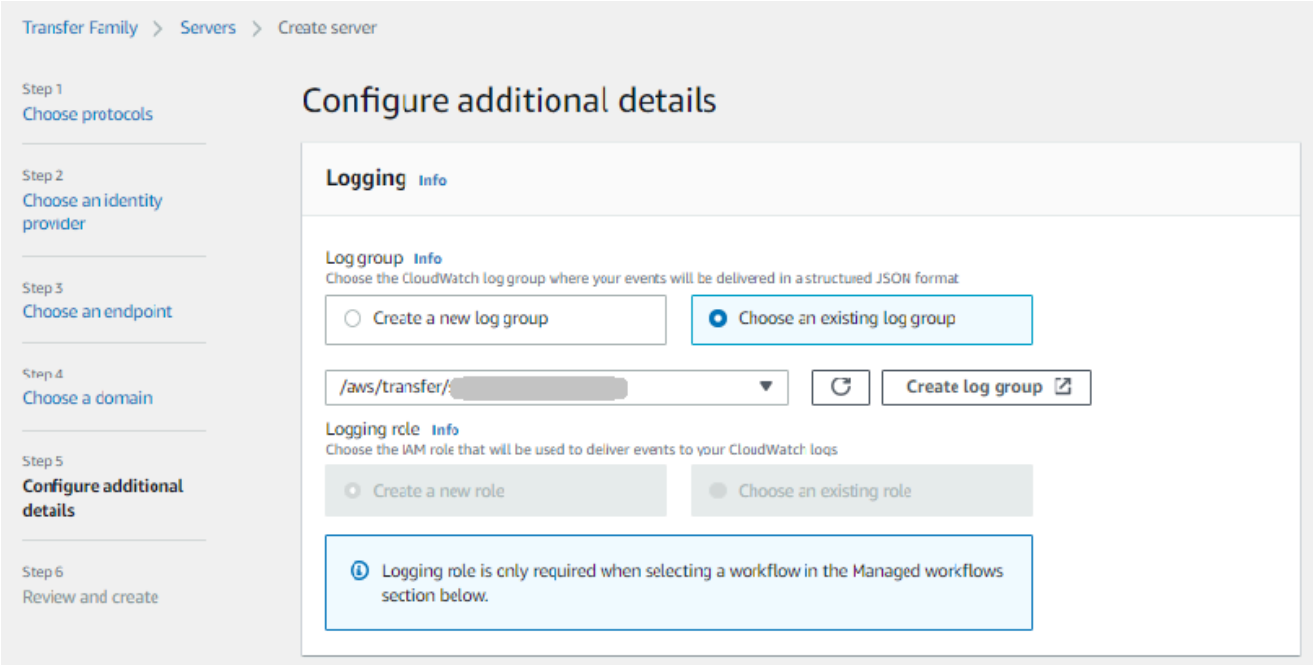
6. 도메인 선택 페이지에서 선택한 프로토콜을 통해 데이터를 저장하고 액세스하는 데 사용할 AWS 스토리지 서비스를 선택합니다.

   - 선택한 프로토콜을 통해 파일을 객체로 저장하고 액세스하려면 Amazon S3를 선택하세요.
   - 선택한 프로토콜을 통해 Amazon EFS 파일 시스템에 파일을 저장하고 액세스하려면 Amazon EFS를 선택합니다.

   다음을 선택합니다.

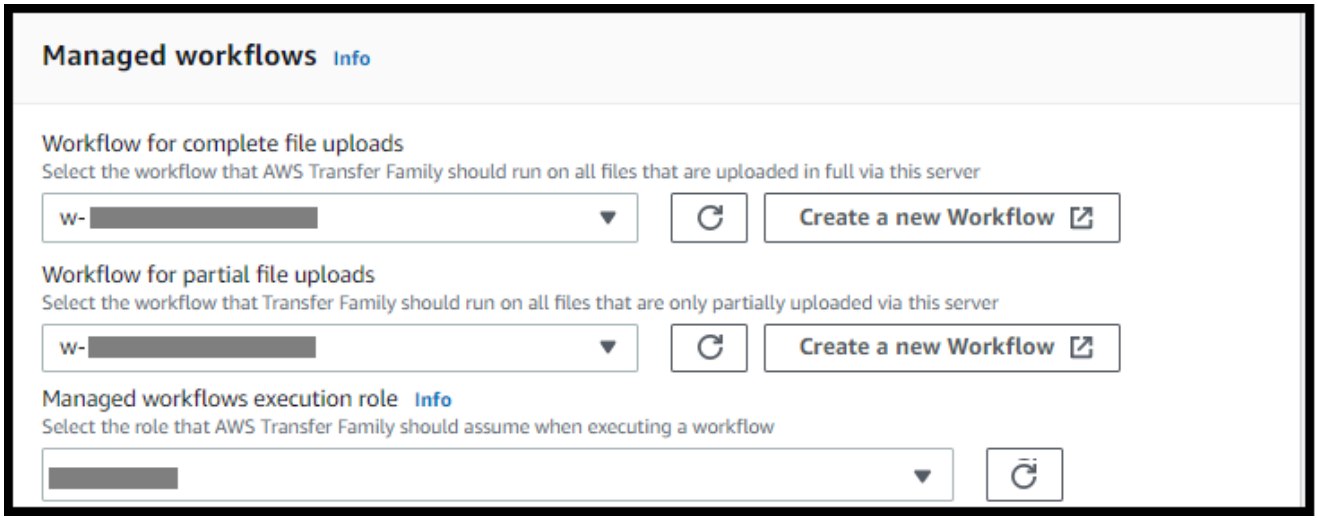
7. 추가 상세 정보 구성에서 다음을 수행합니다.

   a. 로깅의 경우 기존 로그 그룹을 지정하거나 새 로그 그룹을 생성합니다(기본 옵션). 기존 로그 그룹을 선택하는 경우 해당 그룹과 연결된 그룹을 선택해야 AWS 계정합니다.



로그 그룹 생성을 선택하면 CloudWatch 콘솔 (https://console.aws.amazon.com/cloudwatch/)에서 로그 그룹 생성 페이지가 열립니다. 자세한 내용은 로그에 CloudWatch 로그 그룹 만들기를 참조하십시오.

b. (옵션) 관리형 워크플로의 경우 워크플로를 실행할 때 Transfer Family가 맡아야 하는 워크플로 ID(및 해당 역할)를 선택합니다. 업로드 완료 시 실행할 워크플로 하나와 부분 업로드 시 실행할 워크플로 하나를 선택할 수 있습니다. 관리형 워크플로를 사용하여 파일을 처리하는 방법에 대한 자세한 내용은 AWS Transfer Family 관리형 워크플로를 참조하세요.



c. 암호화 알고리즘 옵션의 경우 서버에서 사용하도록 설정된 암호화 알고리즘이 포함된 보안 정책을 선택합니다. 최신 보안 정책이 기본값입니다. 자세한 내용은 을 참조하십시오서버 보안 정책 AWS Transfer Family.

d. (옵션) 서버 호스트 키의 경우 클라이언트가 SFTP를 통해 서버에 연결할 때 서버를 식별하는 데 사용할 RSA, ED25519 또는 ECDSA 프라이빗 키를 입력하세요. (옵션) 설명을 추가하여 여러 서버 호스트 키를 구분합니다.

   서버를 생성한 후 호스트 키를 추가할 수 있습니다. 키를 교체하거나 RSA 키와 ECDSA 키 같은 다양한 타입의 키를 사용하려는 경우 호스트 키가 여러 개 있으면 유용합니다.

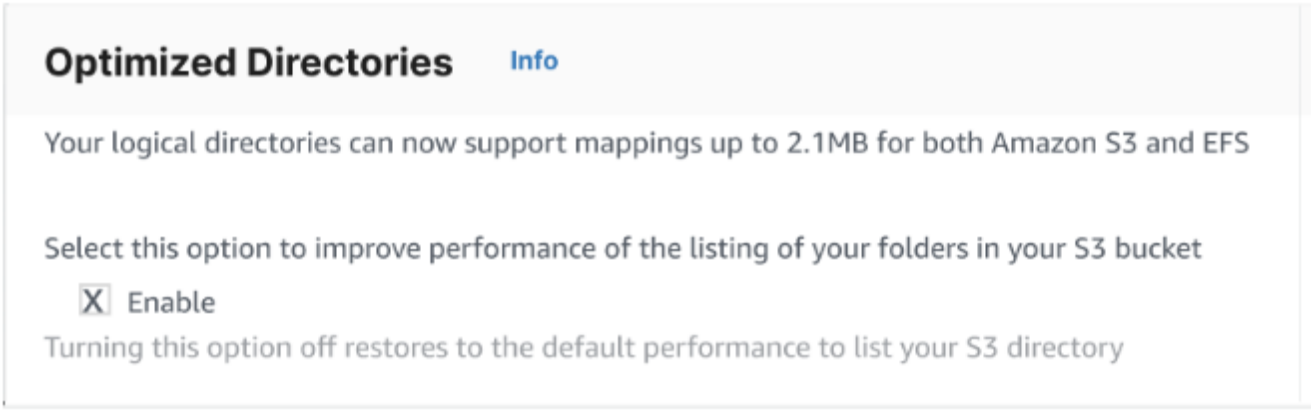
   > **Note**
   >
   > 서버 호스트 키 섹션은 기존 SFTP 지원 서버에서 사용자를 마이그레이션하는 데만 사용됩니다.

e. (옵션) 태그의 키 및 값에서 하나 이상의 태그를 키-값 쌍을 입력한 다음 태그 추가를 선택합니다.

f. 다음을 선택합니다.

g. Amazon S3 디렉터리의 성능을 최적화할 수 있습니다. 예를 들어 홈 디렉터리에 10,000개의 하위 디렉터리가 있다고 가정해 보겠습니다. 즉, Amazon S3 버킷에는 10,000개의 폴더가 있습니다. 이 시나리오에서 `ls` (list) 명령을 실행하면 목록 작업에 6~8분이 소요됩니다. 하지만 디렉터리를 최적화하는 경우 이 작업은 몇 초밖에 걸리지 않습니다.

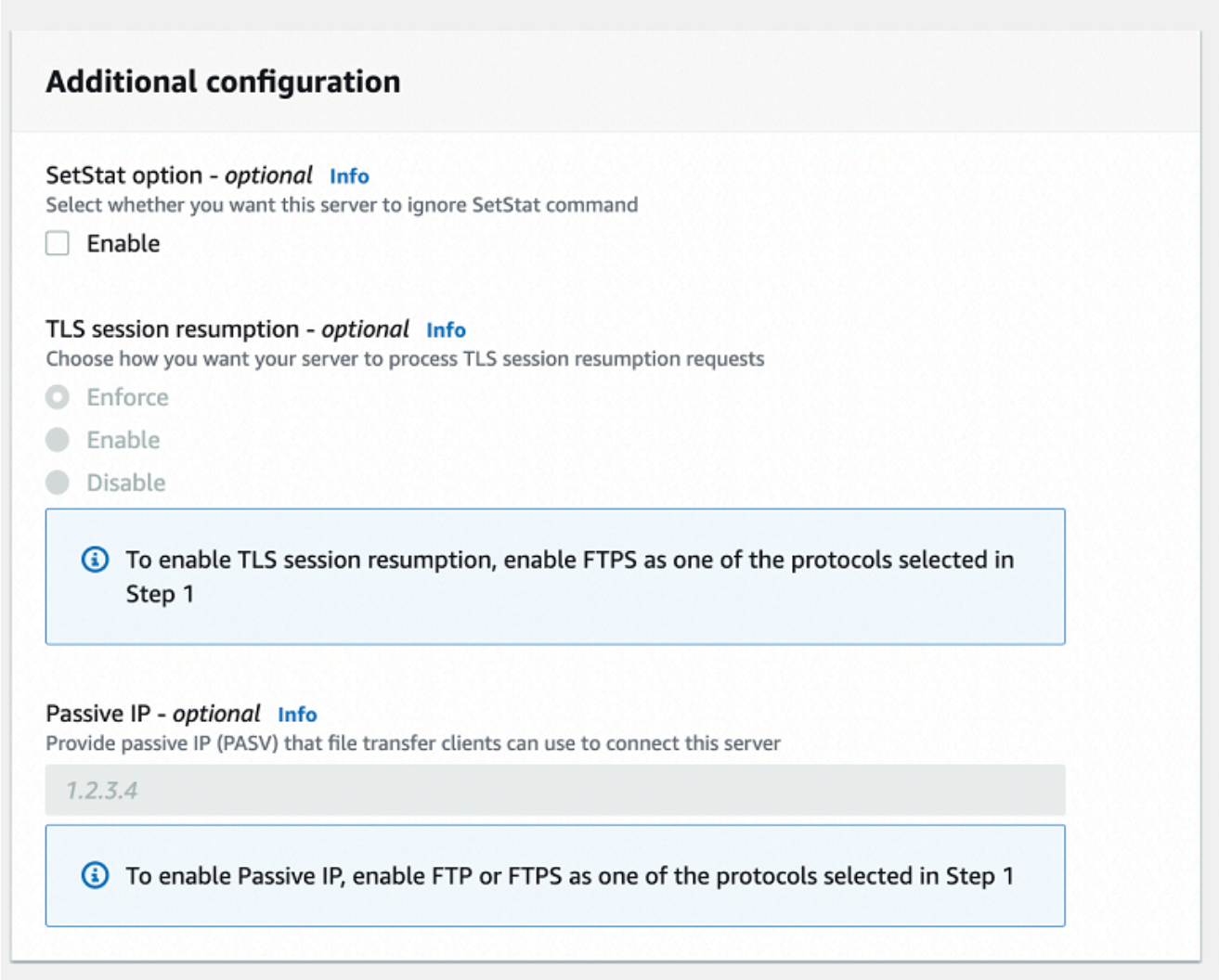
   콘솔을 사용하여 서버를 생성하면 최적화된 디렉터리가 기본적으로 활성화됩니다. API를 사용하여 서버를 생성하는 경우 이 동작은 기본적으로 활성화되지 않습니다.

h. (선택 사항) 조직 정책 또는 이용 약관과 같은 사용자 지정 메시지를 최종 사용자에게 표시하도록 AWS Transfer Family 서버를 구성합니다. 디스플레이 배너의 경우 사전 인증 디스플레이 배너 텍스트 상자에 사용자가 인증하기 전에 표시할 텍스트 메시지를 입력합니다.

i. (옵션) 다음 추가 옵션을 구성할 수 있습니다.

- SetStat 옵션: Amazon S3 SETSTAT 버킷에 업로드하는 파일에서 클라이언트가 사용을 시도할 때 생성되는 오류를 무시하려면 이 옵션을 활성화합니다. 자세한 내용은 의 `SetStatOption` ProtocolDetails설명서를 참조하십시오.
- TLS 세션 재개: 이 옵션은 FTPS를 이 서버의 프로토콜 중 하나로 활성화한 경우에만 사용할 수 있습니다.
- 패시브 IP: 이 옵션은 FTPS 또는 FTP를 이 서버의 프로토콜 중 하나로 활성화한 경우에만 사용할 수 있습니다.



8. 검토 및 생성에서 옵션을 검토합니다.
   - 모든 항목을 편집하려면 단계 옆에 있는 편집을 선택합니다.

> **Note**
>
> 편집하기로 선택한 단계 이후의 각 단계를 검토해야 합니다.

- 변경 사항이 없는 경우 서버 생성을 선택하여 서버를 생성하세요. 서버 페이지로 이동하고, 새 서버가 나열되는 다음 화면이 표시됩니다.

새 서버 상태가 온라인으로 변경되기까지 몇 분 정도 걸릴 수 있습니다. 이때 서버에서 파일 작업을 수행할 수 있지만 먼저 사용자를 만들어야 합니다. 사용자 생성에 대한 자세한 내용은 을 참조하십시오서버 엔드포인트의 사용자 관리.

FTPS 지원 서버 생성

SSL을 통한 파일 전송 프로토콜(FTPS)은 FTP의 확장입니다. 전송 계층 보안(TLS) 및 Sockets Layer(SSL) 암호화 프로토콜을 사용하여 트래픽을 암호화합니다. FTPS를 사용하면 통제 및 데이터 채널 연결을 동시에 또는 개별적으로 암호화할 수 있습니다.

FTPS 지원 서버를 만들려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 열고 탐색 창에서 서버를 선택한 다음 서버 생성을 선택합니다.
2. 프로토콜 선택에서 FTPS를 선택합니다.

   서버 인증서의 경우 클라이언트가 FTPS를 통해 서버에 연결할 때 서버를 식별하는 데 사용되는 AWS Certificate Manager (ACM)에 저장된 인증서를 선택한 후 다음을 선택합니다.

   새 퍼블릭 인증서를 요청하려면 AWS Certificate Manager 사용 설명서의 퍼블릭 인증서 요청을 참조하세요.

   기존 인증서를 ACM으로 가져오려면 AWS Certificate Manager 사용 설명서의 ACM으로 인증서 가져오기를 참조하세요.

   프라이빗 IP 주소를 통해 FTPS를 사용하도록 프라이빗 인증서를 요청하려면 AWS Certificate Manager 사용 설명서의 프라이빗 인증서 요청을 참조하세요.

   다음 암호화 알고리즘 및 키 크기를 사용하는 인증서가 지원됩니다:

   - 2048비트 RSA(RSA_2048)

- 4096비트 RSA(RSA_4096)
- 타원 프라임 곡선 256비트(EC_prime256v1)
- 타원 프라임 곡선 384 비트(EC_secp384r1)
- 타원 프라임 곡선 521비트(EC_secp521r1)

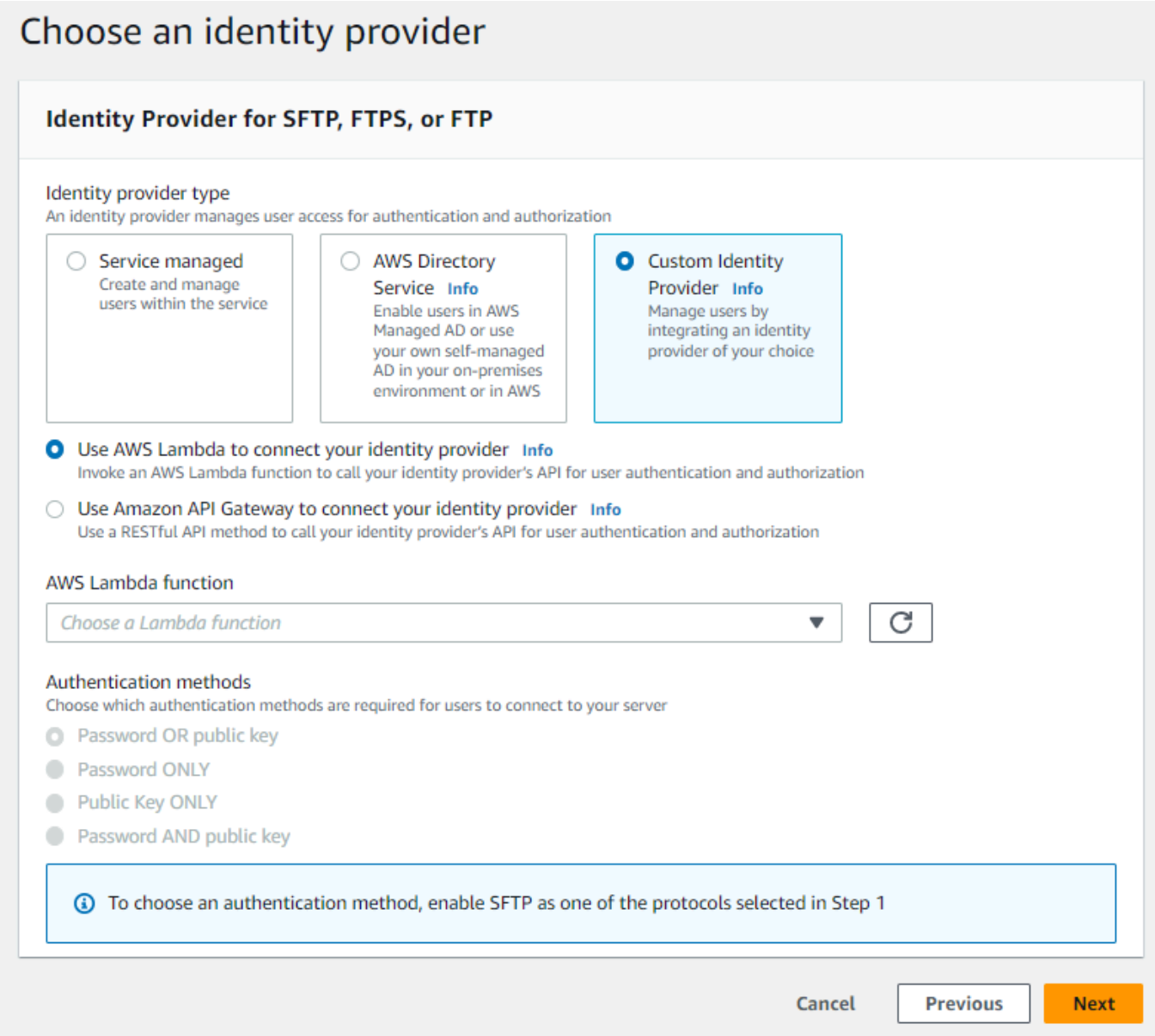
> **Note**
>
> 인증서는 FQDN 또는 IP 주소가 지정된 유효한 SSL/TLS X.509 버전 3 인증서여야 하며 발급자에 대한 정보가 포함되어 있어야 합니다.

3. 자격 증명 공급자 선택에서 사용자 액세스를 관리하는 데 사용할 자격 증명 공급자를 선택합니다. 다음과 같은 옵션이 있습니다:

- AWS Directory Service for Microsoft Active Directory— 엔드포인트에 액세스할 수 있는 디렉터리를 제공합니다. AWS Directory Service 그러면 Active Directory에 저장된 자격 증명을 사용하여 사용자를 인증할 수 있습니다. AWS Managed Microsoft AD ID 공급자와의 작업에 대한 자세한 내용은 을 참조하십시오AWS 디렉터리 서비스 ID 제공자 사용.

  > **Note**
  >
  > - 교차 계정 및 공유 디렉터리는 지원되지 않습니다. AWS Managed Microsoft AD
  > - Directory Service를 ID 제공자로 사용하여 서버를 설정하려면 몇 가지 AWS Directory Service 권한을 추가해야 합니다. 자세한 내용은 사용을 시작하기 전 AWS Directory Service for Microsoft Active Directory단원을 참조하세요.

- 맞춤 ID 공급자 – 다음 옵션 중 하나를 선택합니다.
  - ID 공급자 AWS Lambda 연결에 사용 - Lambda 함수가 지원하는 기존 ID 공급자를 사용할 수 있습니다. 귀하는 Lambda 함수의 명칭을 제공합니다. 자세한 내용은 ID AWS Lambda 제공자를 통합하는 데 사용를 참조하세요.
  - Amazon API Gateway를 사용하는 자격 증명 공급자 연결 - Lambda 함수가 지원하는 API Gateway 방법을 생성하여 자격 증명 공급자로 사용할 수 있습니다. 사용자는 Amazon API Gateway URL과 호출 역할을 제공합니다. 자세한 내용은 Amazon API Gateway를 ID 제공자 통합에 사용를 참조하세요.



4. 다음을 선택합니다.
5. 엔드포인트 선택에서 다음을 수행합니다.

> **Note**
>
> Transfer Family용 FTPS 서버는 포트 21(제어 채널) 및 포트 범위 8192-8200(데이터 채널)을 통해 작동합니다.

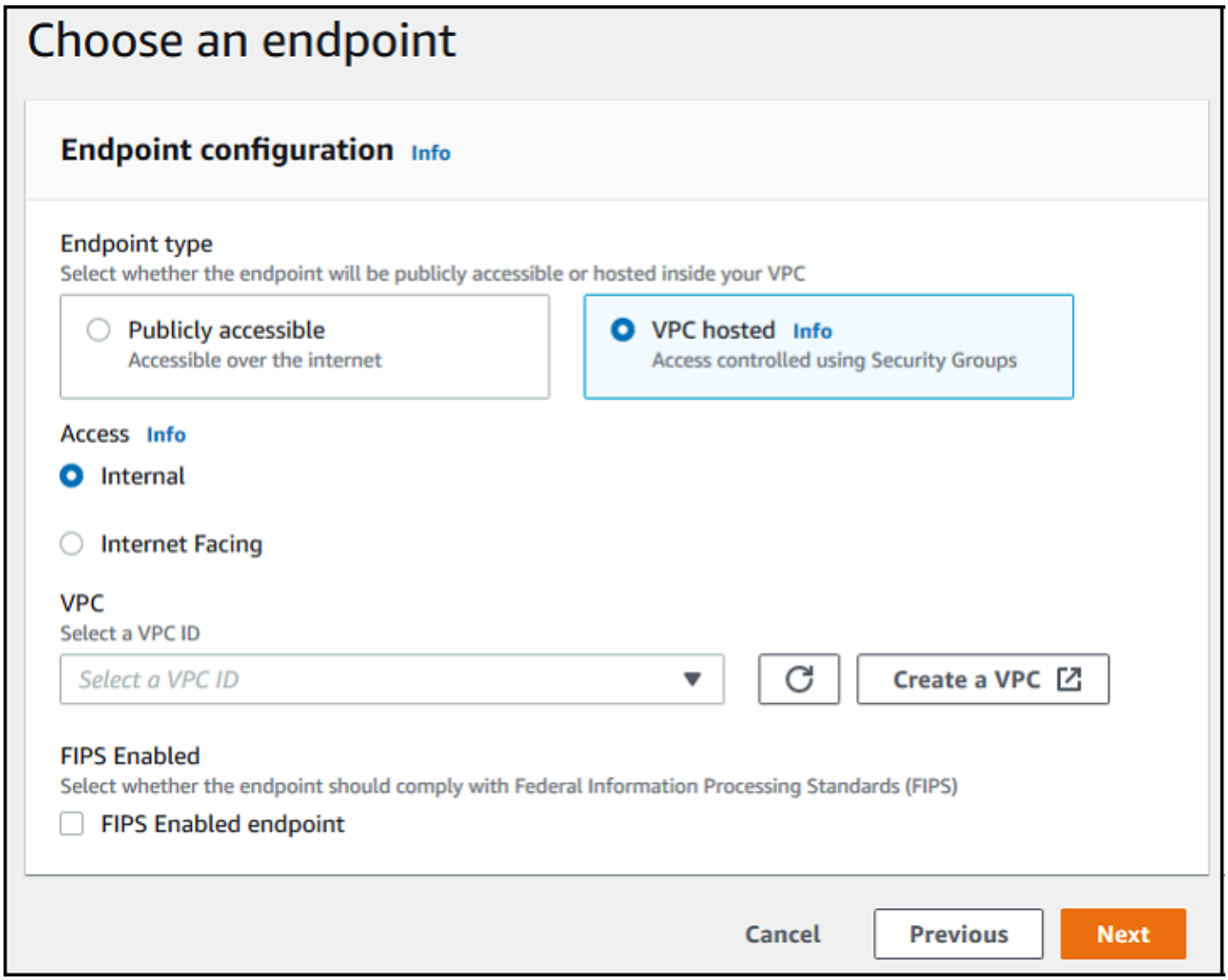
a. 엔드포인트 타입에서 서버의 엔드포인트를 호스팅할 VPC 호스팅 엔드포인트 타입을 선택합니다. VPC 호스팅 엔드포인트를 설정하는 자세한 설명은 Virtual Private Cloud(VPC)에 서버 생성 섹션을 참조하세요.

> **Note**
>
> 공적 액세스 가능 엔드포인트는 지원되지 않습니다.

b. (옵션) FIPS 지원의 경우 FIPS 지원 엔드포인트 확인란을 선택하여 엔드포인트가 미연방 정보 처리 표준(FIPS)를 준수하는지 확인합니다.

> **Note**
>
> FIPS 지원 엔드포인트는 북미 AWS 리전에서만 사용할 수 있습니다. 사용할 수 있는 리전은 AWS 일반 참조의 AWS Transfer Family 엔드포인트 및 할당량을 참조하세요. FIPS에 대한 자세한 설명은 미연방 정보 처리 표준(FIPS) 140-2를 참조하세요.

c. 다음을 선택합니다.



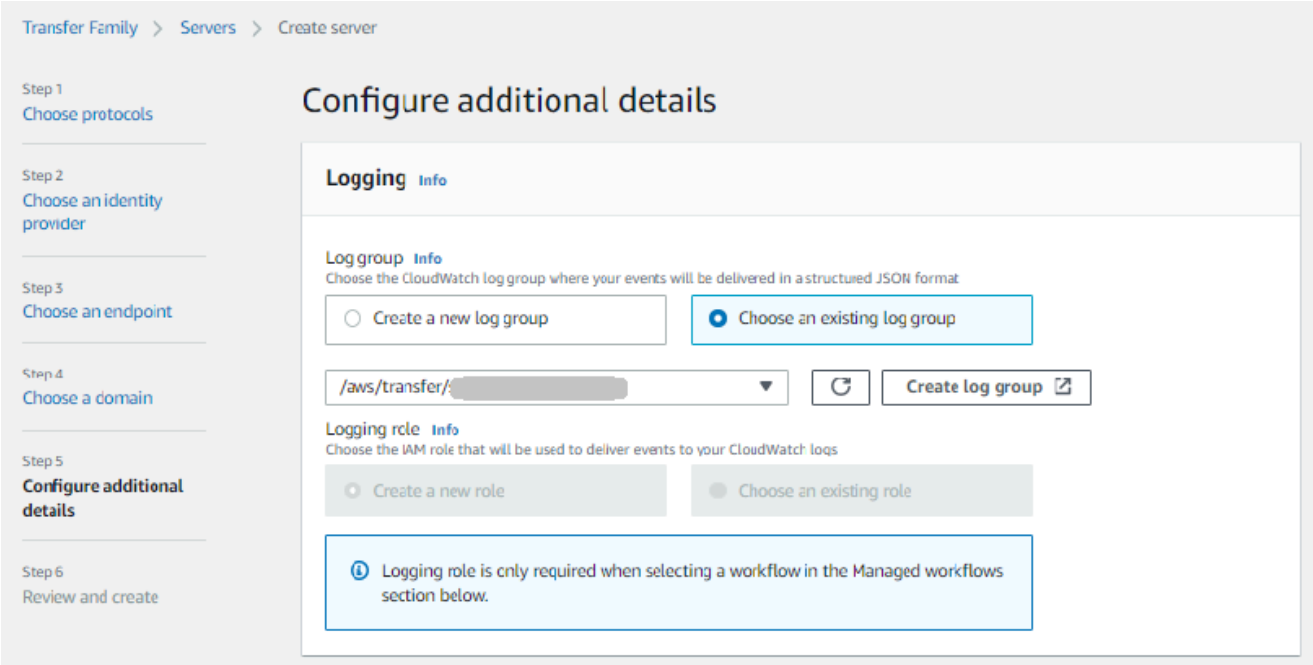
6. 도메인 선택 페이지에서 선택한 프로토콜을 통해 데이터를 저장하고 액세스하는 데 사용할 AWS 스토리지 서비스를 선택합니다.

   - 선택한 프로토콜을 통해 파일을 객체로 저장하고 액세스하려면 Amazon S3를 선택하세요.
   - 선택한 프로토콜을 통해 Amazon EFS 파일 시스템에 파일을 저장하고 액세스하려면 Amazon EFS를 선택합니다.

   다음을 선택합니다.

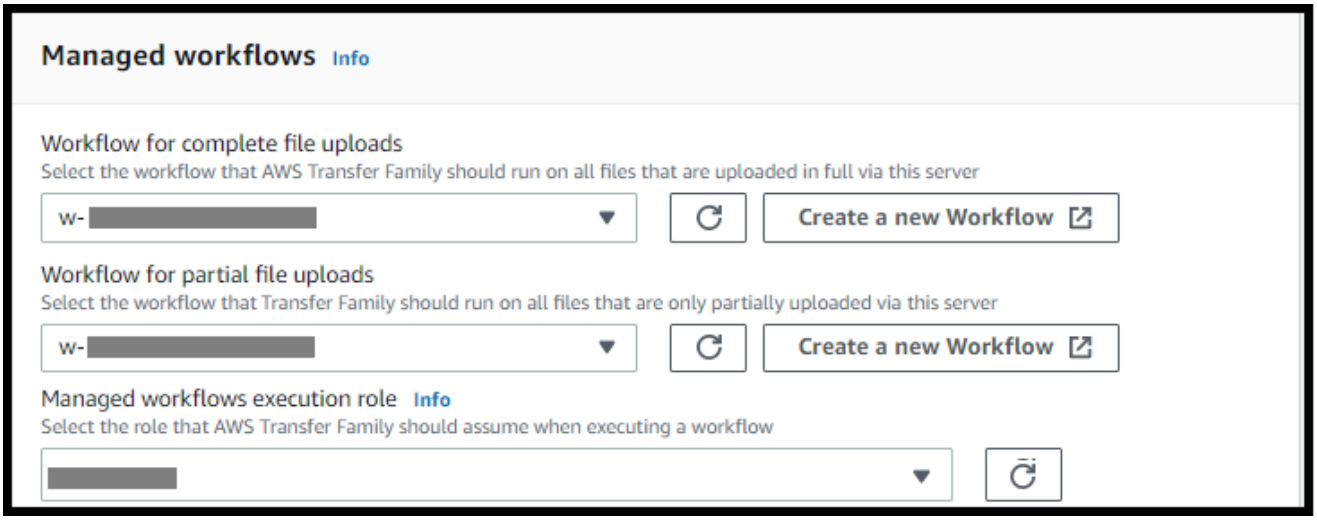
7. 추가 상세 정보 구성에서 다음을 수행합니다.

   a. 로깅의 경우 기존 로그 그룹을 지정하거나 새 로그 그룹을 생성합니다(기본 옵션).



로그 그룹 생성을 선택하면 CloudWatch 콘솔 (https://console.aws.amazon.com/cloudwatch/)에서 로그 그룹 생성 페이지가 열립니다. 자세한 내용은 로그에 CloudWatch 로그 그룹 만들기를 참조하십시오.

b. (옵션) 관리형 워크플로의 경우 워크플로를 실행할 때 Transfer Family가 맡아야 하는 워크플로 ID(및 해당 역할)를 선택합니다. 업로드 완료 시 실행할 워크플로 하나와 부분 업로드 시 실행할 워크플로 하나를 선택할 수 있습니다. 관리형 워크플로를 사용하여 파일을 처리하는 방법에 대한 자세한 내용은 AWS Transfer Family 관리형 워크플로를 참조하세요.



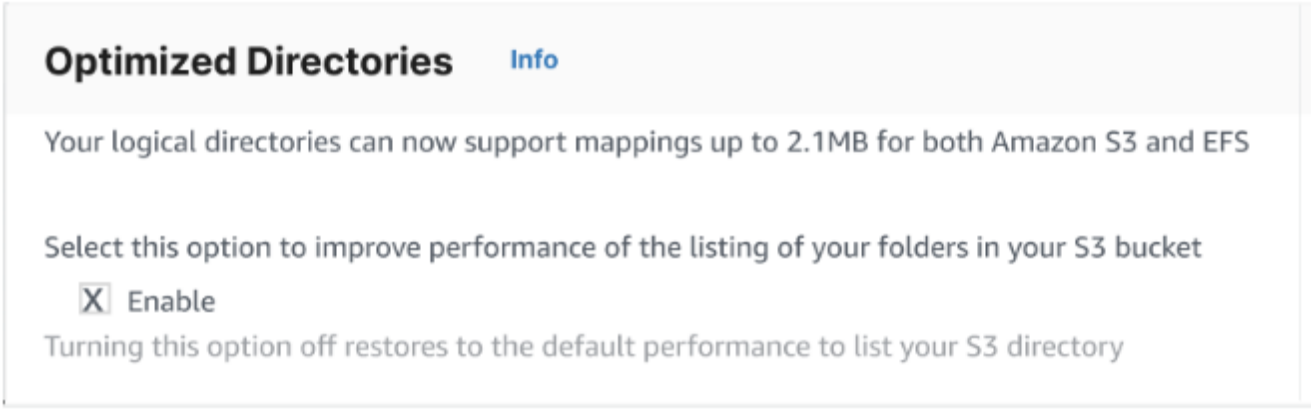
c. 암호화 알고리즘 옵션의 경우 서버에서 사용하도록 설정된 암호화 알고리즘이 포함된 보안 정책을 선택합니다. 최신 보안 정책이 기본값입니다. 자세한 내용은 을 참조하십시오서버 보안 정책 AWS Transfer Family.

d. 서버 호스트 키의 경우 비워 두세요.

e. (옵션) 태그의 키 및 값에서 하나 이상의 태그를 키-값 쌍을 입력한 다음 태그 추가를 선택합니다.

f. Amazon S3 디렉터리의 성능을 최적화할 수 있습니다. 예를 들어 홈 디렉터리에 10,000개의 하위 디렉터리가 있다고 가정해 보겠습니다. 즉, Amazon S3 버킷에는 10,000개의 폴더가 있습니다. 이 시나리오에서 `ls` (list) 명령을 실행하면 목록 작업에 6~8분이 소요됩니다. 하지만 디렉터리를 최적화하는 경우 이 작업은 몇 초밖에 걸리지 않습니다.

콘솔을 사용하여 서버를 생성하면 최적화된 디렉터리가 기본적으로 활성화됩니다. API를 사용하여 서버를 생성하는 경우 이 동작은 기본적으로 활성화되지 않습니다.



g. 다음을 선택합니다.

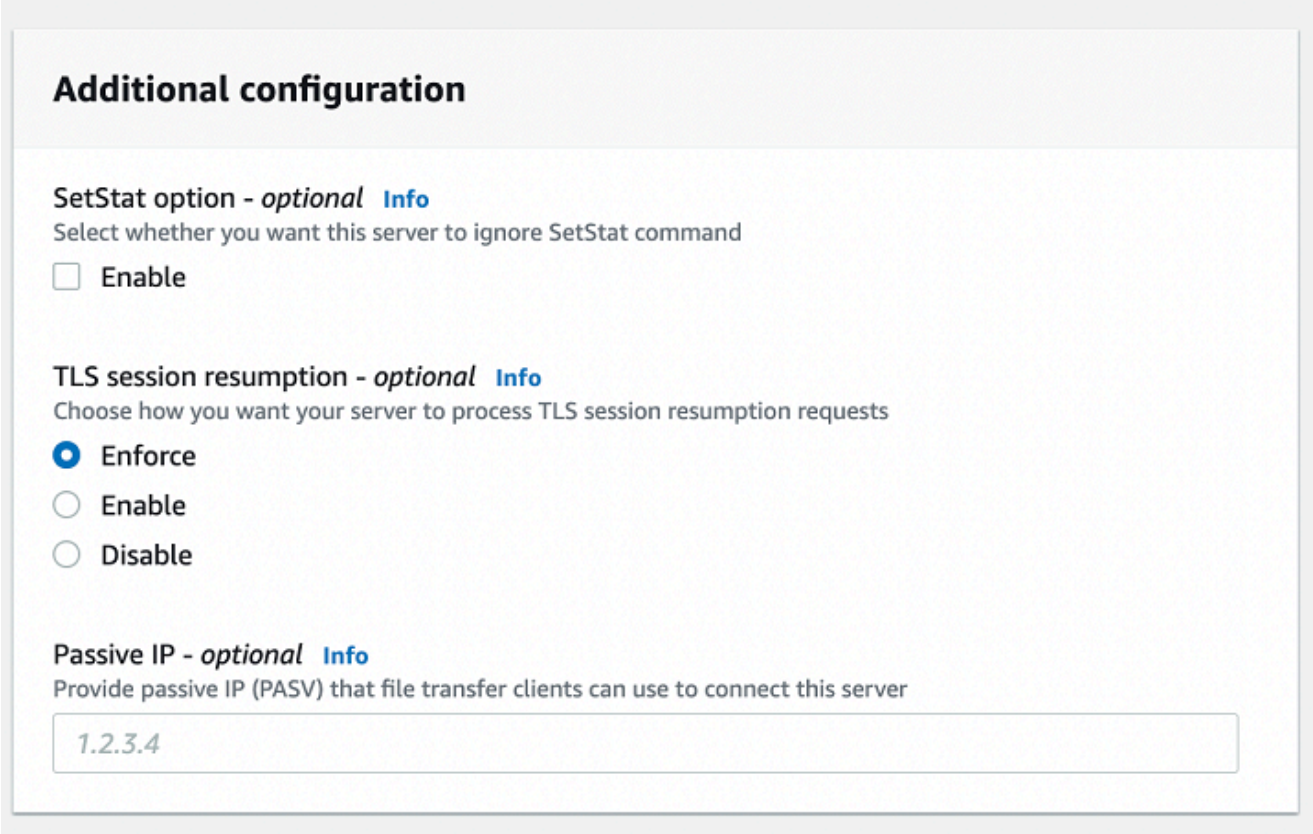
h. (선택 사항) 조직 정책 또는 이용 약관과 같은 사용자 지정 메시지를 최종 사용자에게 표시하도록 AWS Transfer Family 서버를 구성할 수 있습니다. 인증을 성공적으로 완료한 사용자에게 맞춤된 오늘의 메시지(MOTD)를 표시할 수도 있습니다.

디스플레이 배너의 경우 사전 인증 디스플레이 배너 텍스트 상자에 사용자가 인증하기 전에 표시할 텍스트 메시지를 입력하고, 인증 후 디스플레이 배너 텍스트 상자에는 인증에 성공한 후 사용자에게 표시할 텍스트를 입력합니다.

i. (옵션) 다음 추가 옵션을 구성할 수 있습니다.

- SetStat 옵션: Amazon S3 `SETSTAT` 버킷에 업로드하는 파일에서 클라이언트가 사용을 시도할 때 생성되는 오류를 무시하려면 이 옵션을 활성화합니다. 자세한 내용은 해당 ProtocolDetails주제의 `SetStatOption` 설명서를 참조하십시오.
- TLS 세션 재개: FTPS 세션에 대한 통제 및 데이터 연결 간에 협상된 암호 키를 재개하거나 공유하는 메커니즘을 제공합니다. 자세한 내용은 해당 ProtocolDetails항목의 `TlsSessionResumptionMode` 설명서를 참조하십시오.

- 수동 IP: FTP 및 FTPS 프로토콜에 대한 수동 모드를 나타냅니다. 방화벽, 라우터 또는 로드 밸런서의 퍼블릭 IP 주소와 같은 단일 IPv4 주소를 입력합니다. 자세한 내용은 해당 ProtocolDetails항목의 `PassiveIp` 설명서를 참조하십시오.



8. 검토 및 생성에서 옵션을 검토합니다.

   - 모든 항목을 편집하려면 단계 옆에 있는 편집을 선택합니다.

     > **Note**
     >
     > 편집하기로 선택한 단계 이후의 각 단계를 검토해야 합니다.

   - 변경 사항이 없는 경우 서버 생성을 선택하여 서버를 생성하세요. 서버 페이지로 이동하고, 새 서버가 나열되는 다음 화면이 표시됩니다.

새 서버 상태가 온라인으로 변경되기까지 몇 분 정도 걸릴 수 있습니다. 이때부터 서버는 사용자의 파일 작업을 수행할 수 있습니다.

다음 단계: 다음 단계에서는 사용자 지정 자격 증명 공급자와 작업를 계속 진행하여 사용자를 설정하세요.

FTP 지원 서버 생성

파일 전송 프로토콜(FTP)은 데이터 전송에 사용되는 네트워크 프로토콜입니다. FTP는 통제 및 데이터 전송에 별도의 채널을 사용합니다. 통제 채널은 종료되거나 비활성 제한 시간이 초과될 때까지 열려 있습니다. 데이터 채널은 전송 기간 동안 활성 상태입니다. FTP는 일반 텍스트를 사용하며 트래픽 암호화를 지원하지 않습니다.

> **Note**
>
> FTP를 활성화할 때는 vPC 호스팅 엔드포인트에 대한 내부 액세스 옵션을 선택해야 합니다. 서버에서 공용 네트워크를 통해 데이터를 전송해야 하는 경우 SFTP 또는 FTPS와 같은 보안 프로토콜을 사용해야 합니다.

FTP 지원 서버를 만들려면

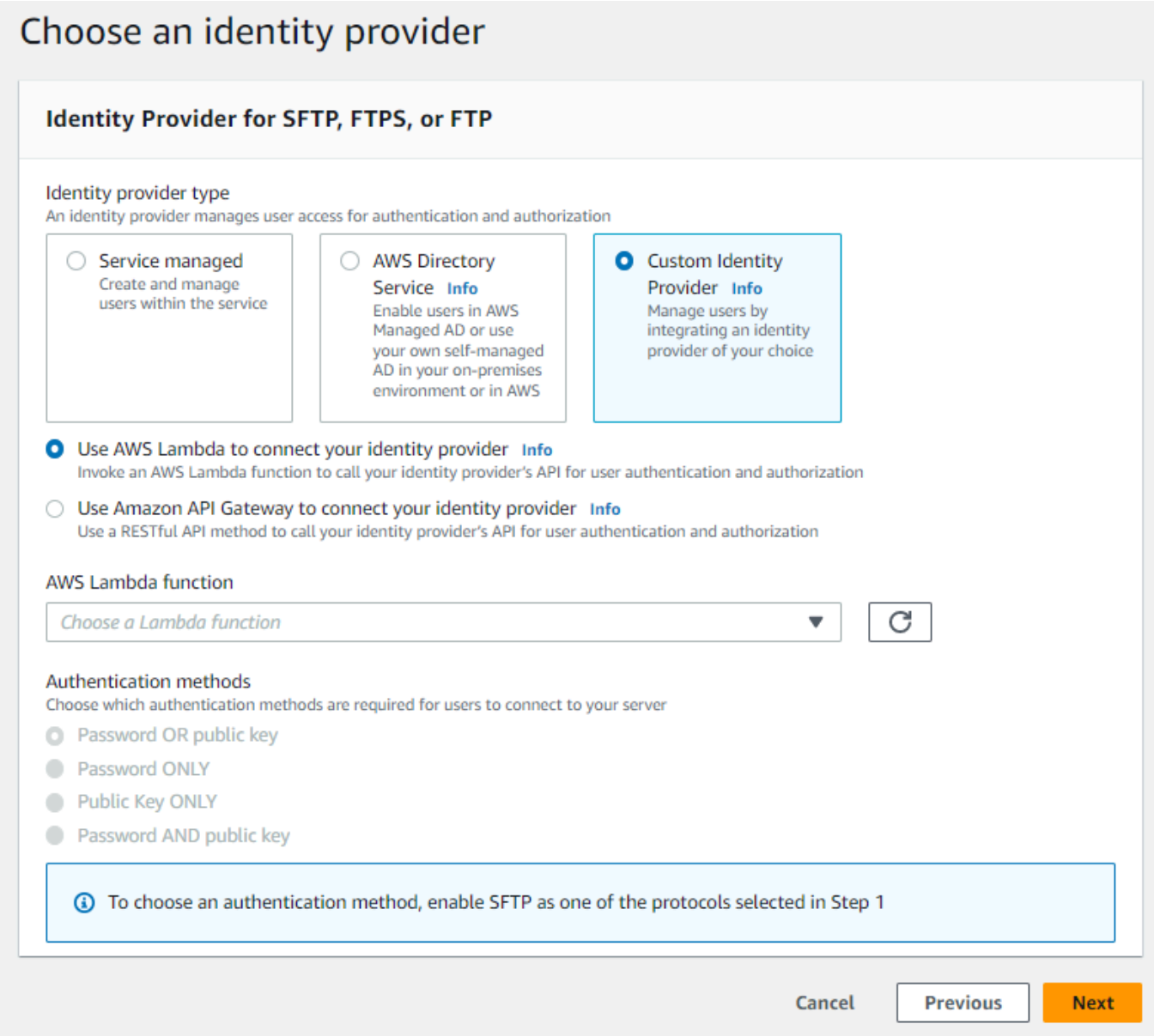
1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 열고 탐색 창에서 서버를 선택한 다음 서버 생성을 선택합니다.
2. 프로토콜 선택에서 FTP를 선택하고 다음을 선택합니다.
3. 자격 증명 공급자 선택에서 사용자 액세스를 관리하는 데 사용할 자격 증명 공급자를 선택합니다. 다음과 같은 옵션이 있습니다:

   - AWS Directory Service for Microsoft Active Directory— 엔드포인트에 액세스할 수 있는 AWS Directory Service 디렉터리를 제공합니다. 그러면 Active Directory에 저장된 자격 증명을 사용하여 사용자를 인증할 수 있습니다. AWS Managed Microsoft AD ID 공급자와의 작업에 대한 자세한 내용은 을 참조하십시오AWS 디렉터리 서비스 ID 제공자 사용.

     > **Note**
     >
     > - 교차 계정 및 공유 디렉터리는 지원되지 않습니다. AWS Managed Microsoft AD
     > - Directory Service를 ID 제공자로 사용하여 서버를 설정하려면 몇 가지 AWS Directory Service 권한을 추가해야 합니다. 자세한 내용은 사용을 시작하기 전 AWS Directory Service for Microsoft Active Directory단원을 참조하세요.

   - 맞춤 ID 공급자 – 다음 옵션 중 하나를 선택합니다.

- ID 공급자 AWS Lambda 연결에 사용 - Lambda 함수가 지원하는 기존 ID 공급자를 사용할 수 있습니다. 귀하는 Lambda 함수의 명칭을 제공합니다. 자세한 내용은 ID AWS Lambda 제공자를 통합하는 데 사용를 참조하세요.
- Amazon API Gateway를 사용하는 자격 증명 공급자 연결 - Lambda 함수가 지원하는 API Gateway 방법을 생성하여 자격 증명 공급자로 사용할 수 있습니다. 사용자는 Amazon API Gateway URL과 호출 역할을 제공합니다. 자세한 내용은 Amazon API Gateway를 ID 제공자 통합에 사용를 참조하세요.



4. 다음을 선택합니다.
5. 엔드포인트 선택에서 다음을 수행합니다.

> **Note**
>
> Transfer Family용 FTP 및 FTPS 서버는 포트 21(제어 채널) 및 포트 범위 8192-8200(데이터 채널)을 통해 작동합니다.

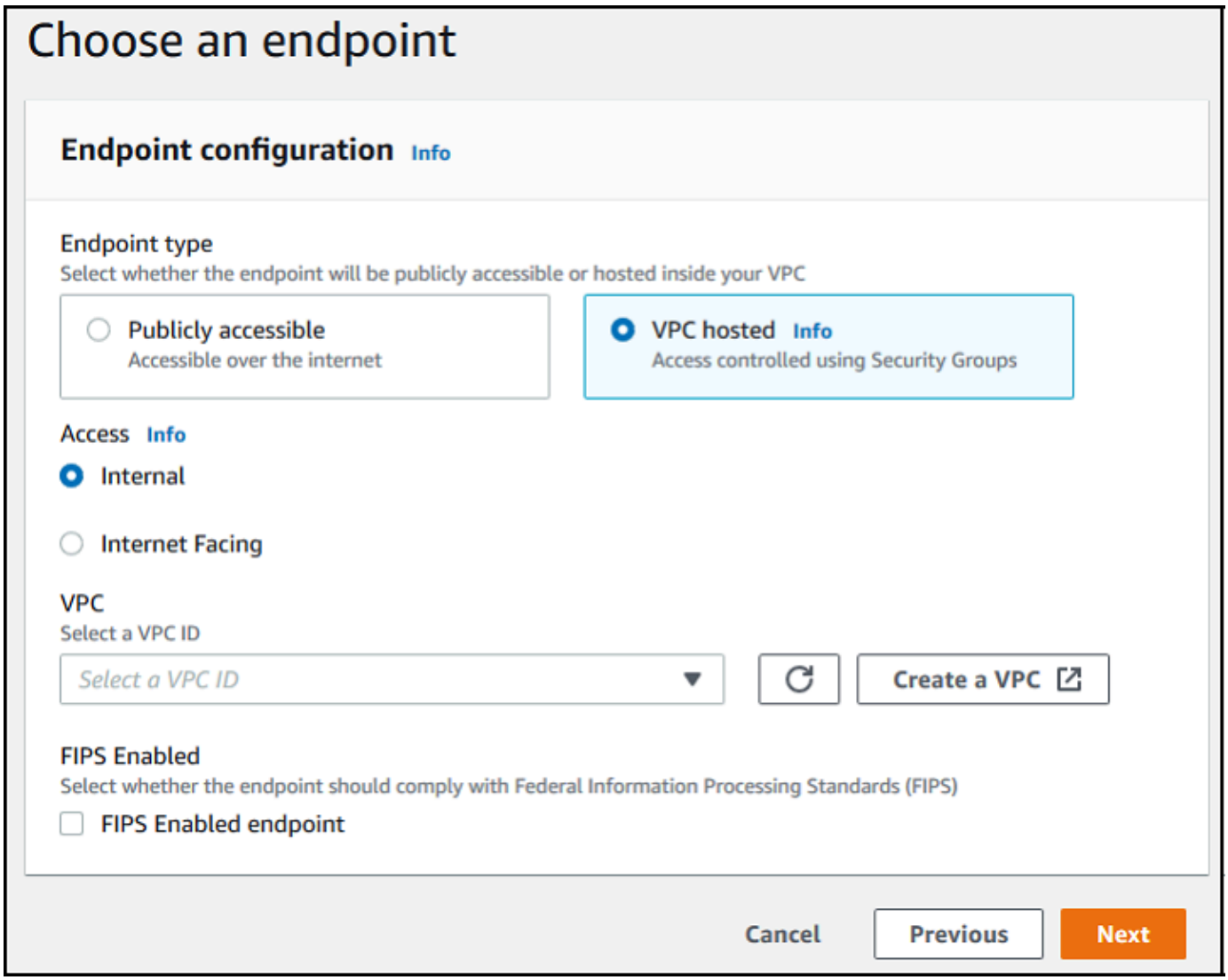
a. 엔드포인트 타입에서 서버의 엔드포인트를 호스팅할 VPC 호스팅을 선택합니다. VPC 호스팅 엔드포인트를 설정하는 자세한 설명은 Virtual Private Cloud(VPC)에 서버 생성 섹션을 참조하세요.

> **Note**
>
> 공적 액세스 가능 엔드포인트는 지원되지 않습니다.

b. FIPS 지원의 경우 FIPS 지원 엔드포인트 확인란의 선택을 취소한 상태로 유지합니다.

> **Note**
>
> FIPS 지원 엔드포인트는 FTP 서버에서 지원되지 않습니다.

c. 다음을 선택합니다.



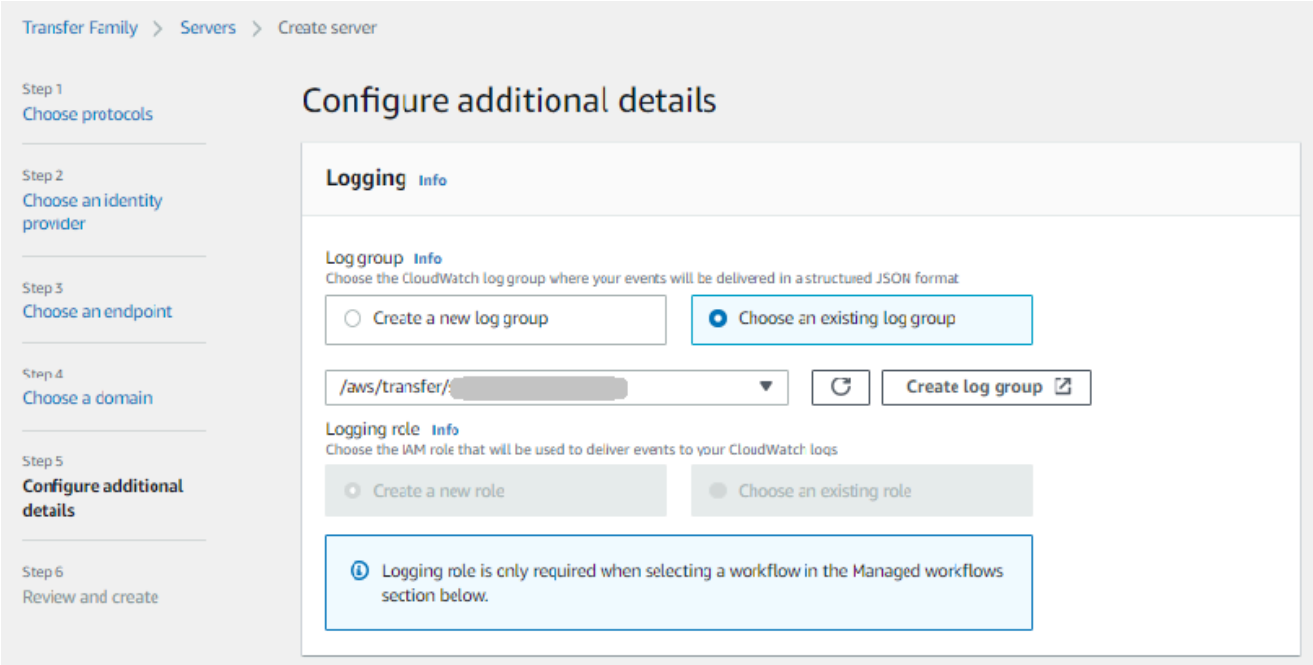
6. 도메인 선택 페이지에서 선택한 프로토콜을 통해 데이터를 저장하고 액세스하는 데 사용할 AWS 스토리지 서비스를 선택합니다.

   - 선택한 프로토콜을 통해 파일을 객체로 저장하고 액세스하려면 Amazon S3를 선택하세요.
   - 선택한 프로토콜을 통해 Amazon EFS 파일 시스템에 파일을 저장하고 액세스하려면 Amazon EFS를 선택합니다.

   다음을 선택합니다.

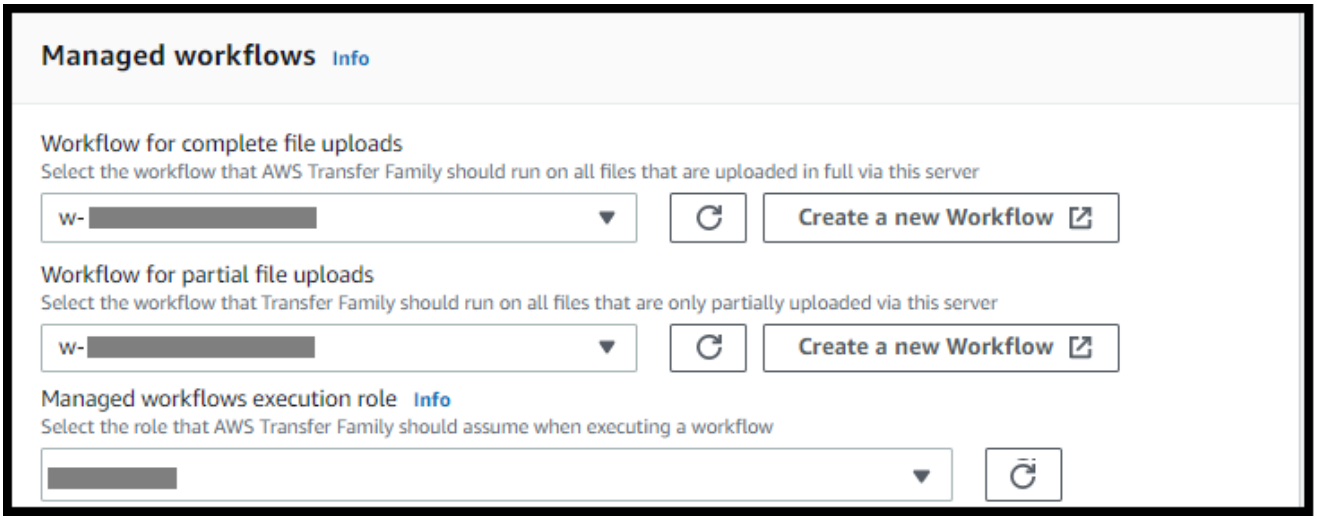
7. 추가 상세 정보 구성에서 다음을 수행합니다.

   a. 로깅의 경우 기존 로그 그룹을 지정하거나 새 로그 그룹을 생성합니다(기본 옵션).



로그 그룹 생성을 선택하면 CloudWatch 콘솔 (https://console.aws.amazon.com/cloudwatch/)에서 로그 그룹 생성 페이지가 열립니다. 자세한 내용은 로그에 CloudWatch 로그 그룹 만들기를 참조하십시오.

b. (옵션) 관리형 워크플로의 경우 워크플로를 실행할 때 Transfer Family가 맡아야 하는 워크플로 ID(및 해당 역할)를 선택합니다. 업로드 완료 시 실행할 워크플로 하나와 부분 업로드 시 실행할 워크플로 하나를 선택할 수 있습니다. 관리형 워크플로를 사용하여 파일을 처리하는 방법에 대한 자세한 내용은 AWS Transfer Family 관리형 워크플로를 참조하세요.



c. 암호화 알고리즘 옵션의 경우 서버에서 사용하도록 설정된 암호화 알고리즘이 포함된 보안 정책을 선택합니다.

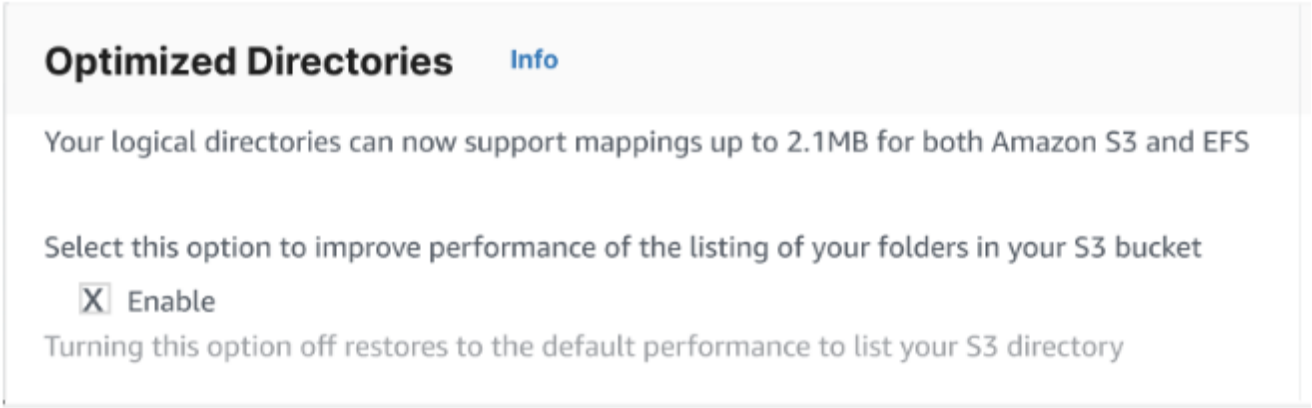
> **Note**
>
> Transfer Family는 FTP 서버에 최신 보안 정책을 할당합니다. 하지만 FTP 프로토콜은 암호화를 사용하지 않으므로 FTP 서버는 보안 정책 알고리즘을 사용하지 않습니다. 서버에서 FTPS 또는 SFTP 프로토콜도 사용하지 않는 한 보안 정책은 사용되지 않습니다.

d. 서버 호스트 키의 경우 비워 두세요.

e. (옵션) 태그의 키 및 값에서 하나 이상의 태그를 키-값 쌍을 입력한 다음 태그 추가를 선택합니다.

f. Amazon S3 디렉터리의 성능을 최적화할 수 있습니다. 예를 들어 홈 디렉터리에 10,000개의 하위 디렉터리가 있다고 가정해 보겠습니다. 즉, Amazon S3 버킷에는 10,000개의 폴더가 있습니다. 이 시나리오에서 `ls` (list) 명령을 실행하면 목록 작업에 6~8분이 소요됩니다. 하지만 디렉터리를 최적화하는 경우 이 작업은 몇 초밖에 걸리지 않습니다.

콘솔을 사용하여 서버를 생성하면 최적화된 디렉터리가 기본적으로 활성화됩니다. API를 사용하여 서버를 생성하는 경우 이 동작은 기본적으로 활성화되지 않습니다.



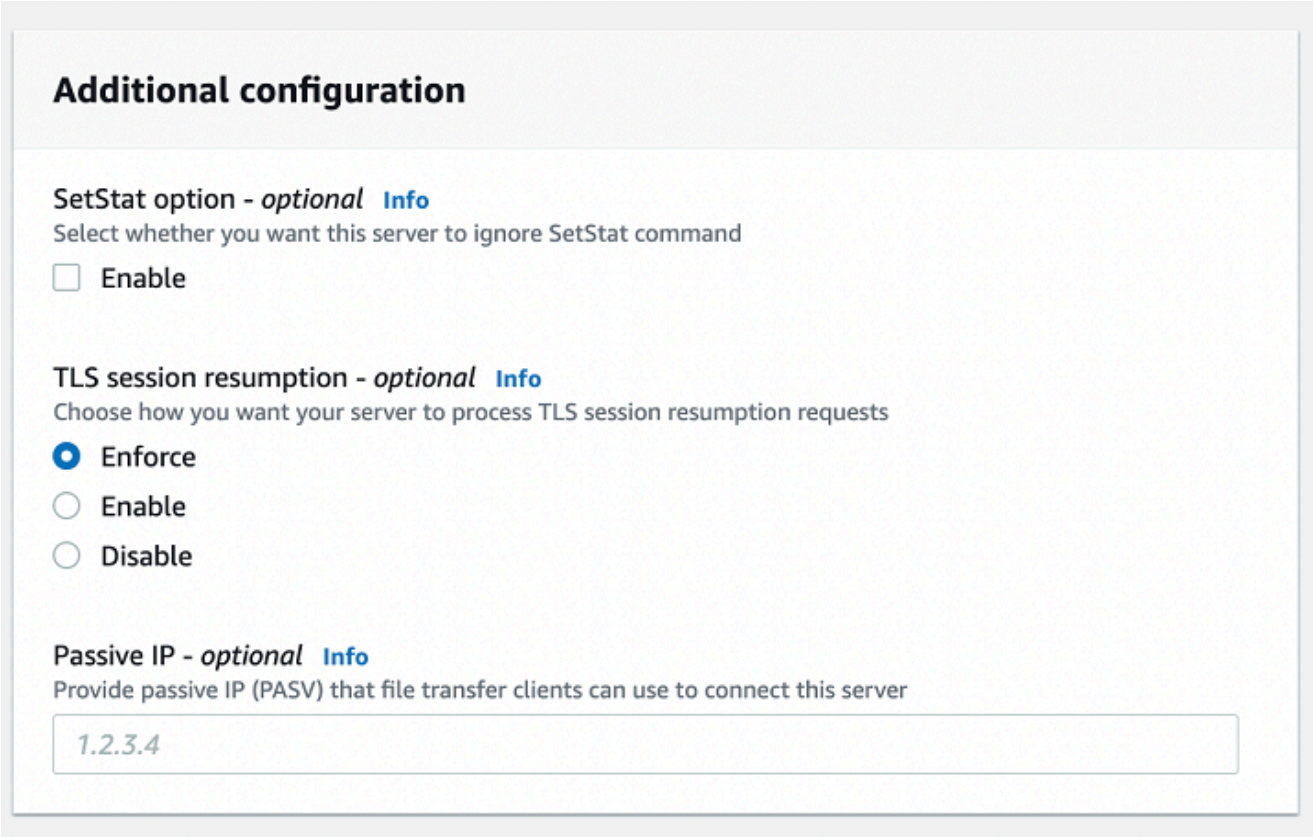
g. 다음을 선택합니다.

h. (선택 사항) 조직 정책 또는 이용 약관과 같은 사용자 지정 메시지를 최종 사용자에게 표시하도록 AWS Transfer Family 서버를 구성할 수 있습니다. 인증을 성공적으로 완료한 사용자에게 맞춤된 오늘의 메시지(MOTD)를 표시할 수도 있습니다.

디스플레이 배너의 경우 사전 인증 디스플레이 배너 텍스트 상자에 사용자가 인증하기 전에 표시할 텍스트 메시지를 입력하고, 인증 후 디스플레이 배너 텍스트 상자에는 인증에 성공한 후 사용자에게 표시할 텍스트를 입력합니다.

i. (옵션) 다음 추가 옵션을 구성할 수 있습니다.

- SetStat 옵션: Amazon S3 SETSTAT 버킷에 업로드하는 파일에서 클라이언트가 사용을 시도할 때 생성되는 오류를 무시하려면 이 옵션을 활성화합니다. 자세한 내용은 해당 ProtocolDetails주제의 SetStatOption 설명서를 참조하십시오.
- TLS 세션 재개: FTPS 세션에 대한 통제 및 데이터 연결 간에 협상된 암호 키를 재개하거나 공유하는 메커니즘을 제공합니다. 자세한 내용은 해당 ProtocolDetails항목의 TlsSessionResumptionMode 설명서를 참조하십시오.
- 수동 IP: FTP 및 FTPS 프로토콜에 대한 수동 모드를 나타냅니다. 방화벽, 라우터 또는 로드 밸런서의 퍼블릭 IP 주소와 같은 단일 IPv4 주소를 입력합니다. 자세한 내용은 해당 ProtocolDetails항목의 PassiveIp 설명서를 참조하십시오.



8. 검토 및 생성에서 옵션을 검토합니다.

- 모든 항목을 편집하려면 단계 옆에 있는 편집을 선택합니다.

> **Note**
>
> 편집하기로 선택한 단계 이후의 각 단계를 검토해야 합니다.

- 변경 사항이 없는 경우 서버 생성을 선택하여 서버를 생성하세요. 서버 페이지로 이동하고, 새 서버가 나열되는 다음 화면이 표시됩니다.

새 서버 상태가 온라인으로 변경되기까지 몇 분 정도 걸릴 수 있습니다. 이때부터 서버는 사용자의 파일 작업을 수행할 수 있습니다.

다음 단계 - 그 다음 단계에서는 사용자 지정 자격 증명 공급자와 작업로 계속 진행하여 사용자를 설정하세요.

Virtual Private Cloud(VPC)에 서버 생성

가상 사설 클라우드(VPC) 내에 서버의 엔드포인트를 호스팅하여 퍼블릭 인터넷을 거치지 않고 Amazon S3 버킷 또는 Amazon EFS 파일 시스템과 데이터를 주고 받는 데 사용할 수 있습니다.

> **Note**
>
> 2021년 5월 19일 이후, 계정에서 2021년 5월 19일 이전에 서버를 아직 생성하지 않은 경우 `EndpointType=VPC_ENDPOINT` AWS 계정에서 를 사용하여 서버를 생성할 수 없습니다. 2021년 2월 21일 또는 그 이전에 AWS 계정에 서버를 이미 생성한 경우 영향을 받지 않습니다. `EndpointType=VPC_ENDPOINT` 이 날짜 이후에는 `EndpointType=VPC`를 사용하세요. 자세한 정보는 the section called “VPC_ENDPOINT 사용 중단”을 참조하세요.

Amazon VPC (Virtual Private Cloud) 를 사용하여 AWS 리소스를 호스팅하는 경우 VPC와 서버 사이에 프라이빗 연결을 설정할 수 있습니다. 그러면 퍼블릭 IP 주소 지정을 사용하거나 인터넷 게이트웨이 필요 없이 이 서버를 사용하여 클라이언트를 통해 Amazon S3 버킷과 데이터를 주고 받을 수 있습니다.

Amazon VPC를 사용하면 사용자 지정 가상 네트워크에서 AWS 리소스를 시작할 수 있습니다. VPC를 사용하여 IP 주소 범위, 서브넷, 라우팅 테이블, 네트워크 게이트웨이 등의 네트워크 설정을 제어할 수 있습니다. VPC에 대한 자세한 내용은 Amazon VPC 사용 설명서의 Amazon VPC란 무엇인가요?를 참조하세요.

다음 섹션에는 VPC를 생성하고 서버에 연결하는 방법에 대한 지침이 나와 있습니다. 개요로 다음과 같이 이 작업을 수행합니다.

1. VPC 엔드포인트를 사용하여 서버를 설정합니다.
2. 그런 다음 VPC 엔드포인트를 통해 VPC 내부에 있는 클라이언트를 사용하여 서버에 연결합니다. 이렇게 하면 AWS Transfer Family를 사용하여 Amazon S3 버킷에 저장된 데이터를 클라이언트를 통

해 전송할 수 있습니다. 네트워크와 퍼블릭 인터넷 연결이 끊긴 경우에도 이 전송을 수행할 수 있습니다.

3. 또한 서버의 엔드포인트를 인터넷에 연결하도록 선택한 경우 탄력적 IP 주소를 엔드포인트와 연결할 수 있습니다. 이렇게 하면 VPC 외부의 클라이언트가 서버에 연결할 수 있습니다. VPC 보안 그룹을 사용하여 허용된 주소에서만 요청을 보내는 인증된 사용자에 대한 액세스를 제어할 수 있습니다.

주제

VPC 내에서만 액세스할 수 있는 서버 엔드포인트 생성

다음 절차에서는 VPC 내 리소스에만 액세스할 수 있는 서버 엔드포인트를 생성합니다.

VPC 내부에 서버 엔드포인트를 생성하려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 탐색 창에서 서버를 선택한 다음 서버 생성을 선택합니다.
3. 프로토콜 선택에서 하나 이상의 프로토콜을 선택한 후 다음을 선택합니다. 프로토콜에 대한 자세한 내용은 2단계: SFTP 지원 서버 생성를 참조하세요.
4. ID 공급자 선택에서 사용자 ID 및 키를 저장할 관리 서비스를 선택하고 다음을 선택합니다. AWS Transfer Family

> **Note**
>
> 이 절차에서는 서비스 관리형 옵션을 사용합니다. 사용자 지정을 선택하는 경우, Amazon API Gateway 엔드포인트와 AWS Identity and Access Management (IAM) 역할을 입력하여 엔드포인트에 액세스해야 합니다. 이렇게 하면 디렉터리 서비스를 통합해 사용자를 인증하고 승인할 수 있습니다. 사용자 지정 자격 증명 공급자와의 작업에 대한 자세한 내용은 사용자 지정 자격 증명 공급자와 작업를 참조하세요.

5. 엔드포인트 선택에서 다음을 수행합니다.

> **Note**
>
> Transfer Family용 FTP 및 FTPS 서버는 포트 21(제어 채널) 및 포트 범위 8192-8200(데이터 채널)을 통해 작동합니다.

a. 엔드포인트 타입에서 서버의 엔드포인트를 호스팅할 VPC 호스팅 엔드포인트 타입을 선택합니다.

b. 액세스에서 내부를 선택하여 엔드포인트의 사설 IP 주소를 사용하는 클라이언트만 엔드포인트에 액세스할 수 있도록 합니다.

> **Note**
>
> 인터넷 연결 옵션에 대한 자세한 내용은 서버용 인터넷 경계 엔드포인트 생성를 참조하세요. 내부 액세스 전용으로 VPC에 생성된 서버는 사용자 지정 호스트 이름을 지원하지 않습니다.

c. VPC의 경우 기존 VPC ID를 선택하거나 VPC 생성을 선택하여 새 VPC를 생성합니다.

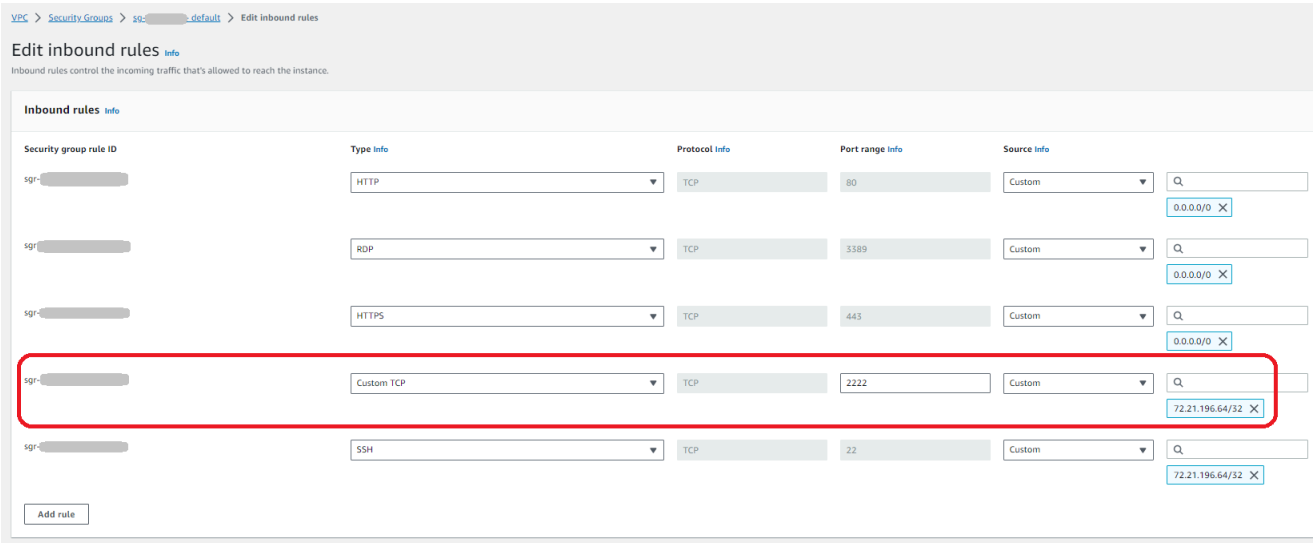
d. 가용 영역 섹션에서 가용 영역 및 관련 서브넷을 최대 3개까지 선택합니다.

e. 보안 그룹 섹션에서 기존 보안 그룹 ID 또는 ID를 선택하거나 보안 그룹 생성을 선택하여 새 보안 그룹을 생성합니다. 보안 그룹에 관한 자세한 내용은 Amazon Virtual Private Cloud 사용 설명서의 VPC의 보안 그룹을 참조하세요. 보안 그룹을 생성하려면 Amazon Virtual Private Cloud 사용 설명서의 보안 그룹 생성을 참조하세요.

> **Note**
>
> VPC는 자동으로 기본 보안 그룹과 함께 제공됩니다. 서버를 시작할 때 다른 보안 그룹 또는 그룹을 지정하지 않을 경우 기본 보안 그룹이 서버에 연결됩니다.

보안 그룹의 인바운드 규칙의 경우 포트 22, 2222, 22000 또는 임의의 조합을 사용하도록 SSH 트래픽을 구성할 수 있습니다. 포트 22는 기본적으로 구성됩니다. 포트 2222 또는 포트 22000을 사용하려면 보안 그룹에 인바운드 규칙을 추가합니다. 유형에는 사용자 지정 TCP를 선택한 다음 포트 범위 중 하나 **2222** 또는 **22000** 하나를 입력하고, 소스에는 SSH 포트 22 규칙과 동일한 CIDR 범위를 입력합니다.

> **Note**
>
> TCP “피기백” ACK가 필요한 클라이언트의 경우 포트 2223을 사용하거나 TCP 3-way 핸드셰이크의 최종 ack에 데이터도 포함할 수 있는 기능이 필요한 클라이언트의 경우 포트 2223을 사용할 수도 있습니다.
> 일부 클라이언트 소프트웨어는 포트 2223과 호환되지 않을 수 있습니다. 예를 들어, 클라이언트보다 먼저 서버에서 SFTP 식별 문자열을 전송해야 하는 클라이언트가 있습니다.



f. (옵션) FIPS 지원의 경우 FIPS 지원 엔드포인트 확인란을 선택하여 엔드포인트가 Federal Information Processing Standard(FIPS)를 준수하는지 확인합니다.

> **Note**
>
> FIPS 지원 엔드포인트는 북미 AWS 리전에서만 사용할 수 있습니다. 사용할 수 있는 리전은 AWS 일반 참조의 AWS Transfer Family 엔드포인트 및 할당량을 참조하세요. FIPS에 대한 자세한 설명은 미연방 정보 처리 표준(FIPS) 140-2를 참조하세요.

g. 다음을 선택합니다.

6. 추가 상세 정보 구성에서 다음을 수행합니다.

a. CloudWatch 로깅의 경우 다음 중 하나를 선택하여 Amazon의 사용자 활동 CloudWatch 로깅을 활성화하십시오.

- 새 역할을 생성할 수 있는 적절한 권한이 있는 한 Transfer Family에서 자동으로 IAM 역할을 생성할 수 있도록 새 역할을 생성합니다. 생성된 IAM 역할을 `AWSTransferLoggingAccess`(이)라고 합니다.
- 계정에서 기존 IAM 역할을 선택하려면 기존 역할을 선택합니다. 로깅 역할에서 역할을 선택합니다. 이 IAM 역할에는 서비스가 `transfer.amazonaws.com`(으)로 설정된 신뢰 정책이 포함되어야 합니다.

  CloudWatch 로깅에 대한 자세한 내용은 을 참조하십시오 CloudWatch 로깅 역할을 구성합니다..

> **Note**
>
> - 로깅 역할을 지정하지 CloudWatch 않으면 에서 최종 사용자 활동을 볼 수 없습니다.
> - 로깅 역할을 설정하지 않으려면 기존 역할 선택을 선택하고 CloudWatch 로깅 역할은 선택하지 마십시오.

b. 암호화 알고리즘 옵션의 경우 서버에서 사용하도록 설정된 암호화 알고리즘이 포함된 보안 정책을 선택합니다.

> **Note**
>
> 다른 정책을 선택하지 않는 한 `TransferSecurityPolicy-2020-06` 보안 정책이 기본적으로 서버에 연결됩니다.

보안 정책에 대한 자세한 설명은 섹션을 참조하세요서버 보안 정책 AWS Transfer Family.

c. (선택 사항: 이 섹션은 기존 SFTP 지원 서버에서 사용자를 마이그레이션하는 데만 사용됩니다.) 서버 호스트 키에는 클라이언트가 SFTP를 통해 서버에 연결할 때 서버를 식별하는 데 사용할 RSA, ED25519 또는 ECDSA 개인 키를 입력합니다.

d. (옵션) 태그의 키 및 값에서 하나 이상의 태그를 키-값 쌍을 입력한 다음 태그 추가를 선택합니다.

e. 다음을 선택합니다.

7. 검토 및 생성에서 옵션을 검토합니다. 경우에 따라 다음 작업을 수행합니다.

- 이들 중 하나를 편집하려면 단계 옆에 있는 편집을 선택합니다.

> **Note**
>
> 편집하기로 선택한 단계 이후의 각 단계를 검토해야 합니다.

- 변경 사항이 없으면 서버 생성을 선택하여 서버를 생성합니다. 서버 페이지로 이동하고, 새 서버가 나열되는 다음 화면이 표시됩니다.

새 서버 상태가 온라인으로 변경되기까지 몇 분 정도 걸릴 수 있습니다. 이때 서버에서 파일 작업을 수행할 수 있지만 먼저 사용자를 만들어야 합니다. 사용자 생성에 대한 자세한 내용은 을 참조하십시오. 서버 엔드포인트의 사용자 관리

서버용 인터넷 경계 엔드포인트 생성

다음 절차에서는 서버 엔드포인트를 만듭니다. 이 엔드포인트는 VPC의 기본 보안 그룹에서 원본 IP 주소를 허용하는 클라이언트만 인터넷을 통해 액세스할 수 있습니다. 또한 탄력적 IP 주소를 사용하여 엔드포인트가 인터넷 경계가 되도록 하면 클라이언트는 탄력적 IP 주소를 사용하여 방화벽에서 엔드포인트에 대한 액세스를 허용할 수 있습니다.

> **Note**
>
> 인터넷 경계 VPC 호스팅 엔드포인트에서는 SFTP와 FTPS만 사용할 수 있습니다.

인터넷 경계 엔드포인트를 생성하려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 탐색 창에서 서버를 선택한 다음 서버 생성을 선택합니다.
3. 프로토콜 선택에서 하나 이상의 프로토콜을 선택한 후 다음을 선택합니다. 프로토콜에 대한 자세한 내용은 2단계: SFTP 지원 서버 생성를 참조하세요.
4. ID 공급자 선택에서 사용자 ID 및 키를 저장할 관리 서비스를 선택하고 다음을 선택합니다. AWS Transfer Family

> **Note**
>
> 이 절차에서는 서비스 관리형 옵션을 사용합니다. 사용자 지정을 선택하는 경우, Amazon API Gateway 엔드포인트와 AWS Identity and Access Management (IAM) 역할을 입력하여 엔드포인트에 액세스해야 합니다. 이렇게 하면 디렉터리 서비스를 통합해 사용자를 인증하고 승인할 수 있습니다. 사용자 지정 자격 증명 공급자와의 작업에 대한 자세한 내용은 사용자 지정 자격 증명 공급자와 작업를 참조하세요.

5. 엔드포인트 선택에서 다음을 수행합니다.

   a. 엔드포인트 타입에서 서버의 엔드포인트를 호스팅할 VPC 호스팅 엔드포인트 타입을 선택합니다.

   b. 액세스의 경우 인터넷 연결을 선택하여 클라이언트가 인터넷을 통해 엔드포인트에 액세스할 수 있게 합니다.

   > **Note**
   >
   > 인터넷 연결을 선택하면 각 서브넷 또는 서브넷에서 기존 탄력적 IP 주소를 선택할 수 있습니다. 또는 VPC 콘솔(https://console.aws.amazon.com/vpc/)으로 이동하여 하나 이상의 새 탄력적 IP 주소를 할당할 수 있습니다. 이러한 주소는 본인이 소유할 수도 있고 본인이 소유할 AWS 수도 있습니다. 이미 사용 중인 탄력적 IP 주소는 엔드포인트와 연결할 수 없습니다.

   c. (옵션) 사용자 지정 호스트 이름의 경우 다음 중 하나를 선택합니다.

   > **Note**
   >
   > 고객은 엘라스틱 IP 주소를 통해 직접 연결하거나 Commercial Route 53 내에 EIP를 가리키는 호스트 이름 레코드를 생성해야 합니다. AWS GovCloud (US) GovCloud 엔드포인트에 Route 53을 사용하는 방법에 대한 자세한 내용은 사용 설명서의 AWS GovCloud (US) 리소스로 Amazon Route 53 설정을 참조하십시오.AWS GovCloud (US)

   - Amazon Route 53 DNS 별칭 – 사용하려는 호스트 이름이 Route 53에 등록된 경우. 이후 호스트 이름을 입력하면 됩니다.

- 기타 DNS – 사용할 호스트 이름이 다른 DNS 공급자를 통해 등록된 경우. 이후 호스트 이름을 입력하면 됩니다.
- 없음 – 서버의 엔드포인트를 사용하고 사용자 지정 호스트 이름은 사용하지 않을 경우. 서버 호스트 이름은 *server-id*.server.transfer.*region*.amazonaws.com 형식을 취합니다.

> **Note**
>
> 의 AWS GovCloud (US)고객의 경우 없음을 선택하면 이 형식의 호스트 이름이 생성되지 않습니다.

사용자 지정 호스트 이름 사용에 대한 자세한 내용은 사용자 지정 호스트 이름으로 작업를 참조하세요.

d. VPC의 경우 기존 VPC ID를 선택하거나 VPC 생성을 선택하여 새 VPC를 생성합니다.

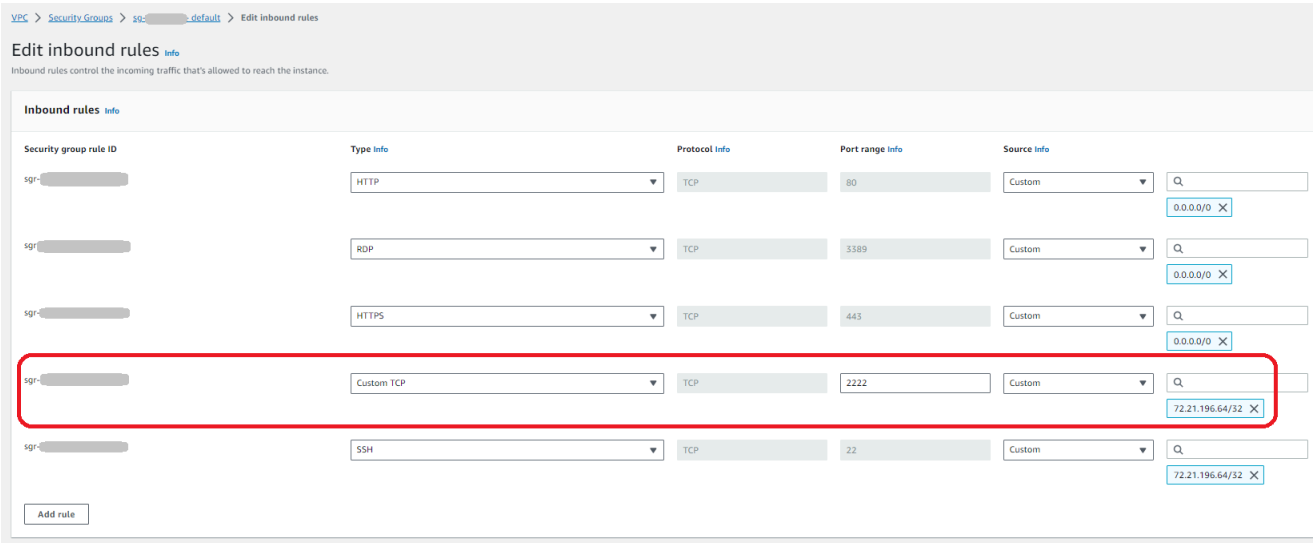
e. 가용 영역 섹션에서 가용 영역 및 관련 서브넷을 최대 3개까지 선택합니다. IPv4 주소의 경우 각 서브넷의 탄력적 IP 주소를 선택합니다. 클라이언트가 방화벽에서 엔드포인트에 대한 액세스를 허용하는 데 사용할 수 있는 IP 주소입니다.

f. 보안 그룹 섹션에서 기존 보안 그룹 ID 또는 ID를 선택하거나 보안 그룹 생성을 선택하여 새 보안 그룹을 생성합니다. 보안 그룹에 관한 자세한 내용은 Amazon Virtual Private Cloud 사용 설명서의 VPC의 보안 그룹을 참조하세요. 보안 그룹을 생성하려면 Amazon Virtual Private Cloud 사용 설명서의 보안 그룹 생성을 참조하세요.

> **Note**
>
> VPC는 자동으로 기본 보안 그룹과 함께 제공됩니다. 서버를 시작할 때 다른 보안 그룹 또는 그룹을 지정하지 않을 경우 기본 보안 그룹이 서버에 연결됩니다.

보안 그룹의 인바운드 규칙의 경우 포트 22, 2222, 22000 또는 임의의 조합을 사용하도록 SSH 트래픽을 구성할 수 있습니다. 포트 22는 기본적으로 구성됩니다. 포트 2222 또는 포트 22000을 사용하려면 보안 그룹에 인바운드 규칙을 추가합니다. 유형에는 사용자 지정 TCP를 선택한 다음 포트 범위 중 하나 **2222** 또는 **22000** 하나를 입력하고, 소스에는 SSH 포트 22 규칙과 동일한 CIDR 범위를 입력합니다.

> **Note**
>
> TCP “피기백” ACK가 필요한 클라이언트의 경우 포트 2223을 사용하거나 TCP 3-way 핸드셰이크의 최종 ack에 데이터도 포함할 수 있는 기능이 필요한 클라이언트의 경우 포트 2223을 사용할 수도 있습니다.
> 일부 클라이언트 소프트웨어는 포트 2223과 호환되지 않을 수 있습니다. 예를 들어, 클라이언트보다 먼저 서버에서 SFTP 식별 문자열을 전송해야 하는 클라이언트가 있습니다.



g. (옵션) FIPS 지원의 경우 FIPS 지원 엔드포인트 확인란을 선택하여 엔드포인트가 Federal Information Processing Standard(FIPS)를 준수하는지 확인합니다.

> **Note**
>
> FIPS 지원 엔드포인트는 북미 AWS 리전에서만 사용할 수 있습니다. 사용할 수 있는 리전은 AWS 일반 참조의 AWS Transfer Family 엔드포인트 및 할당량을 참조하세요. FIPS에 대한 자세한 설명은 미연방 정보 처리 표준(FIPS) 140-2를 참조하세요.

h. 다음을 선택합니다.

6. 추가 상세 정보 구성에서 다음을 수행합니다.

a. CloudWatch 로깅의 경우 다음 중 하나를 선택하여 Amazon의 사용자 활동 CloudWatch 로깅을 활성화하십시오.

- 새 역할을 생성할 수 있는 적절한 권한이 있는 한 Transfer Family에서 자동으로 IAM 역할을 생성할 수 있도록 새 역할을 생성합니다. 생성된 IAM 역할을 `AWSTransferLoggingAccess`(이)라고 합니다.
- 계정에서 기존 IAM 역할을 선택하려면 기존 역할을 선택합니다. 로깅 역할에서 역할을 선택합니다. 이 IAM 역할에는 서비스가 `transfer.amazonaws.com`(으)로 설정된 신뢰 정책이 포함되어야 합니다.

  CloudWatch 로깅에 대한 자세한 내용은 을 참조하십시오 CloudWatch 로깅 역할을 구성합니다..

> **Note**
>
> - 로깅 역할을 지정하지 CloudWatch 않으면 에서 최종 사용자 활동을 볼 수 없습니다.
> - 로깅 역할을 설정하지 않으려면 기존 역할 선택을 선택하고 CloudWatch 로깅 역할은 선택하지 마십시오.

b. 암호화 알고리즘 옵션의 경우 서버에서 사용하도록 설정된 암호화 알고리즘이 포함된 보안 정책을 선택합니다.

> **Note**
>
> 다른 정책을 선택하지 않는 한 `TransferSecurityPolicy-2020-06` 보안 정책이 기본적으로 서버에 연결됩니다.

보안 정책에 대한 자세한 설명은 섹션을 참조하세요서버 보안 정책 AWS Transfer Family.

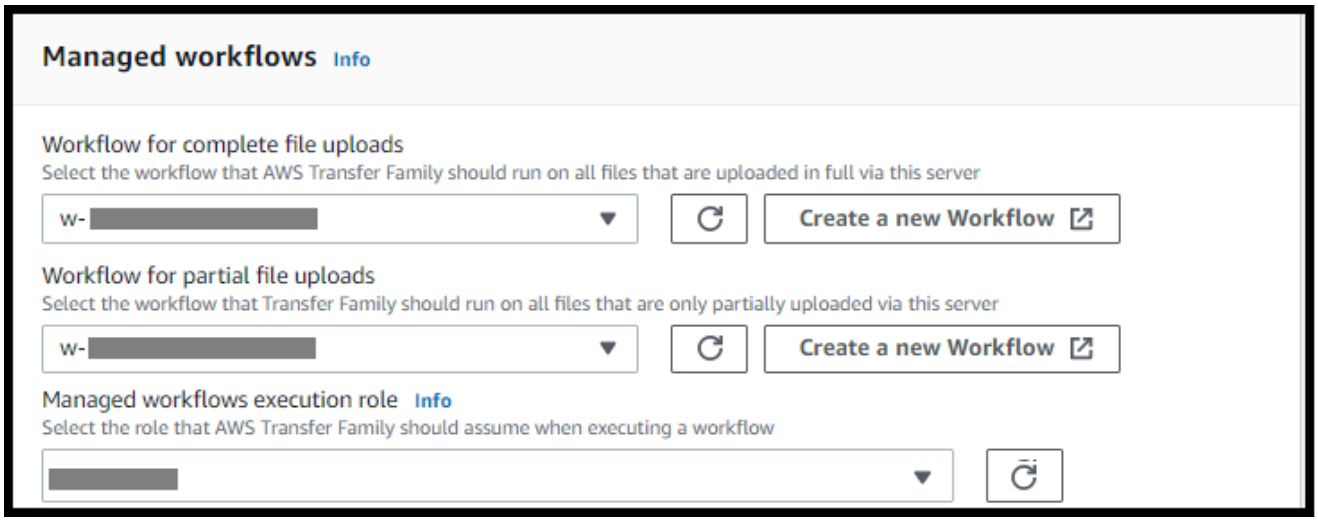
c. (선택 사항: 이 섹션은 기존 SFTP 지원 서버에서 사용자를 마이그레이션하는 데만 사용됩니다.) 서버 호스트 키에는 클라이언트가 SFTP를 통해 서버에 연결할 때 서버를 식별하는 데 사용할 RSA, ED25519 또는 ECDSA 개인 키를 입력합니다.

d. (옵션) 태그의 키 및 값에서 하나 이상의 태그를 키-값 쌍을 입력한 다음 태그 추가를 선택합니다.

e. 다음을 선택합니다.

f. (옵션) 관리형 워크플로의 경우 워크플로를 실행할 때 Transfer Family가 맡아야 하는 워크플로 ID(및 해당 역할)를 선택합니다. 업로드 완료 시 실행할 워크플로 하나와 부분 업로드 시 실

행할 워크플로 하나를 선택할 수 있습니다. 관리형 워크플로를 사용하여 파일을 처리하는 방법에 대한 자세한 내용은 AWS Transfer Family 관리형 워크플로를 참조하세요.



7. 검토 및 생성에서 옵션을 검토합니다. 경우에 따라 다음 작업을 수행합니다.

   - 이들 중 하나를 편집하려면 단계 옆에 있는 편집을 선택합니다.

     > **Note**
     >
     > 편집하기로 선택한 단계 이후의 각 단계를 검토해야 합니다.

   - 변경 사항이 없으면 서버 생성을 선택하여 서버를 생성합니다. 서버 페이지로 이동하고, 새 서버가 나열되는 다음 화면이 표시됩니다.

서버 ID를 선택하여 방금 생성한 서버의 세부 설정을 볼 수 있습니다. 퍼블릭 IPv4 주소 열이 채워지면 입력한 탄력적 IP 주소가 서버의 엔드포인트와 성공적으로 연결됩니다.

> **Note**
>
> VPC의 서버가 온라인 상태인 경우 API를 통해서만 서브넷을 수정할 수 있습니다. UpdateServer 서버 엔드포인트의 탄력적 IP 주소를 추가하거나 변경하려면 서버를 중지해야 합니다.

서버의 엔드포인트 타입 변경

인터넷을 통해 액세스할 수 있는 기존 서버가 있는 경우(즉, 퍼블릭 엔드포인트 타입이 있는 경우) 해당 엔드포인트를 VPC 엔드포인트로 변경할 수 있습니다.

> **Note**
>
> `VPC_ENDPOINT`로 표시된 VPC에 기존 서버가 있는 경우 새 VPC 엔드포인트 타입으로 수정하는 것이 좋습니다. 이 새 엔드포인트 타입을 사용하면 더 이상 NLB(Network Load Balancer)를 사용하여 탄력적 IP 주소를 서버의 엔드포인트와 연결할 필요가 없습니다. 또한 VPC 보안 그룹을 사용하여 서버 엔드포인트에 대한 액세스를 제한할 수 있습니다. 그러나 필요에 따라 `VPC_ENDPOINT` 엔드포인트 타입을 계속 사용할 수 있습니다.

다음 절차에서는 서버가 현재 퍼블릭 엔드포인트 타입 또는 이전 `VPC_ENDPOINT` 타입을 사용하는 것으로 가정합니다.

서버의 엔드포인트 타입을 변경하려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 탐색 창에서 서버를 선택합니다.
3. 엔드포인트 타입을 변경하려는 서버의 확인란을 선택합니다.

   > **Important**
   >
   > 엔드포인트를 변경하려면 먼저 서버를 중지해야 합니다.

4. 작업에서 중지를 선택합니다.
5. 표시되는 확인 대화 상자에서 중지를 선택하여 서버 중지를 확인합니다.

   > **Note**
   >
   > 다음 단계로 진행하기 전에 엔드포인트 세부 정보에서 서버 상태가 오프라인으로 변경될 때까지 기다리세요. 몇 분 정도 걸릴 수 있습니다. 상태 변경을 보려면 서버 페이지에서 새로 고침을 선택해야 할 수도 있습니다.
   > 서버가 오프라인이 될 때까지는 어떤 편집도 할 수 없습니다.

6. 엔드포인트 세부 정보에서 편집을 선택합니다.

7. 엔드포인트 구성 편집에서 다음을 수행합니다.

   a. 엔드포인트 타입 편집에서 VPC 호스팅을 선택합니다.

   b. 액세스에서 다음 중 하나를 선택합니다.

   - 엔드포인트의 사설 IP 주소를 사용하는 클라이언트만 엔드포인트에 액세스할 수 있도록 하려면 내부로 설정합니다.
   - 클라이언트가 퍼블릭 인터넷을 통해 엔드포인트에 액세스할 수 있도록 하려면 인터넷 연결을 선택합니다.

   > **Note**
   >
   > 인터넷 연결을 선택하면 각 서브넷 또는 서브넷에서 기존 탄력적 IP 주소를 선택할 수 있습니다. 또는 VPC 콘솔(https://console.aws.amazon.com/vpc/)로 이동하여 하나 이상의 새 탄력적 IP 주소를 할당할 수 있습니다. 이 주소는 본인이 소유할 수도 있고 본인이 소유할 수도 있습니다. AWS 이미 사용 중인 탄력적 IP 주소는 엔드포인트와 연결할 수 없습니다.

   c. (인터넷 연결 액세스 전용 옵션) 사용자 지정 호스트 이름의 경우 다음 중 하나를 선택합니다.

   - Amazon Route 53 DNS 별칭 – 사용하려는 호스트 이름이 Route 53에 등록된 경우. 이후 호스트 이름을 입력하면 됩니다.
   - 기타 DNS – 사용할 호스트 이름이 다른 DNS 공급자를 통해 등록된 경우. 이후 호스트 이름을 입력하면 됩니다.
   - 없음 – 서버의 엔드포인트를 사용하고 사용자 지정 호스트 이름은 사용하지 않을 경우. 서버 호스트 이름은 `serverId.server.transfer.regionId.amazonaws.com` 형식을 취합니다.

     사용자 지정 호스트 이름 사용에 대한 자세한 내용은 사용자 지정 호스트 이름으로 작업를 참조하세요.

   d. VPC의 경우 기존 VPC ID를 선택하거나 VPC 생성을 선택하여 새 VPC를 생성합니다.

   e. 가용 영역 섹션에서 가용 영역 및 관련 서브넷을 최대 3개까지 선택합니다. 인터넷 연결을 선택한 경우 각 서브넷의 탄력적 IP 주소도 선택합니다.

> **Note**
>
> 최대 3개의 가용 영역을 원하지만 가용 영역이 충분하지 않은 경우 VPC 콘솔(https://console.aws.amazon.com/vpc/)에서 가용 영역을 생성합니다.
> 서브넷 또는 탄력적 IP 주소를 수정할 경우 서버를 업데이트하는 데 몇 분 정도 걸립니다. 서버 업데이트가 완료될 때까지는 변경 내용을 저장할 수 없습니다.

f. 저장을 선택합니다.

8. 작업에서 시작을 선택하고 서버 상태가 온라인으로 변경될 때까지 기다립니다. 이 작업은 몇 분 정도 걸릴 수 있습니다.

> **Note**
>
> 퍼블릭 엔드포인트 타입을 VPC 엔드포인트 타입으로 변경한 경우 서버의 엔드포인트 타입이 VPC로 변경되었음을 알 수 있습니다.

기본 보안 그룹이 엔드포인트에 연결되어 있습니다. 보안 그룹을 변경하거나 추가하려면 보안 그룹 생성을 참조하세요.

VPC_ENDPOINT 사용 중단

AWS Transfer Family 새 AWS 계정을 `EndpointType=VPC_ENDPOINT` 위한 서버 생성 기능을 중단합니다. 2021년 5월 19일부터 엔드포인트 유형이 인 AWS Transfer Family 서버를 소유하지 않은 AWS 계정은 새 서버를 생성할 수 없습니다. `VPC_ENDPOINT` `EndpointType=VPC_ENDPOINT` `VPC_ENDPOINT` 엔드포인트 타입을 사용하는 서버를 이미 소유하고 있는 경우 가능한 한 빨리 `EndpointType=VPC` 사용을 시작하는 것이 좋습니다. 자세한 내용은 VPC_ENDPOINT에서 VPC로 AWS Transfer Family 서버 엔드포인트 유형 업데이트를 참조하십시오.

2020년 초에 새 VPC 엔드포인트 타입을 출시했습니다. 자세한 내용은 SFTP에 대한AWS Transfer Family 가 VPC 보안 그룹 및 탄력적 IP 주소를 지원하는 경우를 참조하세요. 이 새 엔드포인트는 기능이 더 풍부하고 비용 효율적이며 요금이 부과되지 않습니다. PrivateLink 자세한 내용은 AWS PrivateLink 요금을 참조하십시오.

이 엔드포인트 타입은 이전 엔드포인트 타입(`VPC_ENDPOINT`)과 기능적으로 동일합니다. 탄력적 IP 주소를 엔드포인트에 직접 연결하여 인터넷에 연결되도록 하고 소스 IP 필터링에 보안 그룹을 사용할 수 있습니다. 자세한 내용은 SFTP AWS Transfer Family 서버용 IP 사용 허용 목록 보안 블로그 게시물을 참조하십시오.

공유 VPC 환경에서 이 엔드포인트를 호스팅할 수도 있습니다. 자세한 내용은 AWS Transfer Family 에서 현재 공유 서비스 VPC 환경 지원을 참조하세요.

SFTP 외에도 VPC EndpointType를 사용하여 FTPS 및 FTP를 활성화할 수 있습니다. 당사는 이러한 기능과 FTPS/FTP 지원을 EndpointType=VPC_ENDPOINT에 추가할 계획이 없습니다. 또한 AWS Transfer Family 콘솔에서 이 엔드포인트 유형을 옵션으로 제거했습니다.

Transfer Family 콘솔, API AWS CLI, SDK 또는 AWS CloudFormation를 사용하여 서버의 엔드포인트 유형을 변경할 수 있습니다. 서버의 엔드포인트 타입을 변경하려면 VPC_ENDPOINT에서 VPC로 AWS Transfer Family 서버 엔드포인트 유형 업데이트를 참조하세요.

질문이 있는 경우, AWS Support 또는 AWS 계정 팀에 문의하세요.

> **Note**
>
> =VPC_ENDPOINT에 이러한 기능과 FTPS 또는 FTP 지원을 추가할 EndpointType 계획은 없습니다. 콘솔에서는 더 이상 옵션으로 제공하지 않습니다. AWS Transfer Family

추가 질문이 있는 경우, AWS Support 또는 계정 팀을 통해 문의할 수 있습니다.

VPC_ENDPOINT에서 VPC로 AWS Transfer Family 서버 엔드포인트 유형 업데이트

AWS Management Console AWS CloudFormation, 또는 Transfer Family API를 사용하여 서버의 EndpointType 에서 VPC_ENDPOINT 로 업데이트할 수 VPC 있습니다. 이러한 각 방법을 사용하여 서버 엔드포인트 타입을 업데이트하는 자세한 절차와 예가 다음 섹션에 나와 있습니다. 여러 AWS 지역과 여러 AWS 계정에 서버가 있는 경우 다음 섹션에 제공된 예제 스크립트를 수정하여 업데이트해야 하는 VPC_ENDPOINT 유형을 사용하는 서버를 식별할 수 있습니다.

주제

- VPC_ENDPOINT 엔드포인트 타입을 사용한 서버 식별
- 를 사용하여 서버 엔드포인트 유형 업데이트 AWS Management Console
- 를 사용하여 서버 엔드포인트 유형 업데이트 AWS CloudFormation
- API를 EndpointType 사용하여 서버 업데이트

VPC_ENDPOINT 엔드포인트 타입을 사용한 서버 식별

AWS Management Console을 사용하여 VPC_ENDPOINT를 사용하는 서버를 식별할 수 있습니다.

콘솔을 사용하여 **VPC_ENDPOINT** 엔드포인트 타입을 사용하는 서버를 식별하려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 탐색 창에서 서버를 선택하면 해당 리전의 사용자 계정에 있는 서버 목록이 표시됩니다.
3. 엔드포인트 타입별로 서버 목록을 정렬하면 VPC_ENDPOINT를 사용하는 모든 서버를 볼 수 있습니다.

VPC_ENDPOINT여러 AWS 지역 및 계정을 사용하는 서버를 식별하려면

여러 AWS 지역과 여러 AWS 계정에 서버가 있는 경우 다음 예제 스크립트를 수정하여 VPC_ENDPOINT 엔드포인트 유형을 사용하여 서버를 식별할 수 있습니다. 예제 스크립트는 Amazon EC2 DescribeRegions및 Transfer Family ListServers API 호출을 사용하여 사용 중인 모든 서버의 서버 ID 및 지역 목록을 가져옵니다. VPC_ENDPOINT AWS 계정이 많은 경우 ID 공급자에 대한 세션 프로필을 사용하여 인증하면 읽기 전용 감사자 액세스 권한이 있는 IAM 역할을 사용하여 계정을 순회할 수 있습니다.

1. 다음은 간단한 예입니다.

```
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. 업데이트할 서버 목록을 확보한 후에는 다음 섹션에 설명된 방법 중 하나를 사용하여 EndpointType을 VPC(으)로 업데이트할 수 있습니다.

를 사용하여 서버 엔드포인트 유형 업데이트 AWS Management Console

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 탐색 창에서 서버를 선택합니다.
3. 엔드포인트 타입을 변경하려는 서버의 확인란을 선택합니다.

> **Important**
>
> 엔드포인트를 변경하려면 먼저 서버를 중지해야 합니다.

4. 작업에서 중지를 선택합니다.
5. 표시되는 확인 대화 상자에서 중지를 선택하여 서버 중지를 확인합니다.

> **Note**
>
> 다음 단계로 진행하기 전에 서버 상태가 오프라인으로 변경될 때까지 기다리세요. 몇 분 정도 걸릴 수 있습니다. 상태 변경을 보려면 서버 페이지에서 새로 고침을 선택해야 할 수도 있습니다.

6. 상태가 오프라인으로 변경되면 서버 세부 정보 페이지를 표시할 서버를 선택합니다.
7. 엔드포인트 세부 정보 섹션에서 편집을 선택합니다.
8. 엔드포인트 타입으로 VPC 호스팅을 선택합니다.
9. 저장을 선택합니다.
10. 작업에서 시작을 선택하고 서버 상태가 온라인으로 변경될 때까지 기다립니다. 이 작업은 몇 분 정도 걸릴 수 있습니다.

를 사용하여 서버 엔드포인트 유형 업데이트 AWS CloudFormation

이 섹션에서는 서버를 AWS CloudFormation 업데이트하는 데 사용하는 방법을 설명합니다VPC. `EndpointType` 를 사용하여 배포한 Transfer Family 서버의 경우 이 절차를 사용하십시오 AWS CloudFormation. 이 예에서 Transfer Family 서버를 배포하는 데 사용된 원본 AWS CloudFormation 템플릿은 다음과 같습니다.

```
AWS TemplateFormatVersion: '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
```

템플릿이 다음과 같이 변경되어 업데이트되었습니다.

- `EndpointType`이(가) `VPC`(으)로 변경되었습니다.
- `AWS::EC2::VPCEndpoint` 리소스가 제거됩니다.
- `SecurityGroupId`, `SubnetIds`, 및 `VpcId`(이)가 `AWS::Transfer::Server` 리소스의 `EndpointDetails` 섹션으로 이동되었습니다.
- `EndpointDetails`의 `VpcEndpointId` 속성이 삭제되었습니다.

업데이트된 템플릿입니다.

```
AWS TemplateFormatVersion: '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
```

를 사용하여 배포한 Transfer Family 서버의 엔드포인트 유형을 업데이트하려면 AWS CloudFormation

1. 다음 단계를 사용하여 업데이트하려는 서버를 중지합니다.

   a. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.

   b. 탐색 창에서 서버를 선택합니다.

   c. 엔드포인트 타입을 변경하려는 서버의 확인란을 선택합니다.

   > **Important**
   >
   > 엔드포인트를 변경하려면 먼저 서버를 중지해야 합니다.

   d. 작업에서 중지를 선택합니다.

   e. 표시되는 확인 대화 상자에서 중지를 선택하여 서버 중지를 확인합니다.

> **Note**
>
> 다음 단계로 진행하기 전에 서버 상태가 오프라인으로 변경될 때까지 기다리세요. 몇 분 정도 걸릴 수 있습니다. 상태 변경을 보려면 서버 페이지에서 새로 고침을 선택해야 할 수도 있습니다.

2. CloudFormation 스택 업데이트

   a. https://console.aws.amazon.com/cloudformation 에서 AWS CloudFormation 콘솔을 엽니다.

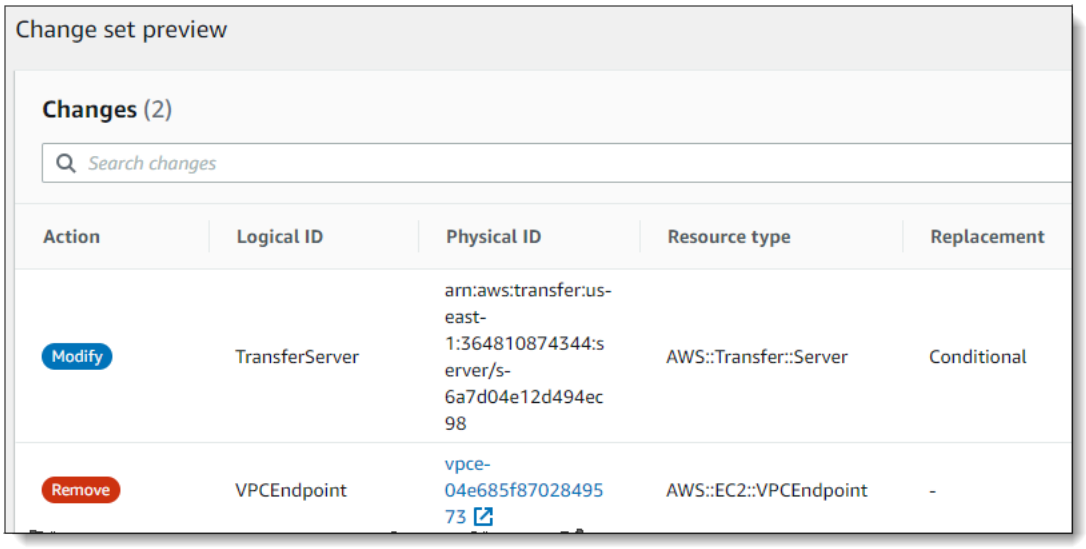
   b. Transfer Family 서버를 생성하는 데 사용할 스택을 선택합니다.

   c. 업데이트를 선택합니다.

   d. 현재 템플릿 교체를 선택합니다.

   e. 새 템플릿을 업로드합니다. CloudFormation 변경 세트를 사용하면 템플릿 변경을 구현하기 전에 실행 중인 리소스에 미치는 영향을 이해하는 데 도움이 됩니다. 이 예에서는 전송 서버 리소스가 수정되고 VPCendPoint 리소스가 제거됩니다. VPC 엔드포인트 타입 서버는 사용자를 대신하여 VPC 엔드포인트를 생성하여 원본 `VPCEndpoint` 리소스를 대체합니다.

      새 템플릿을 업로드한 후 변경 세트는 다음과 유사합니다.



   f. 스택을 업데이트합니다.

3. 스택 업데이트가 완료되면 Transfer Family 관리 콘솔 https://console.aws.amazon.com/transfer/ 로 이동합니다.

4. 서버를 시작합니다. 업데이트한 서버를 선택한 다음 작업 메뉴에서 시작을 선택합니다. AWS CloudFormation

API를 EndpointType 사용하여 서버 업데이트

describe-server AWS CLI 명령 또는 UpdateServer API 명령을 사용할 수 있습니다. 다음 예제 스크립트는 Transfer Family 서버를 중지하고, 업데이트하고 EndpointType, VPC_ENDPOINT를 제거하고, 서버를 시작합니다.

```
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])
```

사용자 지정 호스트 이름으로 작업

서버 호스트 이름은 사용자가 서버에 연결할 때 클라이언트에 입력하는 호스트 이름입니다. 작업할 때 서버 호스트 이름으로 등록한 사용자 지정 도메인을 사용할 수 있습니다. AWS Transfer Family예를 들어 `mysftpserver.mysubdomain.domain.com` 같은 사용자 지정 호스트 이름을 사용할 수 있습니다.

등록한 사용자 지정 도메인에서 서버 엔드포인트로 트래픽을 리디렉션하려면, Amazon Route 53 또는 아무 DNS(Domain Name System) 공급자를 이용하면 됩니다. Route 53은 AWS Transfer Family (이)가 기본적으로 지원하는 DNS 서비스입니다.

주제

- Amazon Route 53을 DNS 공급자로 사용
- 다른 DNS 공급자 사용
- 콘솔에서 생성하지 않은 서버의 사용자 지정 호스트 이름

콘솔에서, 사용자 지정 호스트 이름을 설정하려면 다음 방법 중 하나를 선택합니다.

- Amazon Route 53 DNS 별칭 – 사용하려는 호스트 이름이 Route 53에 등록된 경우. 이후 호스트 이름을 입력하면 됩니다.
- 기타 DNS – 사용할 호스트 이름이 다른 DNS 공급자를 통해 등록된 경우. 이후 호스트 이름을 입력하면 됩니다.
- 없음 – 서버의 엔드포인트를 사용하고 사용자 지정 호스트 이름은 사용하지 않을 경우.

이 옵션은 새 서버를 만들거나 기존 서버의 구성을 수정할 때 설정합니다. 새 서버 생성에 대한 자세한 정보는 2단계: SFTP 지원 서버 생성 섹션을 참조하세요. 기존 서버 구성 편집에 대한 자세한 내용은 서버 세부 정보 편집 섹션을 참조하세요.

서버 호스트 이름에 자체 도메인을 사용하는 방법과 Route 53을 AWS Transfer Family 사용하는 방법에 대한 자세한 내용은 다음 섹션을 참조하십시오.

Amazon Route 53을 DNS 공급자로 사용

서버를 만들 때 DNS 공급자로 Amazon Route 53을 사용할 수 있습니다. Route 53을 도메인과 함께 사용하려면 먼저 도메인을 등록합니다. 자세한 내용은 Amazon Route 53 개발자 안내서의 도메인 등록 방식을 참조하세요.

Route 53을 사용하여 서버에 DNS 라우팅을 제공하는 경우 는 호스팅 영역을 추출하기 위해 입력한 사용자 지정 호스트 이름을 AWS Transfer Family 사용합니다. 호스팅 영역을 AWS Transfer Family 추출할 때 다음과 같은 세 가지 상황이 발생할 수 있습니다.

1. Route 53을 처음 사용하고 호스팅 영역이 없는 경우, AWS Transfer Family 새 호스팅 영역과 CNAME 레코드를 추가하십시오. 이 CNAME 레코드의 값은 서버의 엔드포인트 호스트 이름입니다. CNAME은 대체 도메인 이름입니다.
2. Route 53에 호스팅 영역이 있지만 CNAME 레코드가 없는 경우, AWS Transfer Family 에서 호스팅 영역에 CNAME 레코드를 추가합니다.
3. 서비스가 호스팅 영역에 존재하는 CNAME 레코드를 감지하면, CNAME 레코드가 이미 존재함을 알려주는 오류 메시지가 표시됩니다. 이 경우, CNAME 레코드 값을 서버의 호스트 이름으로 변경합니다.

Route 53의 호스팅 영역에 대한 자세한 내용은 Amazon Route 53 개발자 안내서의 호스팅 영역을 참조하하세요.

다른 DNS 공급자 사용

서버를 생성할 때 Amazon Route 53 대신 DNS 공급자를 사용할 수도 있습니다. 대체 DNS 공급자를 사용하는 경우 도메인에서 나오는 트래픽이 서버 엔드포인트로 가는지 확인합니다.

이렇게 하려면 도메인을 서버의 엔드포인트 호스트 이름으로 설정합니다. 엔드포인트 호스트 이름은 콘솔에서 형식으로 표시됩니다.

```
serverid.server.transfer.region.amazonaws.com
```

> **Note**
>
> 서버에 VPC 엔드포인트가 있는 경우 호스트 이름의 형식은 위에서 설명한 형식과 다릅니다. VPC 엔드포인트를 찾으려면 서버의 세부 정보 페이지에서 VPC를 선택한 다음 VPC 대시보드에서 VPC 엔드포인트 ID를 선택합니다. 엔드포인트는 나열된 DNS 이름 중 첫 번째 DNS 이름입니다.

콘솔에서 생성하지 않은 서버의 사용자 지정 호스트 이름

또는 CLI를 사용하거나 AWS Cloud Development Kit (AWS CDK) CLI를 통해 서버를 생성할 때 해당 서버에 사용자 지정 호스트 이름을 지정하려면 태그를 추가해야 합니다. AWS CloudFormation콘솔을 사용하여 Transfer Family 서버를 생성하는 경우 태깅이 자동으로 수행됩니다.

> **Note**
>
> 또한 도메인에서 서버 엔드포인트로 트래픽을 리디렉션하려면 DNS 레코드를 생성해야 합니다. 자세한 내용은 Amazon Route 53 개발자 안내서의 레코드 작업을 참조하십시오.

사용자 지정 호스트 이름에 사용할 키:

- 콘솔에 사용자 지정 호스트 이름을 표시하려면 `transfer:customHostname`를 추가합니다.
- Route 53을 DNS 공급자로 사용하고 있는 경우 `transfer:route53HostedZoneId`를 추가합니다. 이 태그는 사용자 지정 호스트 이름을 Route 53 호스팅 영역 ID에 연결합니다.

사용자 지정 호스트 이름을 추가하려면 다음 CLI 명령을 실행합니다.

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

예:

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

Route 53을 사용하는 경우 다음 명령을 실행하여 사용자 지정 호스트 이름을 Route 53 호스팅 영역 ID에 연결합니다.

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

예:

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

이전 명령의 샘플 값을 가정하고 다음 명령을 실행하여 태그를 봅니다.

```
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**
>
> Amazon Route 53에서 퍼블릭, 호스팅 영역 및 해당 ID를 사용할 수 있습니다. AWS Management Console 로그인하고 https://console.aws.amazon.com/route53/ 에서 Route 53 콘솔을 엽니다.

클라이언트를 사용하여 서버 엔드포인트를 통한 파일 전송

클라이언트에서 전송 작업을 지정하여 AWS Transfer Family 서비스를 통해 파일을 전송합니다. AWS Transfer Family 다음 클라이언트를 지원합니다.

- SFTP 프로토콜 버전 3을 지원합니다.
- OpenSSH (macOS 및 Linux)

  > **Note**
  >
  > 이 클라이언트는 Secure Shell (SSH) File Transfer Protocol(SSH)를 지원하는 서버에서만 작동합니다.

- WinSCP (Microsoft Windows 전용)
- Cyberduck (Windows, macOS, 및 Linux)
- FileZilla (윈도우, 맥OS, 리눅스)

모든 클라이언트에는 다음과 같은 제한 사항이 적용됩니다:

- 연결당 최대 동시 멀티플렉스 SFTP 세션 수는 10개입니다.
- SFTP/FTP/FTPS 연결에는 두 가지 타임아웃 값이 있습니다. 유휴 연결의 경우 제한 시간 값은 1800초 (30분) 입니다. 기간이 지난 후에도 활동이 없으면 클라이언트 연결이 끊길 수 있습니다. 또한 클라이언트가 완전히 응답하지 않는 경우 300초 (5분) 의 제한 시간이 발생합니다.
- Amazon S3와 Amazon EFS는 (NFSv4 프로토콜로 인해) 파일 명칭이 UTF-8 인코딩이어야 합니다. 다른 인코딩을 사용하면 예상치 못한 결과가 발생할 수 있습니다. Amazon S3의 경우, 객체 키 명칭 지정 가이드라인을 참조하세요.
- SSL을 통한 파일 전송 프로토콜 (FTPS) 의 경우, 명시적 모드만 지원됩니다. 묵시적 모드는 지원되지 않습니다.
- FTP (파일 전송 프로토콜) 및 FTPS의 경우, 패시브 모드만 지원됩니다.
- FTP 및 FTPS의 경우, 스트림 모드만 지원됩니다.
- FTP 및 FTPS의 경우, 이미지/바이너리 모드만 지원됩니다.
- FTP 및 FTPS의 경우 데이터 연결을 위한 TLS - PROT C (보호되지 않음) TLS가 기본값이지만 FTPS 프로토콜에서는 PROT C가 지원되지 않습니다. AWS Transfer Family 따라서 FTPS의 경우, 데이터 작업이 승인되려면 PROT P를 발급해야 합니다.
- Amazon S3를 서버 스토리지로 사용하고 있고 클라이언트에 단일 전송에 복수 연결을 사용하는 옵션이 포함되어 있는 경우, 해당 옵션을 비활성화해야 합니다. 그렇지 않으면 대용량 파일 업로드가 예상치 못한 방식으로 실패할 수 있습니다. Amazon EFS를 스토리지 백엔드로 사용하는 경우, EFS는 단일 전송에 대해 복수 연결을 지원합니다.

FTP 및 FTPS에 사용할 수 있는 명령 목록은 다음과 같습니다:

가용 명령					
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	타입

가용 명령					
EPSV	MLSD	OPTS	QUIT	STAT	USER

> **Note**
>
> APPE는 지원되지 않습니다.

SFTP의 경우, Amazon Elastic File System(Amazon EFS)을 사용하는 서버의 논리적 홈 디렉터리를 사용하는 사용자에게는 현재 다음 작업이 지원되지 않습니다.

지원되지 않는 SFTP 명령			
SSH_FXP_READLINK	SSH_FXP_SYMLINK	요청된 파일이 심볼릭 링크인 경우, SSH_FXP_STAT	요청된 경로에 심볼릭 링크 구성 요소가 포함된 경우, SSH_FXP_REALPATH

퍼블릭-프라이빗 키 쌍을 생성합니다.

파일을 전송하려면 먼저 퍼블릭-프라이빗 키 쌍을 사용할 수 있어야 합니다. 이전에 키 쌍을 생성하지 않은 경우, 서비스 관리 사용자를 위한 SSH 키 생성를 참조하세요.

주제

- 가용 SFTP/FTPS/FTP 명령
- Amazon VPC 엔드포인트 찾기
- setstat 오류 방지
- OpenSSH 사용
- WinSCP 사용
- Cyberduck 사용하기
- 사용 FileZilla
- Perl 클라이언트 사용
- 업로드 후 처리

가용 SFTP/FTPS/FTP 명령

다음 표에서는 SFTP AWS Transfer Family, FTPS 및 FTP 프로토콜에 사용할 수 있는 명령을 설명합니다.

> **Note**
>
> 이 표에는 버킷과 객체만 지원하는 Amazon S3의 파일 및 디렉터리가 나와 있습니다. 계층 구조는 없습니다. 그러나 객체 키 명칭에 접두사를 사용하여 계층 구조를 암시하고 폴더와 유사한 방식으로 데이터를 구성할 수 있습니다. 이 동작은 Amazon Simple Storage Service 사용자 가이드에서 객체 메타데이터 작업에 설명되어 있습니다.

SFTP/FTPS/FTP 명령

Command	Amazon S3	Amazon EFS
`cd`	지원	지원
`chgrp`	지원되지 않음	지원(`root` 또는 `owner`만 해당)
`chmod`	지원되지 않음	지원(`root`만 해당)
`chmtime`	지원되지 않음	지원
`chown`	지원되지 않음	지원(`root`만 해당)
`get`	지원	지원(심볼 링크 해결 포함)
`ln -s`	지원되지 않음	지원
`ls/dir`	지원	지원
`mkdir`	지원	지원
`put`	지원	지원
`pwd`	지원	지원
`rename`	파일에만 지원됩니다.	지원

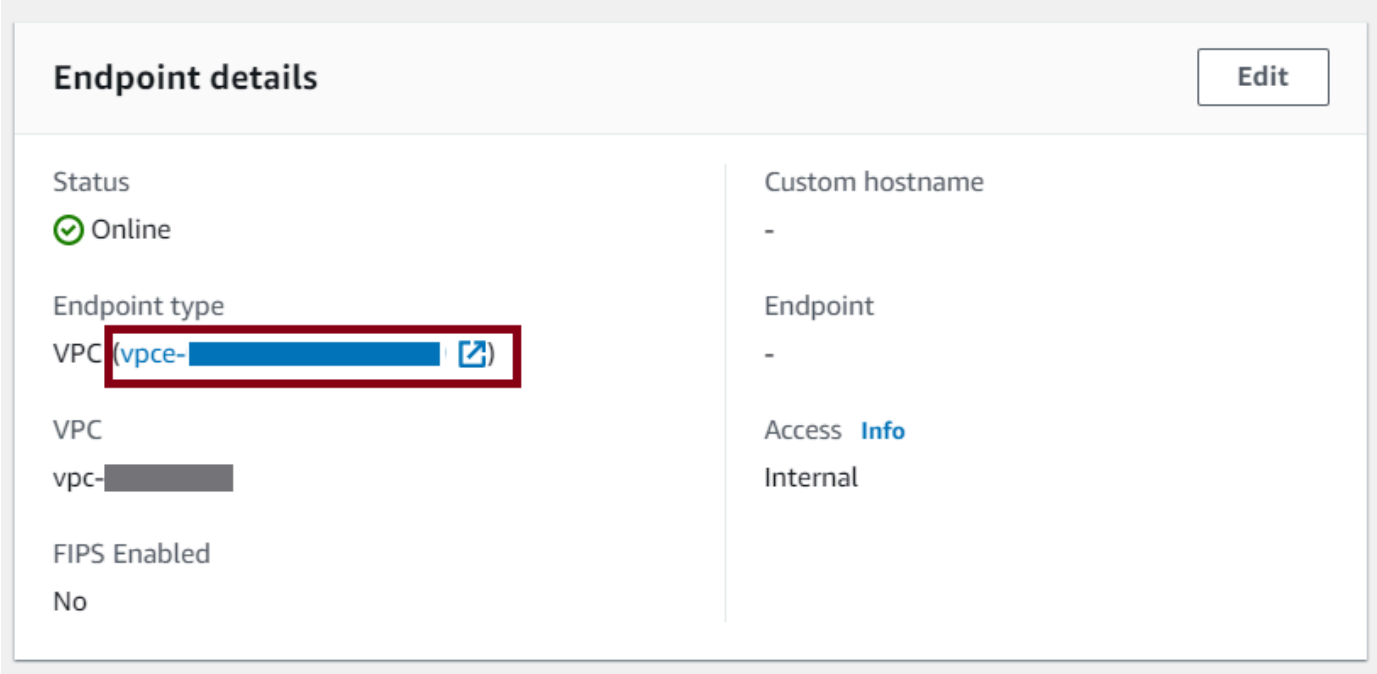
Command	Amazon S3	Amazon EFS
		Note 기존 파일 또는 디렉터리를 덮어쓰는 명칭 변경은 지원되지 않습니다.
`rm`	지원	지원
`rmdir`	지원(빈 디렉터리만 해당)	지원
`version`	지원	지원

Amazon VPC 엔드포인트 찾기

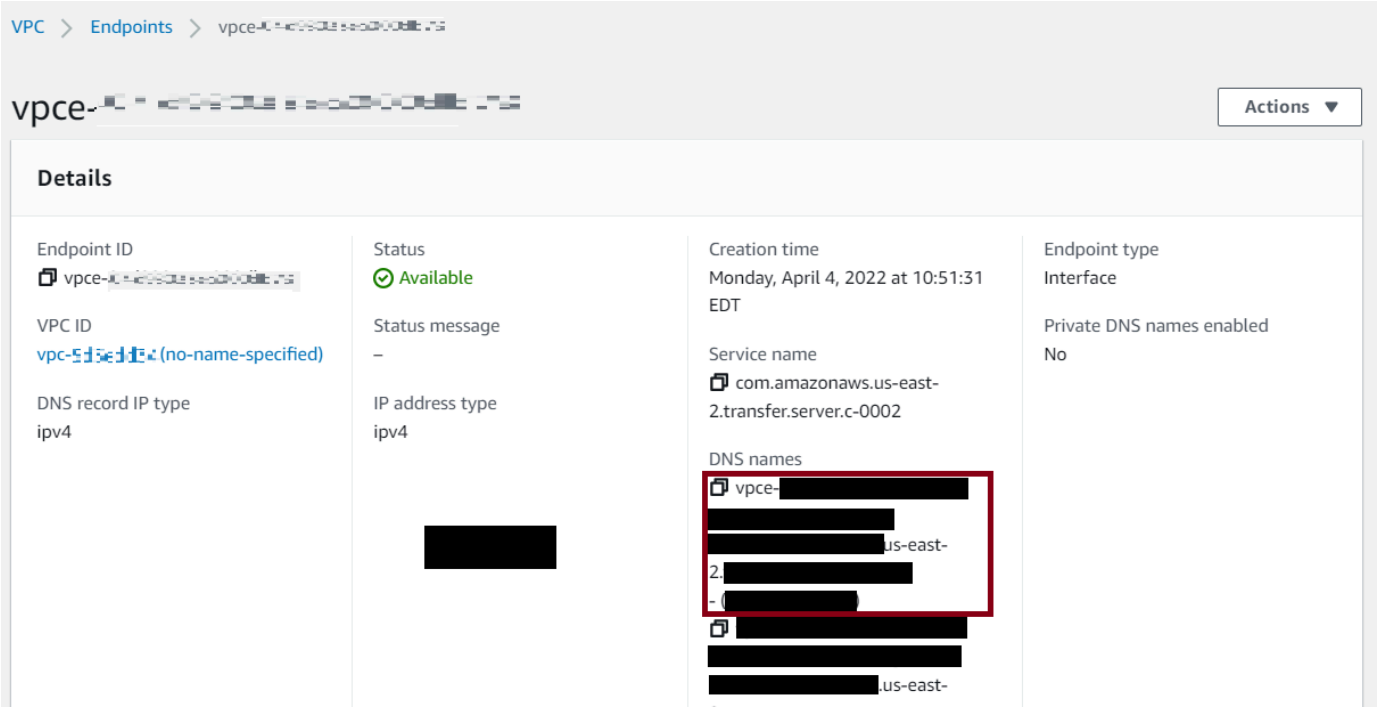
Transfer Family 서버의 엔드포인트 타입이 VPC인 경우, 파일 전송에 사용할 엔드포인트를 식별하는 것은 간단하지 않습니다. 이 경우, 다음 절차에 따라 Amazon VPC 엔드포인트를 찾습니다.

Amazon VPC 엔드포인트 찾기

1. 서버의 세부 정보 페이지로 이동합니다.
2. 엔드포인트 세부 정보 창에서 VPC를 선택합니다.



3. Amazon VPC 대시보드에서 VPC 엔드포인트 ID를 선택합니다.
4. DNS 명칭 목록에서 서버 엔드포인트가 첫 번째로 나열됩니다.



setstat 오류 방지

일부 SFTP 파일 전송 클라이언트는 파일을 업로드할 때 SETSTAT 등의 명령을 사용하여 타임스탬프 및 권한 등 원격 파일의 속성을 변경하려고 시도할 수 있습니다. 그러나 이러한 명령은 Amazon S3와 같은 객체 스토리지 시스템과 호환되지 않습니다. 이러한 비호환성으로 인해 이러한 클라이언트에서 파일을 업로드하면 파일이 성공적으로 업로드된 경우에도 오류가 발생할 수 있습니다.

- `CreateServer` 또는 `UpdateServer` API를 호출할 때 클라이언트가 S3 버킷에 업로드하는 파일에 SETSTAT를 사용하려고 할 때 생성되는 오류를 무시하는 `ProtocolDetails` 옵션 `SetStatOption`를 사용합니다.
- Transfer Family 서버가 SETSTAT 명령을 무시하고 SFTP 클라이언트를 변경할 필요 없이 파일을 업로드하려면 값을 `ENABLE_NO_OP`로 설정합니다.
- 이 *`SetStatOption ENABLE_NO_OP`* 설정은 오류를 무시하지만 CloudWatch Logs에 로그 항목을 생성하므로 클라이언트가 언제 SETSTAT를 호출하는지 확인할 수 있습니다.

이 옵션의 API 세부 정보는 을 참조하십시오. ProtocolDetails

OpenSSH 사용

아래의 지침을 이용해 OpenSSH를 이용하는 명령줄로 파일을 전송하세요.

> **Note**
>
> 이 클라이언트는 SFTP 지원 서버에서만 작동합니다.

OpenSSH 명령줄 유틸리티를 AWS Transfer Family 사용하여 파일을 전송하려면

1. Linux, macOS 또는 Windows에서 명령 터미널을 엽니다.
2. 프롬프트에서 다음 명령을 입력합니다.

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

   앞의 명령에서 *`sftp_user`*는 사용자 이름이고 *`transfer-key`*는 SSH 프라이빗 키입니다. 다음은 선택한 서버의 AWS Transfer Family 콘솔에 표시된 서버 엔드포인트입니다. *`service_endpoint`*

> **Note**
>
> 이 명령은 기본 `ssh_config` 파일에 있는 설정을 사용합니다. 이전에 이 파일을 편집하지 않은 경우 SFTP는 포트 22를 사용합니다. 다음과 같이 명령에 **-P** 플래그를 추가하여 다른 포트 (예: 2222) 를 지정할 수 있습니다.
>
> ```
> sftp -P 2222 -i transfer-key sftp_user@service_endpoint
> ```
>
> 또는 항상 포트 2222 또는 포트 22000을 사용하려는 경우 파일의 기본 포트를 업데이트할 수 있습니다. `ssh_config`

`sftp` 프롬프트가 나타날 것입니다.

3. (옵션) 사용자의 홈 디렉터리를 보려면 `sftp` 프롬프트에 다음 명령을 입력합니다.

   ```
   pwd
   ```

4. 파일 시스템에서 Transfer Family 서버로 파일을 업로드하려면 `put` 명령을 사용합니다. 예를 들어 업로드하려면 `hello.txt` (파일이 파일 시스템의 현재 디렉터리에 있다고 간주) `sftp` 프롬프트에서 다음 명령을 실행합니다.

   ```
   put hello.txt
   ```

   파일 전송이 진행 중이거나 완료되었음을 나타내는, 다음과 비슷한 메시지가 표시됩니다.

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

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

> **Note**
>
> 서버가 생성된 후, 환경에서 DNS 서비스에 대한 서버 엔드포인트 호스트 명칭을 해석하려면 몇 분 정도 걸릴 수 있습니다.

WinSCP 사용

아래의 지침을 이용해 WinSCP를 이용하는 명령줄로 파일을 전송하세요.

> **Note**
>
> WinSCP 5.19를 사용하는 경우 자격 증명을 사용하여 Amazon S3에 직접 연결하고 파일을 업로드/다운로드할 수 AWS 있습니다. 자세한 내용은 Amazon S3 서비스에 연결을 참조하세요.

AWS Transfer Family WinSCP를 사용하여 파일을 전송하려면

1. WinSCP 클라이언트를 엽니다.
2. 로그인 대화 상자의 파일 프로토콜에서 프로토콜을 선택합니다. SFTP 또는 FTP.

   FTP를 선택한 경우, 암호화에서 다음 중 하나를 선택합니다.

   - FTP에는 암호화가 적용되지 않습니다
   - FTPS를 위한 TLS/SSL 명시적 암호화
3. 호스트 명칭에 서버 엔드포인트를 입력합니다. 서버 엔드포인트는 서버 세부 정보 페이지에 있습니다. 자세한 내용은 SFTP, FTPS 및 FTP 서버 세부 정보 보기를 참조하세요.

   > **Note**
   >
   > 서버가 VPC 엔드포인트를 사용하는 경우, Amazon VPC 엔드포인트 찾기를 참조하세요.
4. 포트 번호에 다음을 입력합니다:

   - SFTP에 대한 **22**
   - FTP/FTPS에 대한 **21**
5. 사용자 이름에는 특정 ID 공급자용으로 생성한 사용자 이름을 입력합니다.

   > **Note**
   >
   > 사용자 이름은 ID 공급자용으로 만들거나 구성한 사용자 중 한 명이어야 합니다. AWS Transfer Family 는 다음과 같은 ID 제공자를 제공합니다.
   >
   > - 서비스 관리형과 작업
   > - AWS 디렉터리 서비스 ID 제공자 사용
   > - 사용자 지정 자격 증명 공급자와 작업
6. 고급을 선택하여 고급 사이트 설정 대화 상자를 엽니다. SSH 섹션에서 인증을 선택합니다.

7. 프라이빗 키 파일의 경우, 파일 시스템에서 SSH 프라이빗 키 파일을 찾아 선택합니다.

> **Note**
>
> WinSCP가 SSH 프라이빗 키를 PPK 형식으로 변환하는 기능을 제공한다면, OK(확인)을 선택합니다.

8. OK(확인)을 선택해 Login(로그인) 대화 상자로 돌아간 다음 Save(저장)을 선택합니다.
9. 다음과 같은 Save session as site(세션을 사이트로 저장) 대화 상자에서 OK(확인)을 선택해 연결 설정을 완료합니다.
10. 로그인 대화 상자에서 도구를 선택한 다음 기본 설정을 선택합니다.
11. 환경 설정 대화 상자의 전송에서 Endurance를 선택합니다.

    임시 파일 명칭으로 전송 재개/전송 활성화 옵션에 대해 비활성화를 선택합니다.

> **Note**
>
> 이 옵션을 활성화한 상태로 두면 업로드 비용이 증가하여 업로드 성능이 크게 저하됩니다. 또한 대용량 파일 업로드가 실패할 수도 있습니다.

12. 전송의 경우, 배경을 선택하고 단일 전송에 복수 연결 사용 확인란의 선택을 취소합니다.

> **Note**
>
> 이 옵션을 선택한 상태로 두면 대용량 파일 업로드가 예상치 못한 방식으로 실패할 수 있습니다. 예를 들어 Amazon S3 요금이 발생하는 분리된 멀티파트 업로드를 생성할 수 있습니다. 조용한 데이터 손상도 발생할 수 있습니다.

13. 파일 전송을 실행합니다.

    drag-and-drop 메서드를 사용하여 대상 창과 소스 창 간에 파일을 복사할 수 있습니다. 도구 모음 아이콘을 이용해 WinSCP의 파일을 업로드, 다운로드, 삭제, 편집하거나 속성을 수정할 수 있습니다.

> **Note**
>
> Amazon EFS를 스토리지로 사용하는 경우에는 이 참고 사항이 적용되지 않습니다.

타임스탬프를 포함하여 원격 파일의 속성을 변경하려는 명령은 Amazon S3와 같은 객체 스토리지 시스템과 병립되지 않습니다. 따라서 Amazon S3를 스토리지로 사용하는 경우, 파일 전송을 수행하기 전에 WinSCP 타임스탬프 설정을 비활성화해야 합니다(또는 `setstat` 오류 방지에 설명된 대로 `SetStatOption` 사용). 이렇게 하려면, WinSCP Transfer 설정 대화 상자에서 권한 설정 업로드 옵션과 타임스탬프 유지 일반 옵션을 비활성화해야 합니다.

Cyberduck 사용하기

아래의 지침을 이용해 Cyberduck을 이용하는 명령줄로 파일을 전송하세요.

Cyberduck을 사용하여 파일을 AWS Transfer Family 전송하려면

1. Cyberduck 클라이언트를 엽니다.
2. 연결 열기를 선택합니다.
3. 연결 열기 대화 상자에서 프로토콜을 선택합니다. SFTP (SSH 파일 전송 프로토콜), FTP-SSL (명시적 AUTH TLS), 또는 FTP (파일 전송 프로토콜).
4. 서버에 서버 엔드포인트를 입력합니다. 서버 엔드포인트는 서버 세부 정보 페이지에 있습니다. 자세한 내용은 SFTP, FTPS 및 FTP 서버 세부 정보 보기를 참조하세요.

   > **Note**
   >
   > 서버가 VPC 엔드포인트를 사용하는 경우, Amazon VPC 엔드포인트 찾기를 참조하세요.

5. 포트 번호에 다음을 입력합니다:
   - SFTP에 대한 **22**
   - FTP/FTPS에 대한 **21**
6. 사용자 이름에 서버 엔드포인트의 사용자 관리에서 생성한 사용자 이름을 입력합니다.
7. SFTP를 선택한 경우, SSH 프라이빗 키에 대해 SSH 프라이빗 키를 선택하거나 입력합니다.
8. 연결을 선택합니다.
9. 파일 전송을 실행합니다.

   파일 위치에 따라, 다음 중 하나를 수행하세요.

   - 로컬 디렉터리(소스)에서 전송할 파일을 선택하고, Amazon S3 디렉터리(대상)로 끌어다 놓습니다.

- Amazon S3 디렉터리(소스)에서 전송할 파일을 선택하고, 로컬 디렉터리(대상)로 끌어다 놓습니다.

사용 FileZilla

를 사용하여 파일을 전송하려면 다음 지침을 따르십시오 FileZilla.

파일 FileZilla 전송을 설정하려면

1. FileZilla 클라이언트를 엽니다.
2. 파일을 선택한 다음 사이트 관리자를 선택합니다.
3. 사이트 관리자 대화 상자에서 새 사이트를 선택합니다.
4. 일반 탭에서 프로토콜을 위해 프로토콜을 선택합니다.SFTP 또는 FTP.

   FTP를 선택한 경우, 암호화에서 다음 중 하나를 선택합니다.

   - 일반 FTP(비보안)만 사용 – FTP의 경우
   - 가능한 경우, TLS를 통한 명시적 FTP를 사용 – FTPS의 경우
5. 호스트 명칭에는 사용 중인 프로토콜을 입력한 다음 서버 엔드포인트를 입력합니다. 서버 엔드포인트는 서버 세부 정보 페이지에 있습니다. 자세한 내용은 SFTP, FTPS 및 FTP 서버 세부 정보 보기를 참조하세요.

   > **Note**
   >
   > 서버가 VPC 엔드포인트를 사용하는 경우, Amazon VPC 엔드포인트 찾기를 참조하세요.

   - SFTP를 사용하는 경우, 다음을 입력합니다: `sftp://`*`hostname`*
   - FTPS를 사용하는 경우, 다음을 입력합니다: `ftps://`*`hostname`*

   `###` *`##`*을 실제 서버 엔드포인트로 바꿔야 합니다.
6. 포트 번호에 다음을 입력합니다:

   - SFTP에 대한 **22**
   - FTP/FTPS에 대한 **21**
7. SFTP를 선택한 경우, 로그온 타입으로 키 파일을 선택합니다.

키 파일의 경우, SSH 프라이빗 키를 선택하거나 입력합니다.

8. 사용자 이름에 서버 엔드포인트의 사용자 관리에서 생성한 사용자 이름을 입력합니다.
9. 연결을 선택합니다.
10. 파일 전송을 실행합니다.

> **Note**
>
> 진행 중인 파일 전송을 중단하는 경우 Amazon S3 버킷에 부분 객체를 작성할 AWS Transfer Family 수 있습니다. 업로드를 중단했다면, Amazon S3 버킷의 파일 크기가 소스 개체의 파일 크기와 동일한지 확인한 후 다음 단계로 계속합니다.

Perl 클라이언트 사용

NET::SFTP::Foreignperl 클라이언트를 사용하는 경우 를 로 설정해야 합니다. `queue_size` `1` 예:

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

> **Note**
>
> 이 해결 방법은 1.92.02 이전의 `Net::SFTP::Foreign`을 위해 필요합니다.

업로드 후 처리

Amazon S3 객체 메타데이터 및 이벤트 알림을 비롯한 업로드 후 처리 정보를 볼 수 있습니다.

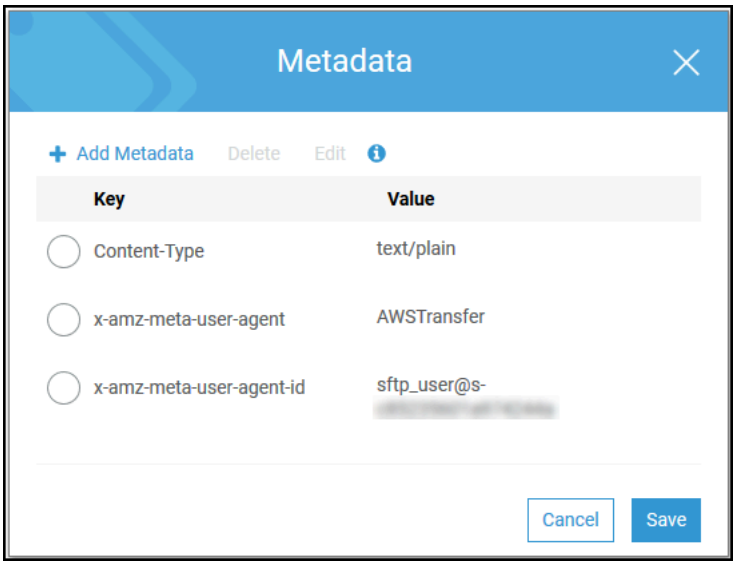
주제

- Amazon S3 객체 메타데이터
- Amazon S3 이벤트 알림

Amazon S3 객체 메타데이터

객체 메타데이터의 일부로, `'x-amz-meta-user-agent`의 값은 `AWSTransfer`(이)고 `x-amz-meta-user-agent-id`의 값은 `username@server-id'`(이)라는 키가 표시됩니다. `username`은(는) 파일을

업로드한 Transfer Family 사용자이며, `server-id`은(는) 업로드에 사용되는 서버입니다. Lambda 함수 내의 S3 객체에 대한 HeadObject작업을 사용하여 이 정보에 액세스할 수 있습니다.



Amazon S3 이벤트 알림

Transfer Family를 사용하여 S3 버킷에 객체를 업로드하면 `RoleSessionName`이(가) S3 이벤트 알림 구조의 요청자 필드에 `[AWS:Role Unique Identifier]/username.sessionid@server-id`와(과) 같이 포함됩니다. 예를 들어, 다음은 S3 버킷으로 복사된 파일에 대한 S3 액세스 로그의 샘플 요청자 필드 콘텐츠입니다.

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

위의 요청자 필드에는 `IamRoleName`(이)라고 불리는 IAM 역할이 표시됩니다. S3 이벤트 알림 구성에 대한 자세한 설명은 Amazon Simple Storage Service 개발자 가이드의 Configuring Amazon S3 이벤트 알림을 참조하세요. AWS Identity and Access Management (IAM) 역할 고유 식별자에 대한 자세한 내용은 사용 설명서의 고유 식별자를 참조하십시오.AWS Identity and Access Management

서버 엔드포인트의 사용자 관리

다음 섹션에서는 AWS Transfer Family, AWS Directory Service for Microsoft Active Directory 또는 맞춤 자격 증명 공급자를 사용하여 사용자를 추가하는 방법에 대한 정보를 찾아볼 수 있습니다.

서비스 관리형 자격 증명 타입을 사용한다면, 사용자를 파일 전송 프로토콜 지원 서버에 추가해야 합니다. 이 작업을 할 때 각 사용자 이름은 서버 내에서 고유해야 합니다.

또한 각 사용자 속성의 일부로, 사용자의 SSH(보안 쉘) 퍼블릭 키를 저장합니다. 이는 이 절차에서 사용하는 키 기반 인증에 필요합니다. 프라이빗 키는 사용자의 컴퓨터에 로컬로 저장됩니다. 사용자가 클라이언트를 사용하여 서버에 인증 요청을 보내면, 서버에서는 먼저 사용자를 사용자가 연관된 SSH 프라이빗 키에 대한 액세스 권한을 가지고 있는지 확인합니다. 그러면 서버가 사용자를 성공적으로 인증합니다.

또한 사용자의 홈 디렉터리 또는 랜딩 디렉터리를 지정하고, AWS Identity and Access Management(IAM) 역할을 사용자에게 할당합니다. 원한다면 세션 정책을 제공해 사용자 액세스를 Amazon S3 버킷의 홈 디렉터리로 제한할 수도 있습니다.

> **Important**
>
> AWS Transfer Family에서는 1~2자 길이의 사용자 이름이 SFTP 서버에 인증되지 않도록 차단합니다. 또한 `root` 사용자 이름도 차단합니다.
> 그 이유는 암호 스캐너의 악의적인 로그인 시도가 대량으로 발생하기 때문입니다.

아마존 EFS와 아마존 S3

각 스토리지 옵션의 특징:

- 액세스를 제한하려면: Amazon S3는 세션 정책을 지원하고, Amazon EFS는 POSIX 사용자, 그룹 및 보조 그룹 ID를 지원합니다.
- 둘 다 퍼블릭/프라이빗 키 지원
- 둘 다 홈 디렉터리 지원
- 둘 다 논리적 디렉터리 지원

> **Note**
>
> Amazon S3의 경우 논리적 디렉터리에 대한 대부분의 지원은 API/CLI를 통해 이루어집니다. 콘솔의 제약 확인란을 사용하여 사용자를 홈 디렉터리에 잠글 수 있지만 가상 디렉터리 구조를 지정할 수는 없습니다.

논리적 디렉터리

사용자의 논리적 디렉터리 값을 지정하는 경우 사용하는 파라미터는 사용자 타입에 따라 달라집니다.

- 서비스 관리 사용자의 경우 `HomeDirectoryMappings`에서 논리적 디렉터리 값을 제공하세요.

- 사용자 지정 ID 제공자 사용자의 경우 에서 `HomeDirectoryDetails` 논리적 디렉터리 값을 제공하십시오.

주제

- 서비스 관리형과 작업
- AWS 디렉터리 서비스 ID 제공자 사용
- 사용자 지정 자격 증명 공급자와 작업

서비스 관리형과 작업

서버의 도메인 설정에 따라 Amazon S3 또는 Amazon EFS 서비스 관리 사용자를 서버에 추가할 수 있습니다. 자세한 설명은 SFTP, FTPS 또는 FTP 서버 엔드포인트 구성 섹션을 참조하세요.

프로그래밍 방식으로 서비스 관리 사용자를 추가하려면 API 예제를 참조하십시오. CreateUser

> **Note**
>
> 서비스 관리 사용자의 경우 논리적 디렉터리 항목은 2,000개로 제한됩니다. 논리적 디렉터리 사용에 대한 자세한 내용은 을 참조하십시오. 논리적 디렉터리를 사용하여 Transfer Family 디렉터리 구조를 단순화합니다.

주제

- Amazon S3 서비스 관리 사용자 추가
- Amazon EFS 서비스 관리 사용자 추가
- 서비스 관리 사용자 관리

Amazon S3 서비스 관리 사용자 추가

> **Note**
>
> 교차 계정 Amazon S3 버킷을 구성하려면 이 지식 센터 문서의 다른 AWS 계정에 있는 Amazon Simple Storage Service 버킷을 사용하도록 AWS Transfer Family 서버를 구성하려면 어떻게 해야 하는가?에 설명된 단계를 따르세요.

Amazon S3 서비스 관리 사용자를 서버에 추가하려면

1. https://console.aws.amazon.com/transfer/에서 AWS Transfer Family 콘솔을 연 다음 탐색 창에서 서버를 선택합니다.
2. 서버 페이지에서 사용자를 추가할 서버의 확인란을 선택합니다.
3. 사용자 추가를 선택합니다.
4. 사용자 구성 섹션의 사용자 이름에 사용자 이름을 입력합니다. 이 사용자 이름은 3~32자여야 합니다. 사용자 이름에는 다음 글자를 사용할 수 있습니다. a~z, A~Z, 0~9, 밑줄 '_', 하이픈 '-', 마침표 '.'과 골뱅이 사인 "@"입니다. 사용자 이름은 하이픈 '-', 마침표 '.' 및 "@" 기호로 시작할 수 없습니다.
5. 액세스에는 Amazon S3 버킷에 대한 액세스를 제공하는, 이전에 생성한 IAM 역할을 선택합니다.

   이 IAM 역할은 IAM 역할 및 정책 생성의 절차를 사용하여 생성했습니다. 해당 IAM 역할은 Amazon S3 버킷에 대한 액세스를 제공하는 IAM 정책을 포함합니다. 또한 다른 IAM 정책에서 정의하는, AWS Transfer Family 서비스와의 신뢰 관계도 제공합니다. 사용자에 대한 세밀한 액세스 통제가 필요한 경우 AWS Transfer Family 및 Amazon S3를 통한 데이터 액세스 통제 강화 블로그 게시물을 참조하세요.
6. (옵션) 정책에서 다음 중 하나를 선택합니다:

   - None(없음)
   - 기존 정책
   - IAM에서 정책 선택: 기존 세션 정책을 선택할 수 있습니다. 정책의 세부 정보가 포함된 JSON 객체를 보려면 보기를 선택합니다.
   - 홈 폴더 기반 정책 자동 생성: 세션 정책을 자동으로 생성합니다. 정책의 세부 정보가 포함된 JSON 객체를 보려면 보기를 선택합니다.

   > **Note**
   >
   > 홈 폴더를 기반으로 정책 자동 생성을 선택하는 경우 이 사용자에 대해 제약을 선택하지 마십시오.

   세션 정책에 대한 자세한 설명은 IAM 역할 및 정책 생성 섹션을 참조하세요. 세션 정책 생성에 대한 자세한 설명은 Amazon S3 버킷을 위한 세션 정책 생성 섹션을 참조하세요.

7. 홈 디렉터리에는 AWS Transfer Family를 이용해 전송할 데이터를 저장할 Amazon S3 버킷을 선택합니다. 사용자가 클라이언트를 이용해 로그인하면 도착하게 되는 home 디렉터리의 경로를 입력합니다.

   이 파라미터를 입력하지 않으면 Amazon 버킷의 `root` 디렉터리를 사용합니다. 이 경우 IAM 역할이 이 `root` 디렉터리에 대한 액세스를 제공하는지 확인하십시오.

   > **Note**
   >
   > 사용자의 사용자 이름을 포함하는 디렉터리 경로 선택을 권장합니다. 세션 정책을 효과적으로 사용할 수 있습니다. 세션 정책은 사용자의 home 디렉터리에 대한 Amazon S3 버킷에서의 사용자 액세스를 제한합니다.

8. (옵션) 제약의 경우, 사용자가 해당 폴더 외부에 액세스할 수 없고 Amazon S3 버킷 또는 폴더 이름을 볼 수 없도록 확인란을 선택합니다.

   > **Note**
   >
   > 사용자에게 홈 디렉터리를 할당하고 사용자를 해당 홈 디렉터리로 제한하면 지정된 폴더에 대한 사용자 액세스를 차단하기에 충분해야 합니다. 추가 통제를 적용해야 하는 경우 세션 정책을 사용하세요.
   > 이 사용자에 대해 제약을 선택하면 홈 폴더가 제한된 사용자에 대해 정의된 값이 아니기 때문에 홈 폴더 기반 정책 자동 생성을 선택할 수 없습니다.

9. SSH 퍼블릭 키에는 SSH 키 쌍의 SSH 퍼블릭 키 부분을 입력합니다.

   서비스에서 사용자의 키를 확인해야 새 사용자를 추가할 수 있습니다.

   > **Note**
   >
   > SSH 키 쌍을 만드는 방법에 대한 지침은 서비스 관리 사용자를 위한 SSH 키 생성 섹션을 참조하세요.

10. (선택 사항) 키와 값에 하나 이상의 태그를 키-값 페어로 입력하고 태그 추가를 선택합니다.
11. 추가를 선택해 새 사용자를 원하는 서버에 추가합니다.

    새 사용자는 서버 세부 정보 페이지의 사용자 섹션에 나타납니다.

다음 단계 - 그 다음 단계에서는 클라이언트를 사용하여 서버 엔드포인트를 통한 파일 전송로 계속 진행하세요.

Amazon EFS 서비스 관리 사용자 추가

Amazon EFS는 Portable Operating System Interface(POSIX) 파일 권한 모델을 사용하여 파일 소유권을 나타냅니다.

- Amazon EFS 파일 소유권에 대한 자세한 설명은 Amazon EFS 파일 소유권을 참조하세요.
- EFS 사용자를 위한 디렉터리 설정에 대한 자세한 설명은 Transfer Family를 위한 Amazon EFS 사용자 설정 섹션을 참조하세요.

Amazon EFS 서비스 관리 사용자를 서버에 추가하려면

1. https://console.aws.amazon.com/transfer/에서 AWS Transfer Family 콘솔을 연 다음 탐색 창에서 서버를 선택합니다.
2. 서버 페이지에서 사용자를 추가하려는 Amazon EFS 서버를 선택합니다.
3. 사용자 추가를 선택하여 사용자 추가 페이지를 표시합니다.
4. 사용자 구성 섹션에서 다음 설정을 구성합니다.

    a. 이 사용자 이름은 최소 3글자에서 최대 100자여야 합니다. 사용자 이름에는 다음 글자들을 사용할 수 있습니다: a~z, A~Z, 0~9, 밑줄 '_', 하이픈 '-', 마침표 '.'과 골뱅이 부호 "@". 사용자 이름은 하이픈 '-', 마침표 '.' 및 "@" 기호로 시작할 수 없습니다.

    b. 사용자 ID 및 그룹 ID의 경우, 다음을 참고하세요:

    - 처음 생성하는 사용자의 경우, 그룹 ID와 사용자 ID 모두에 대한 **0**의 값을 입력하는 것이 좋습니다. 이렇게 하면 Amazon EFS에 대한 사용자 관리자 권한이 부여됩니다.
    - 추가 사용자의 경우, 사용자의 POSIX 사용자 ID 및 그룹 ID를 입력합니다. 이러한 ID는 사용자가 수행하는 모든 Amazon Elastic File System 작업에 사용됩니다.
    - 사용자 ID 및 그룹 ID의 경우 앞에 0을 사용하지 마십시오. 예컨대, **12345**는 허용되지만 **012345**는 허용되지 않습니다.

    c. (옵션) 보조 그룹 ID의 경우 각 사용자에 대해 하나 이상의 추가 POSIX 그룹 ID를 쉼표로 구분하여 입력합니다.

    d. 액세스에서 다음과 같은 IAM 역할을 선택합니다:

    - 액세스하려는 Amazon EFS 리소스(파일 시스템)에 대한 액세스 권한만 사용자에게 부여합니다.

- 사용자가 수행할 수 있는 파일 시스템 작업과 수행할 수 없는 파일 시스템 작업을 정의합니다.

  마운트 액세스 및 읽기/쓰기 권한이 있는 Amazon EFS 파일 시스템 선택에는 IAM 역할을 사용하는 것이 좋습니다. 예컨대, 다음 두 AWS 관리형 정책을 조합하면 상당히 관대하긴 하지만 사용자에게 필요한 권한을 부여할 수 있습니다.

  - AmazonElasticFileSystemClientFullAccess
  - AWSTransferConsoleFullAccess

  자세한 설명은 블로그 게시물 Amazon Elastic File System을 위한 AWS Transfer Family 지원을 참조하세요.

e. 홈 디렉터리의 경우, 다음을 수행하세요:

- AWS Transfer Family를 사용하여 전송할 데이터를 저장하는 데 사용할 Amazon EFS 파일 시스템을 선택합니다.
- 홈 디렉터리를 제약으로 설정할지 여부를 결정하세요. 홈 디렉터리를 제약으로 설정하면 다음과 같은 결과가 발생합니다:
  - Amazon EFS 사용자는 해당 폴더 외부의 파일이나 디렉터리에 액세스할 수 없습니다.
  - 아마존 EFS 사용자는 아마존 EFS 파일 시스템 이름(fs-xxxxxxx)을 볼 수 없습니다.

  > **Note**
  >
  > 제약 옵션을 선택하면 Amazon EFS 사용자의 경우 심볼릭 링크가 확인되지 않습니다.

- (옵션) 사용자가 클라이언트를 사용하여 로그인할 때 들어갈 홈 디렉터리의 경로를 입력합니다.

  홈 디렉터리를 지정하지 않는 경우 Amazon EFS 파일 시스템의 루트 디렉터리가 사용됩니다. 이 경우 IAM 역할이 이 루트 디렉터리에 대한 액세스를 제공하는지 확인하세요.

5. SSH 퍼블릭 키에는 SSH 키 쌍의 SSH 퍼블릭 키 부분을 입력합니다.

   서비스에서 사용자의 키를 확인해야 새 사용자를 추가할 수 있습니다.

> **Note**
>
> SSH 키 쌍을 만드는 방법에 대한 지침은 서비스 관리 사용자를 위한 SSH 키 생성 섹션을 참조하세요.

6. (옵션) 사용자의 태그를 입력하세요. 키 및 값에 하나 이상의 태그를 키-값 쌍로 입력하고 태그 추가를 선택합니다.
7. 추가를 선택해 새 사용자를 원하는 서버에 추가합니다.

   새 사용자는 서버 세부 정보 페이지의 사용자 섹션에 나타납니다.

Transfer Family 서버에 처음 SFTP를 연결할 때 발생할 수 있는 문제는 다음과 같습니다:

- `sftp` 명령을 실행해도 프롬프트가 나타나지 않으면 다음 메시지가 표시될 수 있습니다.

  ```
  Couldn't canonicalize: Permission denied
  ```

  ```
  Need cwd
  ```

  이 경우 사용자 역할에 대한 정책 권한을 늘려야 합니다. `AmazonElasticFileSystemClientFullAccess`와(과) 같은 AWS 관리형 정책을 추가할 수 있습니다
- 사용자의 홈 디렉터리를 보라는 `sftp` 프롬프트에 `pwd`를 입력하면 다음 메시지가 표시될 수 있습니다. 여기서 *`USER-HOME-DIRECTORY`*는 SFTP 사용자의 홈 디렉터리입니다.

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

  이 경우 상위 디렉터리(`cd ..`)로 이동하여 사용자의 홈 디렉터리(`mkdir` *`username`*)를 만들 수 있어야 합니다.

다음 단계 - 그 다음 단계에서는 클라이언트를 사용하여 서버 엔드포인트를 통한 파일 전송로 계속 진행하세요.

서비스 관리 사용자 관리

이 섹션에서는 사용자 목록을 보는 방법, 사용자 세부 정보를 편집하는 방법, SSH 퍼블릭 키를 추가하는 방법에 대한 정보를 찾을 수 있습니다.

- 사용자 목록 보기
- 사용자 세부 정보 보기 또는 편집
- 사용자 삭제
- SSH 공개 키 추가
- SSH 퍼블릭 키 삭제

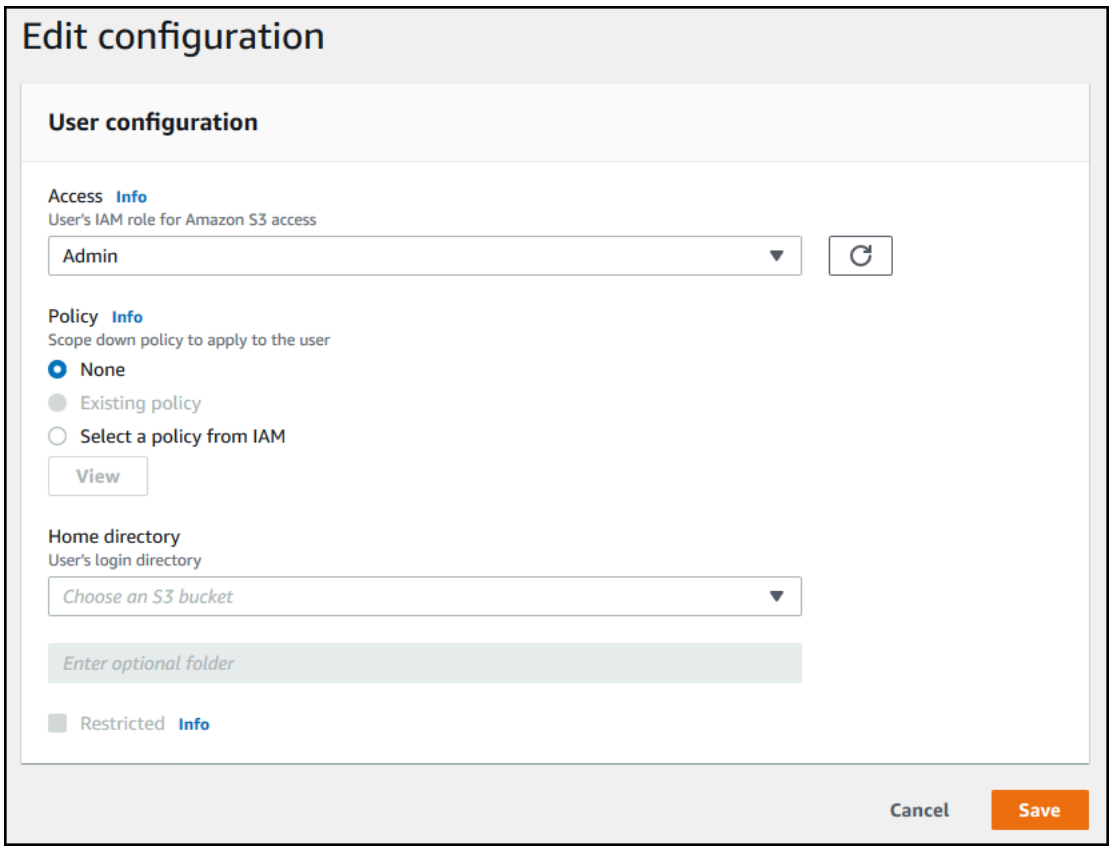
사용자 목록을 찾으려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 탐색 창에서 서버를 선택하여 서버 페이지를 표시합니다.
3. 서버 ID 열에서 식별자를 선택하여 서버 세부 정보 페이지를 표시합니다.
4. 사용자에서 사용자 목록을 확인합니다.

사용자 세부 정보를 보거나 편집하려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 탐색 창에서 서버를 선택하여 서버 페이지를 표시합니다.
3. 서버 ID 열에서 식별자를 선택하여 서버 세부 정보 페이지를 표시합니다.
4. 사용자에서 사용자 이름을 선택하면 사용자 세부 정보 페이지가 표시됩니다.

   편집을 선택하여 이 페이지에서 사용자의 속성을 변경할 수 있습니다.
5. 사용자 세부 정보 페이지에서 사용자 구성 옆의 편집을 선택합니다.



6. 구성 편집 페이지의 Access에서 Amazon S3 버킷에 대한 액세스를 제공하는 이전에 생성한 IAM 역할을 선택합니다.

   이 IAM 역할은 IAM 역할 및 정책 생성의 절차를 사용하여 생성했습니다. 해당 IAM 역할은 Amazon S3 버킷에 대한 액세스를 제공하는 IAM 정책을 포함합니다. 또한 다른 IAM 정책에서 정의하는, AWS Transfer Family 서비스와의 신뢰 관계도 제공합니다.

7. (옵션) 정책에서 다음 중 하나를 선택합니다.

   - None(없음)
   - 기존 정책
   - IAM에서 정책을 선택하여 기존 정책을 선택합니다. 정책의 세부 정보가 포함된 JSON 객체를 보려면 보기를 선택합니다.

   세션 정책에 대한 자세한 설명은 IAM 역할 및 정책 생성 섹션을 참조하세요. 세션 정책 생성에 대한 자세한 설명은 Amazon S3 버킷을 위한 세션 정책 생성 섹션을 참조하세요.

8. 홈 디렉터리에는 AWS Transfer Family를 이용해 전송할 데이터를 저장할 Amazon S3 버킷을 선택합니다. 사용자가 클라이언트를 사용하여 로그인하면 도착하게 되는 home 디렉터리의 경로를 입력합니다.

이 파라미터를 공란으로 두면 Amazon S3 버킷의 `root` 디렉터리가 사용됩니다. 이 경우 IAM 역할이 이 `root` 디렉터리에 대한 액세스를 제공하는지 확인하십시오.

> **Note**
>
> 사용자의 사용자 이름을 포함하는 디렉터리 경로 선택을 권장합니다. 세션 정책을 효과적으로 사용할 수 있습니다. 세션 정책은 사용자의 home 디렉터리에 대한 Amazon S3 버킷에서의 사용자 액세스를 제한합니다.

9. (옵션) 제약의 경우, 사용자가 해당 폴더 외부에 액세스할 수 없고 Amazon S3 버킷 또는 폴더 이름을 볼 수 없도록 확인란을 선택합니다.

> **Note**
>
> 사용자에게 홈 디렉터리를 할당하고 해당 홈 디렉터리로 사용자를 제한하는 경우 이 정도면 지정된 폴더에 대한 사용자 액세스를 잠그기에 충분합니다. 추가 통제를 적용해야 하는 경우 세션 정책을 사용하세요.

10. 저장을 선택하여 변경 사항을 저장합니다.

사용자 삭제

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 탐색 창에서 서버를 선택하여 서버 페이지를 표시합니다.
3. 서버 ID 열에서 식별자를 선택하여 서버 세부 정보 페이지를 표시합니다.
4. 사용자에서 사용자 이름을 선택하면 사용자 세부 정보 페이지가 표시됩니다.
5. 사용자 세부 정보 페이지에서 사용자 이름 오른쪽에 있는 삭제를 선택합니다.
6. 표시되는 확인 대화 상자에서 단어 **`delete`**을 입력한 다음 삭제를 선택하여 사용자 삭제를 확인합니다.

사용자 목록에서 사용자가 삭제됩니다.

사용자를 위한 SSH 공개 키를 추가하려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 탐색 창에서 서버를 선택합니다.

3. 서버 ID 열에서 식별자를 선택하여 서버 세부 정보 페이지를 표시합니다.
4. 사용자에서 사용자 이름을 선택하면 사용자 세부 정보 페이지가 표시됩니다.
5. Add SSH public key(SSH 퍼블릭 키 추가)를 선택해 새 SSH 퍼블릭 키를 사용자에 추가합니다.

> **Note**
>
> SSH 키는 SSH(Secure Shell) File Transfer Protocol(SFTP)을 지원하는 서버에서만 사용됩니다. SSH 키 쌍을 생성하는 자세한 방법은 서비스 관리 사용자를 위한 SSH 키 생성 섹션을 참조하세요.

6. SSH public key(SSH 퍼블릭 키)에는 SSH 키 쌍의 SSH 퍼블릭 키 부분을 입력합니다.

   서비스에서 사용자의 키를 확인해야 새 사용자를 추가할 수 있습니다. SSH 키의 형식은 `ssh-rsa` *`string`*입니다. SSH 키 쌍을 생성하려면 서비스 관리 사용자를 위한 SSH 키 생성 섹션을 참조하세요.
7. 키 추가를 선택합니다.

사용자의 SSH 공개 키를 삭제하려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 탐색 창에서 서버를 선택합니다.
3. 서버 ID 열에서 식별자를 선택하여 서버 세부 정보 페이지를 표시합니다.
4. 사용자에서 사용자 이름을 선택하면 사용자 세부 정보 페이지가 표시됩니다.
5. 공개 키를 삭제하려면 해당 SSH 키 확인란을 선택하고 삭제를 선택합니다.

AWS 디렉터리 서비스 ID 제공자 사용

이 항목에서는 AWS Directory Service ID 제공자를 사용하는 방법에 대해 설명합니다 AWS Transfer Family.

주제

- 사용 AWS Directory Service for Microsoft Active Directory
- Azure 액티브 AWS 디렉터리 도메인 서비스를 위한 디렉터리 서비스 사용

사용 AWS Directory Service for Microsoft Active Directory

를 AWS Transfer Family 사용하여 파일 전송 최종 사용자를 인증할 수 있습니다. AWS Directory Service for Microsoft Active Directory최종 사용자의 자격 증명을 변경하거나 사용자 지정 권한 부여자를 사용하지 않고도 Active Directory 인증을 사용하는 파일 전송 워크플로를 원활하게 마이그레이션할 수 있습니다.

를 사용하면 Amazon Simple Storage Service (Amazon S3) 또는 Amazon Elastic File System (Amazon EFS) 에 저장된 데이터에 대해 SFTP, FTPS 및 FTP를 통한 액세스 권한을 AWS Directory Service 사용자와 그룹에 안전하게 제공할 수 있습니다. AWS Managed Microsoft AD Active Directory를 사용하여 사용자의 자격 증명을 저장하면 이제 이러한 사용자에 대한 파일 전송을 더 쉽게 활성화할 수 있습니다.

Active Directory 커넥터를 사용하여 온프레미스 환경이나 AWS 클라우드의 Active Directory 그룹에 대한 액세스를 제공할 수 있습니다. AWS Managed Microsoft AD Microsoft Windows 환경 (AWS 클라우드 또는 온-프레미스 네트워크) 에 이미 구성되어 있는 사용자에게 ID를 사용하는 AWS Transfer Family 서버에 AWS Managed Microsoft AD 대한 액세스 권한을 부여할 수 있습니다.

> **Note**
>
> - AWS Transfer Family 는 Simple AD를 지원하지 않습니다.
> - Transfer Family는 리전 간 Active Directory 구성을 지원하지 않습니다. Transfer Family 서버와 동일한 리전에 있는 Active Directory 통합만 지원합니다.
> - Transfer Family는 기존 RADIUS 기반 MFA 인프라에 대해 멀티 팩터 인증 (MFA) 을 활성화하기 위해 둘 중 하나 AWS Managed Microsoft AD 또는 AD Connector를 사용하는 것을 지원하지 않습니다.
> - AWS Transfer Family 관리형 Active Directory의 복제 영역은 지원하지 않습니다.

사용하려면 AWS Managed Microsoft AD다음 단계를 수행해야 합니다.

1. AWS Directory Service 콘솔을 사용하여 AWS Managed Microsoft AD 디렉터리를 하나 이상 생성합니다.
2. Transfer Family 콘솔을 사용하여 ID AWS Managed Microsoft AD 제공자로 사용할 서버를 만들 수 있습니다.
3. 하나 이상의 AWS Directory Service 그룹에서 액세스 권한을 추가하세요.
4. 필수 사항은 아니지만 사용자 액세스를 테스트하고 확인하는 것이 좋습니다.

주제

사용을 시작하기 전 AWS Directory Service for Microsoft Active Directory

AD 그룹의 고유 식별자를 입력합니다.

사용하려면 AWS Managed Microsoft AD먼저 Microsoft AD 디렉터리의 각 그룹에 대해 고유한 식별자를 제공해야 합니다. 각 그룹의 SID(보안 식별자)를 사용하여 이 작업을 수행할 수 있습니다. 연결하는 그룹의 사용자는 AWS Transfer Family를 사용하여 활성화된 프로토콜을 통해 Amazon S3 또는 Amazon EFS 리소스에 액세스할 수 있습니다.

다음 Windows PowerShell 명령을 사용하여 그룹의 SID를 검색하고 그룹 이름으로 *YourGroupName*대체하십시오.

```
Get-ADGroup -Filter {samAccountName -like "YourGroupName*"} -Properties * | Select
 SamAccountName,ObjectSid
```

> **Note**
>
> 를 ID AWS Directory Service 공급자로 사용하고 있고 값이 다른 경우 `userPrincipalName` 의 `SamAccountName` 값을 AWS Transfer Family 수락합니다. `SamAccountName` Transfer Family는 `userPrincipalName`에 지정된 값을 수락하지 않습니다.

역할에 AWS Directory Service 권한 추가

ID AWS Directory Service 공급자로 사용하려면 AWS Directory Service API 권한도 필요합니다. 다음 권한이 필요하거나 제안됩니다.

- Transfer Family에서 디렉토리를 조회하려면 `ds:DescribeDirectories`가 필요합니다.
- Transfer Family에 대한 승인을 추가하려면 `ds:AuthorizeApplication`가 필요합니다.
- `ds:UnauthorizeApplication`에서는 서버 생성 프로세스 중에 문제가 발생할 경우를 대비하여 임시로 생성된 모든 리소스를 제거하는 것이 좋습니다.

Transfer Family 서버를 만드는 데 사용하는 역할에 이러한 권한을 추가하세요. 이러한 권한에 대한 자세한 내용은 AWS Directory Service API 권한: 작업, 리소스 및 조건 참조를 참조하세요.

액티브 디렉터리 영역 사용

Active Directory 사용자가 AWS Transfer Family 서버에 액세스하도록 하는 방법을 고려할 때는 사용자의 영역과 해당 그룹의 영역을 염두에 두세요. 가장 바람직한 것은 사용자 영역과 그룹 영역이 일치해야 하는 것입니다. 즉, 사용자와 그룹이 모두 기본 영역에 있거나 둘 다 신뢰할 수 있는 영역에 속해 있다는 뜻입니다. 그렇지 않은 경우 Transfer Family에서 사용자를 인증할 수 없습니다.

사용자를 테스트하여 구성이 올바른지 확인할 수 있습니다. 자세한 내용은 테스트 사용자를 참조하세요. 사용자/그룹 영역에 문제가 있는 경우 사용자 그룹에 연결된 액세스 권한을 찾을 수 없습니다라는 오류 메시지가 표시됩니다.

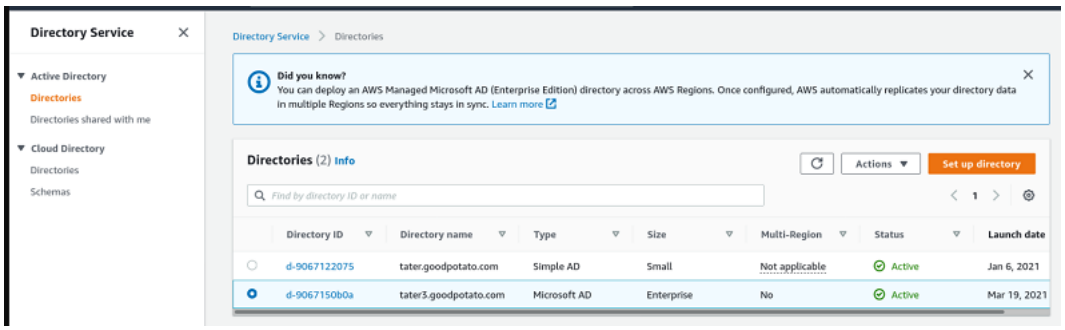
ID AWS Managed Microsoft AD 공급자로 선택

이 섹션에서는 AWS Directory Service for Microsoft Active Directory 서버와 함께 사용하는 방법을 설명합니다.

Transfer AWS Managed Microsoft AD Family와 함께 사용하려면

1. 에 AWS Management Console 로그인하고 https://console.aws.amazon.com/directoryservicev2/ 에서 AWS Directory Service 콘솔을 엽니다.

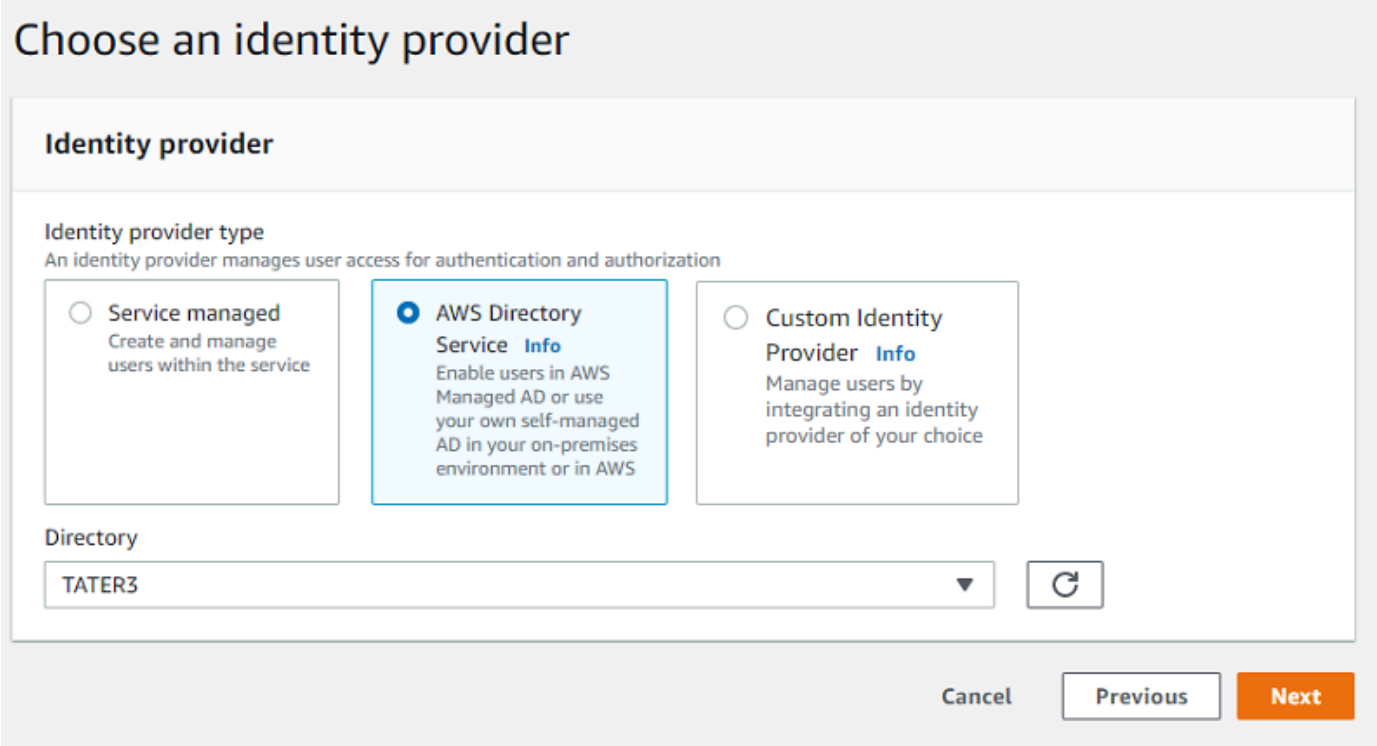
   AWS Directory Service 콘솔을 사용하여 하나 이상의 관리 디렉터리를 구성할 수 있습니다. 자세한 내용을 알아보려면 AWS Directory Service 관리 안내서의 AWS Managed Microsoft AD 단원을 참조하세요.



2. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 열고 [서버 만들기]를 선택합니다.
3. 프로토콜 선택 페이지의 목록에서 하나 이상의 프로토콜을 선택합니다.

> **Note**
>
> FTPS를 선택하는 경우 AWS Certificate Manager 인증서를 제공해야 합니다.

4. 자격 증명 제공자 선택에서 AWS Directory Service를 선택합니다.



5. 디렉터리 목록에는 구성한 모든 관리 디렉터리가 포함됩니다. 목록에서 디렉터리를 선택하고 다음을 선택합니다.

> **Note**
>
> - 교차 계정 및 공유 디렉터리는 지원되지 않습니다. AWS Managed Microsoft AD
> - Directory Service를 ID 제공자로 사용하여 서버를 설정하려면 몇 가지 AWS Directory Service 권한을 추가해야 합니다. 자세한 내용은 사용을 시작하기 전 AWS Directory Service for Microsoft Active Directory를 참조하세요.

6. 서버 생성을 완료하려면 다음 절차 중 하나를 사용합니다.

- SFTP 지원 서버 생성
- FTPS 지원 서버 생성
- FTP 지원 서버 생성

해당 절차에서 ID 제공자 선택 다음 단계를 계속 진행하세요.

> **Important**
>
> Transfer Family 서버에서 Microsoft AD 디렉터리를 사용한 AWS Directory Service 경우에는 해당 디렉터리를 삭제할 수 없습니다. 먼저 서버를 삭제해야 디렉터리를 삭제할 수 있습니다.

그룹에 액세스 권한 부여

서버를 만든 후에는 를 사용하여 활성화된 프로토콜을 통해 파일을 업로드하고 다운로드할 수 있는 액세스 권한을 부여할 디렉터리의 그룹을 선택해야 AWS Transfer Family합니다. 액세스 권한을 생성하여 이 작업을 수행할 수 있습니다.

> **Note**
>
> 사용자는 액세스 권한을 부여하는 그룹에 직접 속해야 합니다. 예를 들어 Bob은 사용자이고 그룹 A에 속해 있으며 그룹 A 자체가 그룹 B에 포함되어 있다고 가정해 보겠습니다.
>
> - 그룹 A에 액세스 권한을 부여하면 Bob에게 액세스 권한이 부여됩니다.
> - 그룹 A가 아닌 그룹 B에 액세스 권한을 부여하는 경우 Bob은 액세스 권한을 갖지 않습니다.

그룹에 액세스 권한을 부여하려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 서버 세부 정보 페이지로 이동합니다.
3. 액세스 섹션에서 액세스 추가를 선택합니다.
4. 이 서버에 액세스하려는 AWS Managed Microsoft AD 디렉터리의 SID를 입력합니다.

> **Note**
>
> 그룹의 SID를 찾는 방법에 대한 자세한 내용은 the section called “사용을 시작하기 전 AWS Directory Service for Microsoft Active Directory”를 참조하세요.

5. Access에서 그룹의 AWS Identity and Access Management (IAM) 역할을 선택합니다.
6. 정책 섹션에서 정책을 선택합니다. 기본 설정은 없음입니다.
7. 홈 디렉터리의 경우 그룹의 홈 디렉터리에 해당하는 S3 버킷을 선택합니다.

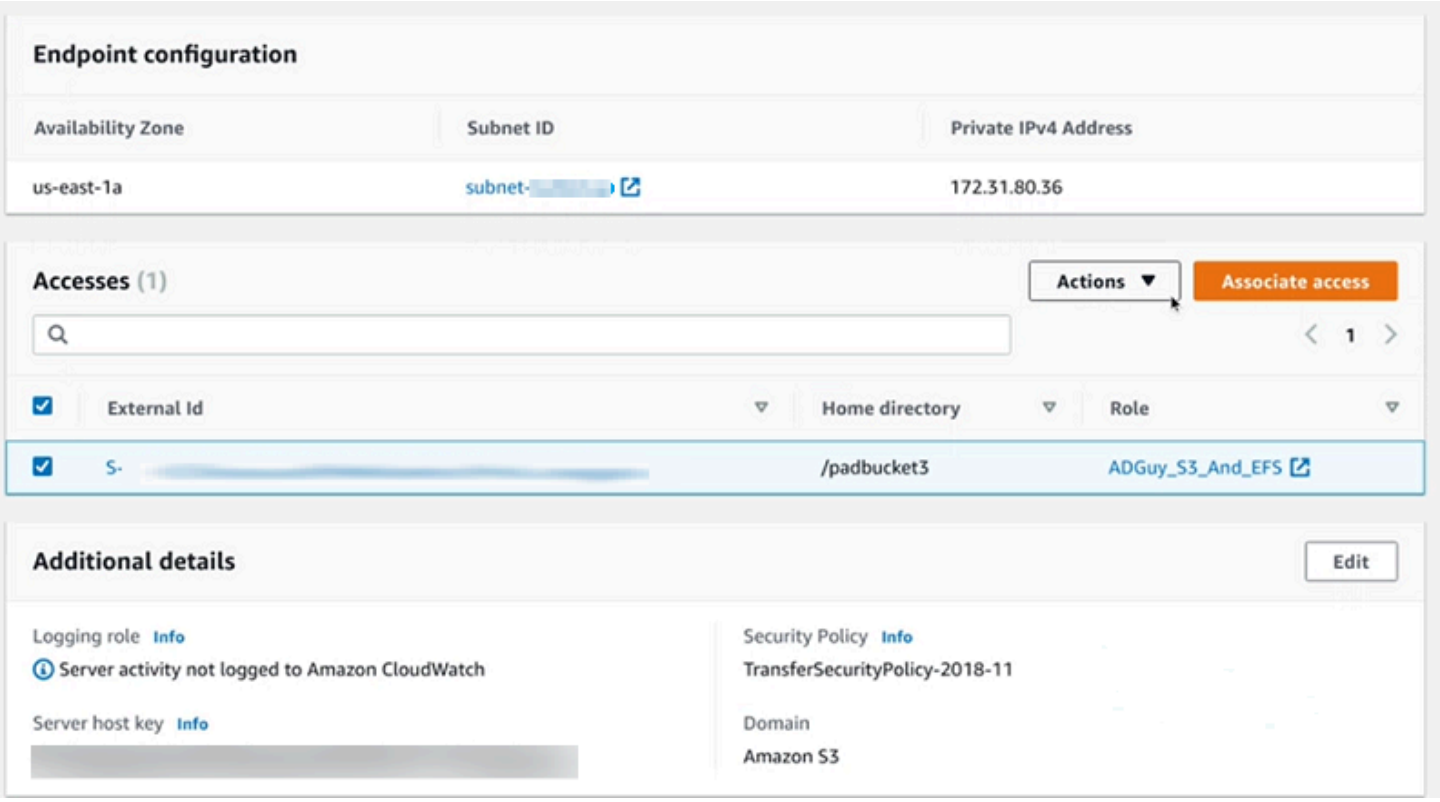
> **Note**
>
> 세션 정책을 생성하여 사용자에게 표시되는 버킷 부분을 제한할 수 있습니다. 예를 들어, `/filetest` 디렉터리 아래에 있는 자신의 폴더로만 사용자를 제한하려면 상자에 다음 텍스트를 입력합니다.
>
> ```
> /filetest/${transfer:UserName}
> ```
>
> 세션 정책 생성에 대한 자세한 설명은 Amazon S3 버킷을 위한 세션 정책 생성 섹션을 참조하세요.

8. 추가를 선택하여 연결을 생성합니다.
9. 서버를 선택합니다.
10. 액세스 추가를 선택합니다.
    - 그룹의 SID를 입력합니다.

> **Note**
>
> SID를 찾는 방법에 대한 자세한 내용은 the section called “사용을 시작하기 전 AWS Directory Service for Microsoft Active Directory”를 참조하세요.

11. 액세스 추가를 선택합니다.

액세스 섹션에는 서버에 대한 액세스가 나열됩니다.



테스트 사용자

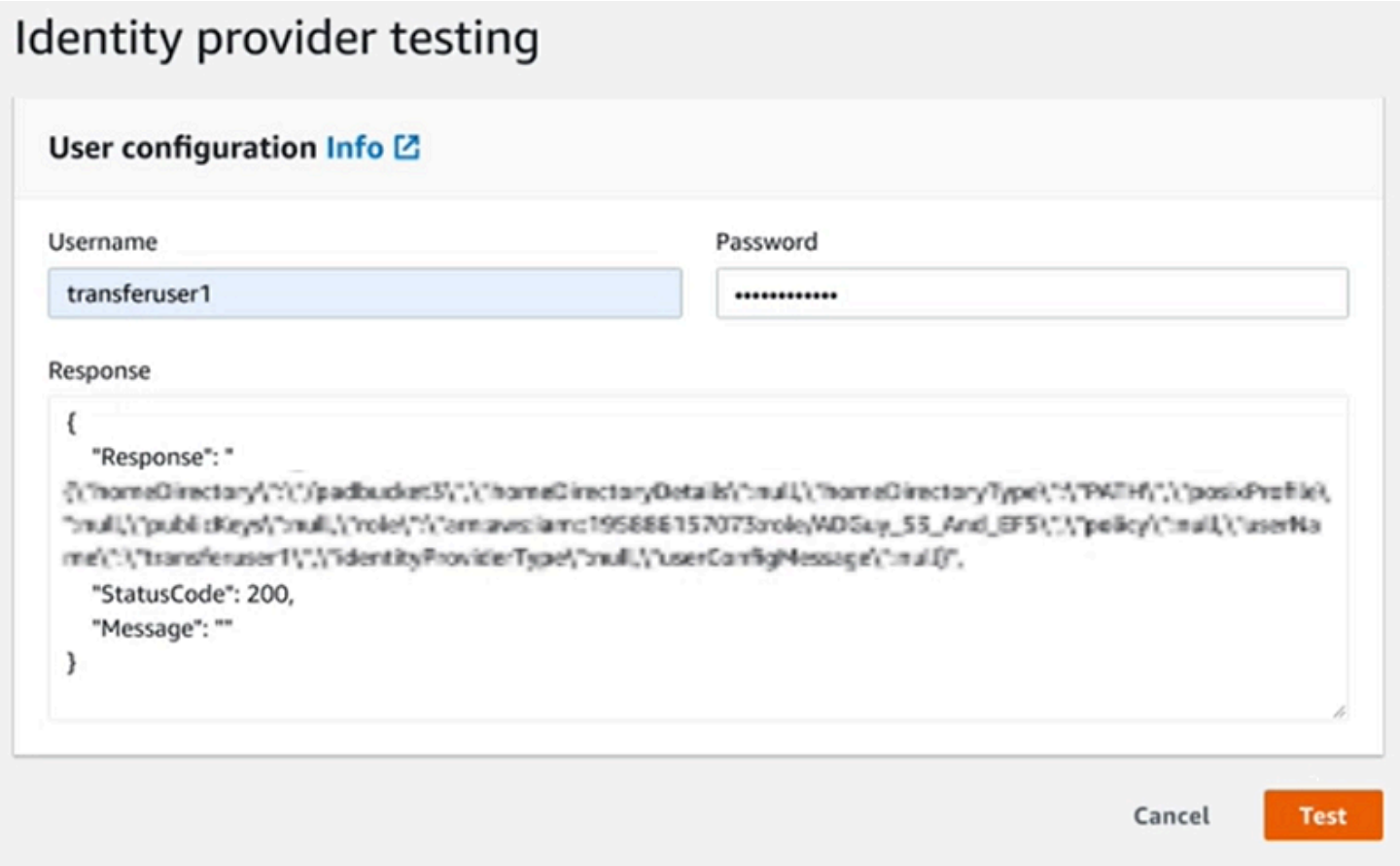
사용자가 서버의 AWS Managed Microsoft AD 디렉터리에 액세스할 수 있는지 테스트할 수 있습니다.

> **Note**
>
> 사용자는 엔드포인트 구성 페이지의 액세스 섹션에 나열된 정확히 하나의 그룹(외부 ID)에 속해야 합니다. 사용자가 그룹에 속하지 않거나 둘 이상의 그룹에 속해 있는 경우 해당 사용자에게 액세스 권한이 부여되지 않습니다.

특정 사용자에게 액세스 권한이 있는지 테스트하려면

1. 서버 세부 정보 페이지에서 작업을 선택한 다음 테스트를 선택합니다.
2. 자격 증명 제공자 테스트의 경우 액세스 권한이 있는 그룹 중 하나에 속하는 사용자의 로그인 자격 증명을 입력합니다.
3. 테스트를 선택합니다.

ID 제공자 테스트에 성공하면 선택한 사용자에게 서버 액세스 권한이 부여되었음을 알 수 있습니다.



사용자가 액세스 권한이 있는 두 개 이상의 그룹에 속해 있는 경우 다음과 같은 응답을 받게 됩니다.

```
"Response":"",
"StatusCode":200,
"Message":"More than one associated access found for user's groups."
```

그룹의 서버 액세스 삭제

그룹의 서버 액세스 삭제

1. 서버 세부 정보 페이지에서 작업을 선택한 다음 액세스 삭제를 선택합니다.
2. 대화 상자에서 이 그룹에 대한 액세스 권한을 제거합니다.

서버 세부 정보 페이지로 돌아가면 이 그룹에 대한 액세스 권한이 더 이상 나열되지 않는 것을 볼 수 있습니다.

SSH(보안 셸)를 사용하여 서버에 연결

서버와 사용자를 구성한 후 SSH를 사용하여 서버에 연결하고 액세스 권한이 있는 사용자의 정식 사용자 이름을 사용할 수 있습니다.

```
sftp user@active-directory-domain@vpc-endpoint
```

예를 들면 `transferuserexample@mycompany.com@vpce-0123456abcdef-789xyz.vpc-svc-987654zyxabc.us-east-1.vpce.amazonaws.com`입니다.

이 형식은 페더레이션 검색을 대상으로 하므로 크기가 클 수 있는 Active Directory의 검색을 제한합니다.

> **Note**
>
> 간단한 사용자 이름을 지정할 수 있습니다. 하지만 이 경우 Active Directory 코드가 페더레이션의 모든 디렉터리를 검색해야 합니다. 이로 인해 검색이 제한될 수 있으며 사용자에게 액세스 권한이 있어야 하는 경우에도 인증이 실패할 수 있습니다.

인증이 완료되면 사용자는 사용자를 구성할 때 지정한 홈 디렉터리에 위치해 있습니다.

포리스트 및 AWS Transfer Family 트러스트를 사용하여 자체 관리형 Active Directory에 연결

자체 관리형 Active Directory (AD) 의 사용자는 싱글 사인온 액세스 및 Transfer AWS 계정 Family 서버에도 사용할 AWS IAM Identity Center 수 있습니다. 이를 위해 다음과 같은 옵션을 AWS Directory Service 사용할 수 있습니다.

- 단방향 포리스트 트러스트 (온-프레미스 Active Directory의 송신 AWS Managed Microsoft AD 및 수신) 는 루트 도메인에서만 작동합니다.
- 하위 도메인의 경우 다음 중 하나를 사용할 수 있습니다.
  - 온-프레미스 Active Directory AWS Managed Microsoft AD 간에는 양방향 신뢰를 사용하십시오.
  - 각 하위 도메인에는 단방향 외부 트러스트를 사용합니다.

예를 들어 `transferuserexample@mycompany.com` 신뢰할 수 있는 도메인을 사용하여 서버에 연결하는 경우 사용자는 신뢰할 수 있는 도메인을 지정해야 합니다.

Azure 액티브 AWS 디렉터리 도메인 서비스를 위한 디렉터리 서비스 사용

- SFTP 전송 요구 사항에 맞게 기존 액티브 디렉터리 포리스트를 활용하려면 Active Directory Connector를 사용할 수 있습니다.
- 완전 관리형 서비스에서 Active Directory의 이점과 고가용성을 활용하려면 AWS Directory Service for Microsoft Active Directory를 사용할 수 있습니다. 자세한 내용은 AWS 디렉터리 서비스 ID 제공자 사용를 참조하세요.

이 항목에서는 Active Directory Connector 및 Azure Active Directory 도메인 서비스(Azure ADDS)를 사용하여 Azure Active Directory에서 SFTP Transfer 사용자를 인증하는 방법을 설명합니다.

주제

Azure AWS Active Directory 도메인 서비스용 디렉터리 서비스를 사용하기 전에

를 AWS위해서는 다음이 필요합니다.

- Transfer Family 서버를 사용하는 AWS 지역의 가상 사설 클라우드 (VPC)
- VPC에 최소 두 개의 프라이빗 서브넷이 있어야 함
- VPC는 인터넷에 연결되어 있어야 함
- Microsoft Azure와의 site-to-site VPN 연결을 위한 고객 게이트웨이 및 가상 프라이빗 게이트웨이

Microsoft Azure의 경우 다음이 필요합니다.

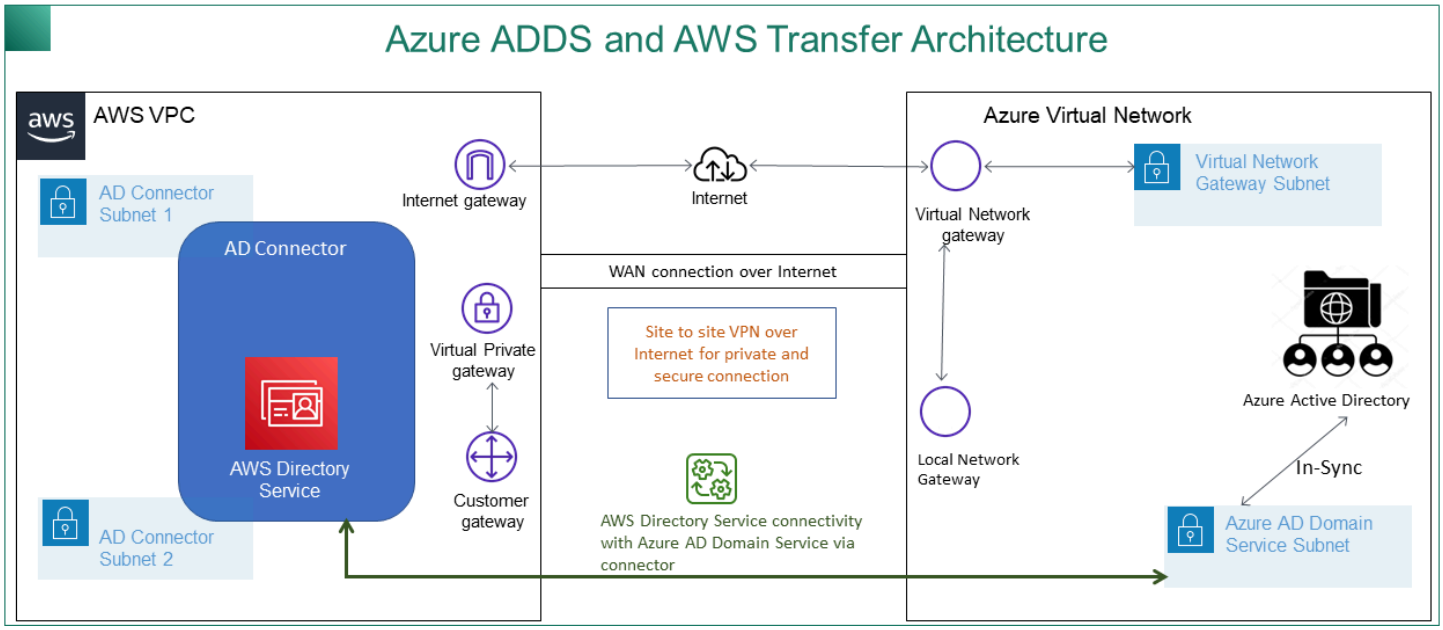
- Azure Active Directory 및 Active Directory 도메인 서비스(Azure ADDS)
- Azure 리소스 그룹
- Azure 가상 네트워크

- Amazon VPC와 Azure 리소스 그룹 간의 VPN 연결

> **Note**
>
> 네이티브 IPSEC 터널을 통하거나 VPN 어플라이언스를 사용하여 연결할 수 있습니다. 이 항목에서는 Azure 가상 네트워크 게이트웨이와 로컬 네트워크 게이트웨이 간에 IPSEC 터널을 사용합니다. Azure ADDS 엔드포인트와 VPC가 있는 서브넷 간의 트래픽을 허용하도록 터널을 구성해야 합니다. AWS

- Microsoft Azure와의 site-to-site VPN 연결을 위한 고객 게이트웨이 및 가상 프라이빗 게이트웨이

다음 다이어그램은 시작하기 전에 필요한 구성을 보여줍니다.

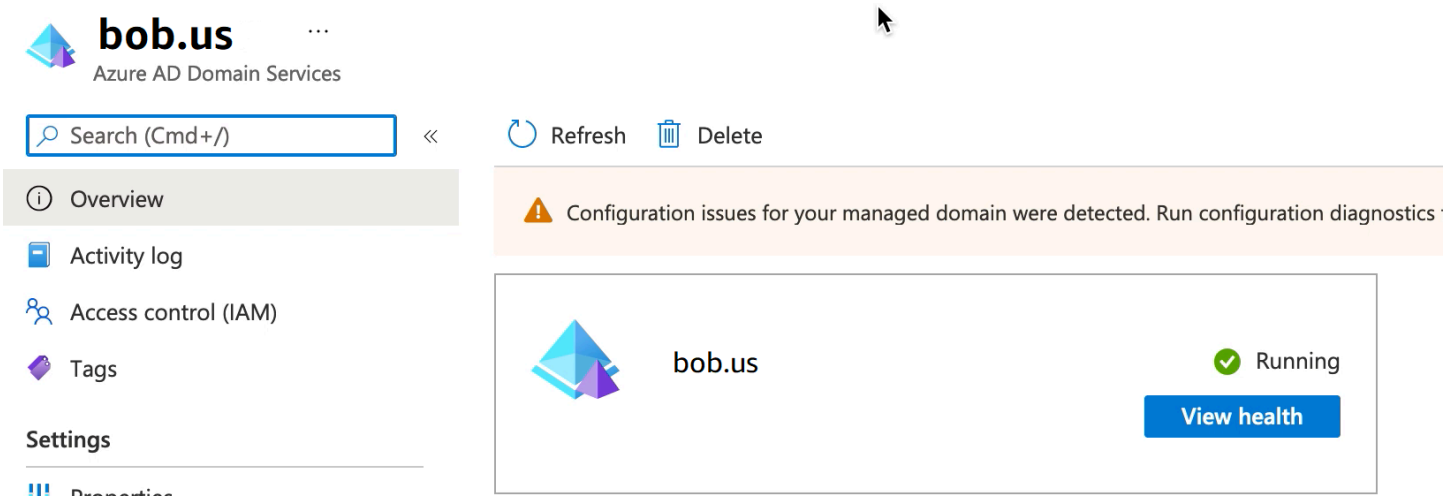


1단계: Azure Active Directory 도메인 서비스 추가

Azure AD는 기본적으로 도메인 가입 인스턴스를 지원하지 않습니다. 도메인 가입과 같은 작업을 수행하고 그룹 정책과 같은 도구를 사용하려면 관리자가 Azure Active Directory 도메인 서비스를 사용하도록 설정해야 합니다. Azure AD DS를 아직 추가하지 않았거나 기존 구현이 SFTP 전송 서버에서 사용할 도메인과 연결되어 있지 않은 경우 새 인스턴스를 추가해야 합니다.

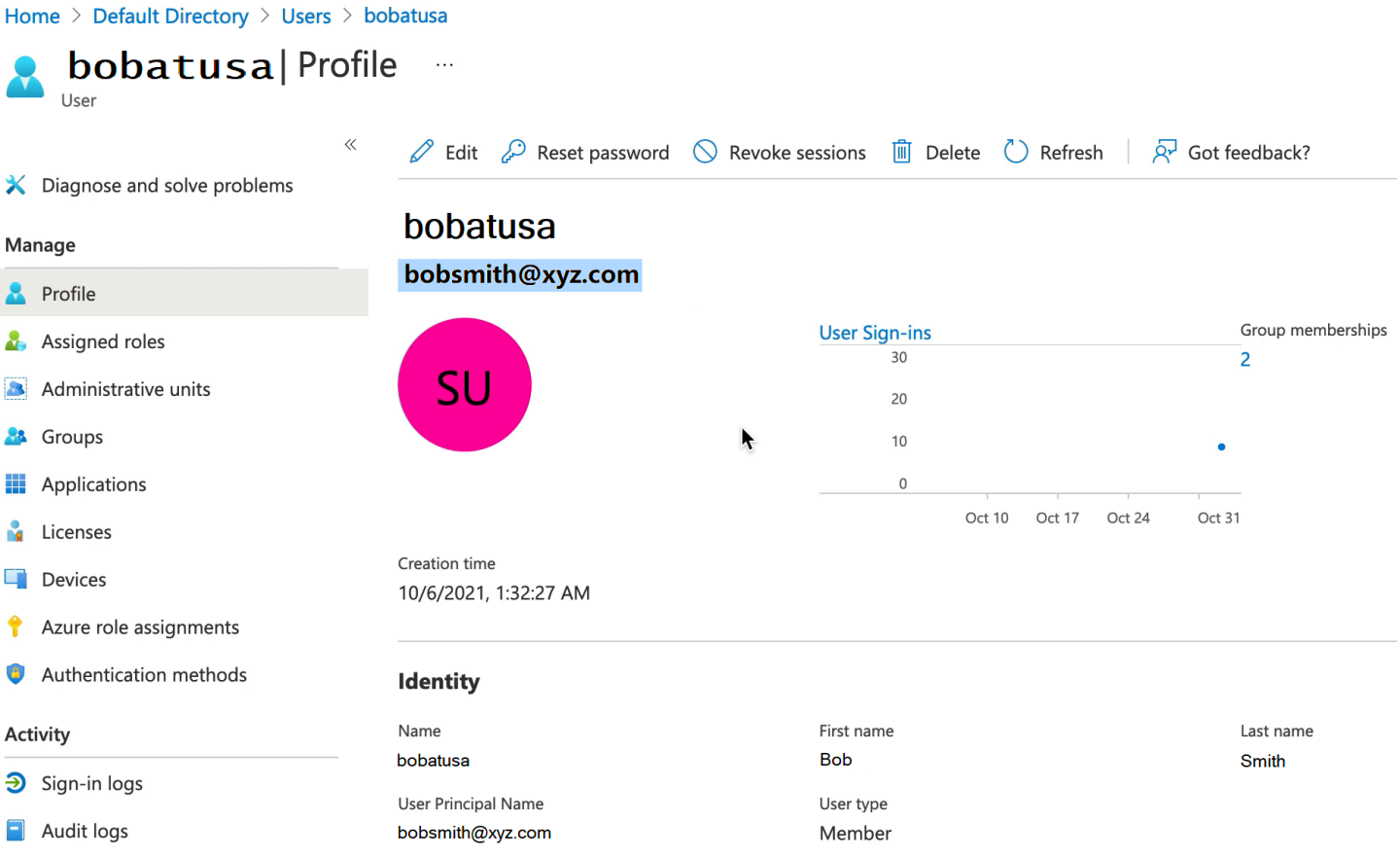
Azure Active Directory 도메인 서비스(Azure ADDS)를 사용하도록 설정하는 방법에 대한 자세한 내용은 자습서: Azure Active Directory 도메인 서비스 관리 도메인 생성 및 구성을 참조하세요.

> **Note**
>
> Azure ADDS를 사용하도록 설정하는 경우 SFTP 전송 서버를 연결하는 리소스 그룹 및 Azure AD 도메인에 맞게 구성되었는지 확인하세요.



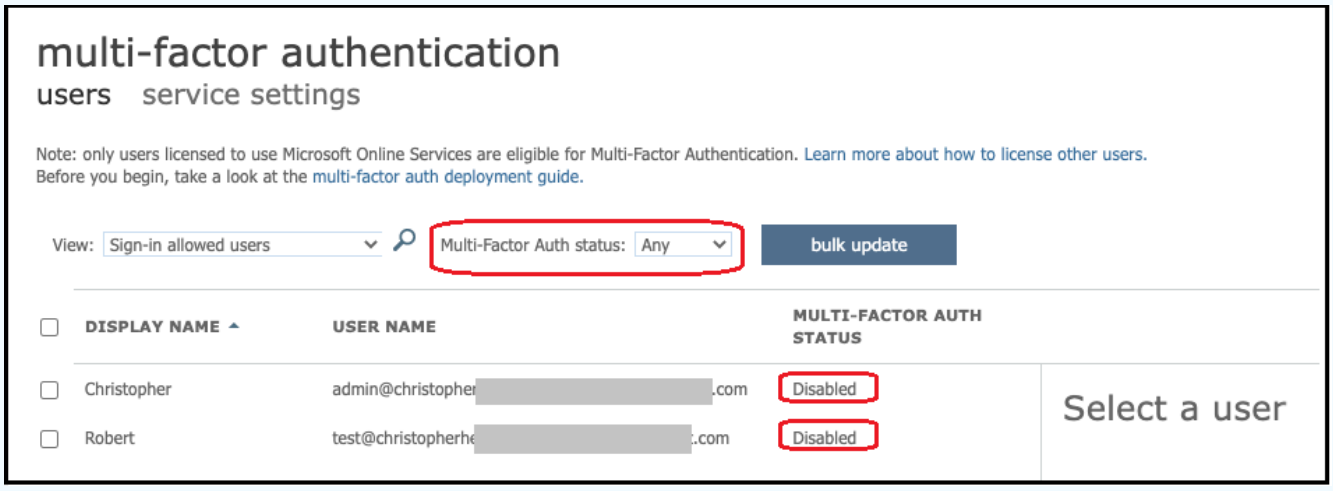
2단계: 서비스 계정 생성

Azure AD에는 Azure ADS의 관리자 그룹에 속하는 서비스 계정이 하나 있어야 합니다. 이 계정은 Active Directory 커넥터와 함께 사용됩니다. AWS 이 계정이 Azure ADDS와 동기화되어 있는지 확인하세요.



Tip

SFTP 프로토콜을 사용하는 Transfer Family 서버에서는 Azure Active Directory에 대한 다단계 인증이 지원되지 않습니다. Transfer Family 서버는 사용자가 SFTP에 인증한 후에는 MFA 토큰을 제공할 수 없습니다. 연결을 시도하기 전에 MFA를 비활성화해야 합니다.



3단계: AD AWS Connector를 사용하여 디렉터리 설정

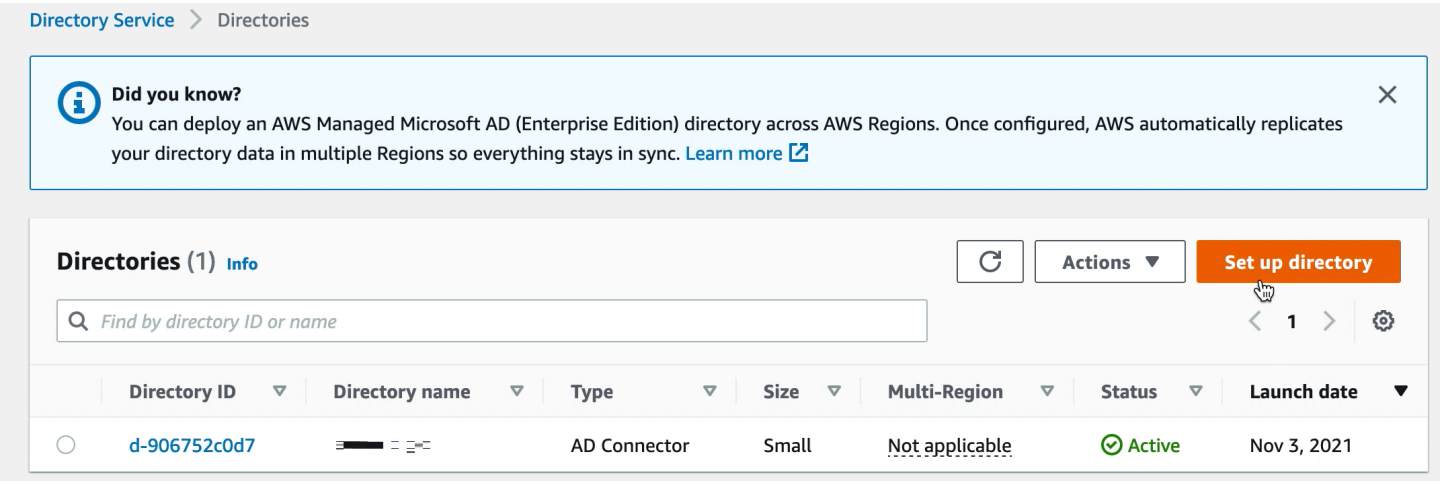
Azure ADDS를 구성하고 AWS VPC와 Azure 가상 네트워크 간에 IPSEC VPN 터널을 사용하여 서비스 계정을 만든 후에는 모든 EC2 인스턴스에서 Azure ADDS DNS IP 주소를 핑하여 연결을 테스트할 수 있습니다. AWS

연결이 활성화되었는지 확인한 후 아래에서 계속할 수 있습니다.

AD AWS Connector를 사용하여 디렉터리를 설정하려면

1. 디렉터리 서비스 콘솔을 열고 디렉터리를 선택합니다.
2. 디렉터리 설정을 선택합니다.
3. 디렉터리 타입으로는 AD Connector를 선택합니다.
4. 디렉터리 크기를 선택하고 다음을 선택한 다음 VPC와 서브넷을 선택합니다.
5. 다음을 선택하고 다음과 같이 필드를 채웁니다.
   - 디렉터리 DNS 이름: Azure ADS에 사용할 도메인 이름을 입력합니다.
   - DNS IP 주소: Azure ADS IP 주소를 입력합니다.
   - 서버 계정 사용자 이름 및 암호: 2단계: 서비스 계정 생성에서 만든 서비스 계정의 세부 정보를 입력합니다.
6. 화면을 완료하여 디렉터리 서비스를 생성합니다.

이제 디렉터리 상태가 활성이어야 하며 SFTP 전송 서버에서 사용할 준비가 되었습니다.



4단계: AWS Transfer Family 서버 설정

SFTP 프로토콜과 ID 제공자 타입의 AWS Directory Service를 사용하여 Transfer Family 서버를 생성합니다. 디렉터리 드롭다운 목록에서 3단계: AD Connector를 사용하여 AWS 디렉터리 설정에서 추가한 디렉터리를 선택합니다.

> **Note**
>
> Transfer Family 서버에서 Microsoft AD AWS 디렉터리를 사용한 경우 디렉터리 서비스에서 해당 디렉터리를 삭제할 수 없습니다. 먼저 서버를 삭제해야 디렉터리를 삭제할 수 있습니다.

5단계: 그룹에 액세스 권한 부여

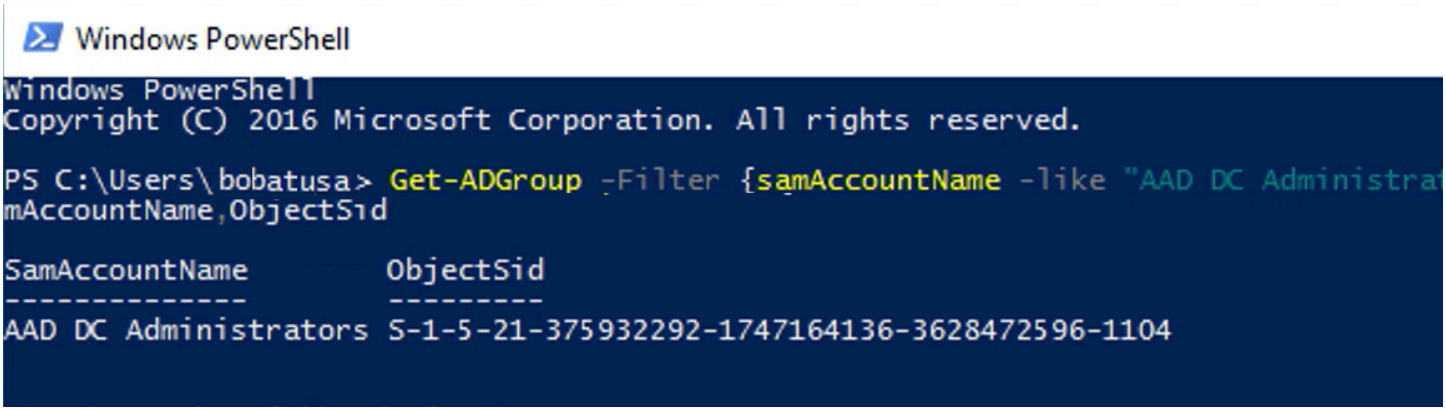
서버를 만든 후에는 를 사용하여 활성화된 프로토콜을 통해 파일을 업로드하고 다운로드할 수 있는 액세스 권한을 부여할 디렉터리의 그룹을 선택해야 AWS Transfer Family합니다. 액세스 권한을 생성하여 이 작업을 수행할 수 있습니다.

> **Note**
>
> 사용자는 액세스 권한을 부여하는 그룹에 직접 속해야 합니다. 예를 들어 Bob은 사용자이고 그룹 A에 속해 있으며 그룹 A 자체가 그룹 B에 포함되어 있다고 가정해 보겠습니다.
>
> - 그룹 A에 액세스 권한을 부여하면 Bob에게 액세스 권한이 부여됩니다.
> - 그룹 A가 아닌 그룹 B에 액세스 권한을 부여하는 경우 Bob은 액세스 권한을 갖지 않습니다.

액세스 권한을 부여하려면 그룹의 SID를 검색해야 합니다.

다음 Windows PowerShell 명령을 사용하여 그룹의 SID를 검색하고 그룹 이름으로 *YourGroupName*바꿉니다.

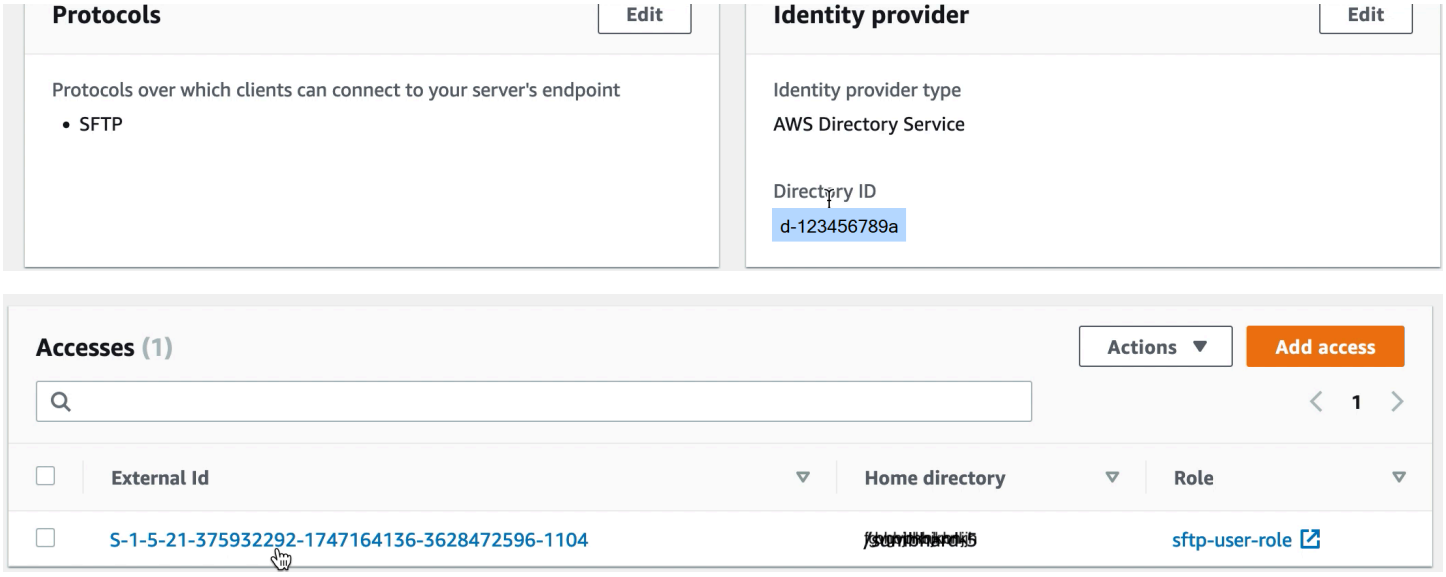
```
Get-ADGroup -Filter {samAccountName -like "YourGroupName*"} -Properties * | Select
 SamAccountName,ObjectSid
```



그룹에 액세스 권한을 부여하려면

1. https://console.aws.amazon.com/transfer/ 을 엽니다.
2. 서버 세부 정보 페이지로 이동한 다음 액세스 섹션에서 액세스 추가를 선택합니다.
3. 이전 절차의 출력에서 받은 SID를 입력합니다.
4. 액세스에서 그룹의 AWS Identity and Access Management 역할을 선택합니다.
5. 정책 섹션에서 정책을 선택합니다. 기본값은 없음입니다.
6. 홈 디렉터리의 경우 그룹의 홈 디렉터리에 해당하는 S3 버킷을 선택합니다.
7. 추가를 선택하여 연결을 생성합니다.

전송 서버의 세부 정보는 다음과 유사해야 합니다.



6단계: 사용자 테스트

사용자가 서버의 AWS Managed Microsoft AD 디렉터리에 액세스할 수 있는지 테스트(테스트 사용자)할 수 있습니다. 사용자는 엔드포인트 구성 페이지의 액세스 섹션에 나열된 정확히 하나의 그룹(외부 ID)에 속해야 합니다. 사용자가 그룹에 속하지 않거나 둘 이상의 그룹에 속해 있는 경우 해당 사용자에게 액세스 권한이 부여되지 않습니다.

사용자 지정 자격 증명 공급자와 작업

사용자를 인증하려면 기존 ID 공급자를 와 함께 AWS Transfer Family사용할 수 있습니다. Amazon S3 또는 Amazon Elastic File System (Amazon EFS) 에 액세스할 수 있도록 사용자를 인증하고 권한을 부여하는 AWS Lambda 함수를 사용하여 자격 증명 공급자를 통합합니다. 자세한 내용은 ID AWS Lambda 제공자를 통합하는 데 사용단원을 참조하세요. 또한 AWS Transfer Family Management Console에서 전송된 파일 수 및 바이트 수와 같은 지표에 대한 CloudWatch 그래프에 액세스할 수 있으므로 중앙 집중식 대시보드를 사용하여 파일 전송을 모니터링할 수 있는 단일 창을 제공합니다.

대안적으로 단일 Amazon API Gateway 방법을 사용하여 RESTful 인터페이스를 제공할 수도 있습니다. Transfer Family는 이 메서드를 호출하여 자격 증명 공급자에 연결합니다. 자격 증명 공급자는 사용자가 Amazon S3 또는 Amazon EFS에 액세스할 수 있도록 인증하고 권한을 부여합니다. ID 제공자를 통합하기 위한 RESTful API가 필요하거나 지역 차단 또는 속도 제한 요청에 RESTful API의 기능을 활용하는 AWS WAF 데 사용하려는 경우 이 옵션을 사용하십시오. 자세한 내용은 Amazon API Gateway를 ID 제공자 통합에 사용를 참조하세요.

어느 경우든 AWS Transfer Family 콘솔 또는 CreateServer API 작업을 사용하여 새 서버를 만들 수 있습니다.

> **Note**
>
> Transfer Family는 파일 전송 솔루션 구축 과정을 안내하는 블로그 게시물과 워크샵을 제공합니다. 이 솔루션은 관리형 SFTP/FTPS 엔드포인트와 Amazon Cognito 및 DynamoDB를 활용하여 AWS Transfer Family 사용자 관리를 수행합니다.
> 블로그 게시물은 Amazon Cognito를 Amazon AWS Transfer Family S3와 함께 자격 증명 공급자로 사용하기 페이지에서 확인할 수 있습니다. 워크숍에 대한 세부 정보는 여기에서 확인할 수 있습니다.

AWS Transfer Family 사용자 지정 ID 공급자를 사용하기 위한 다음 옵션을 제공합니다.

- ID 공급자 AWS Lambda 연결에 사용 - Lambda 함수가 지원하는 기존 ID 공급자를 사용할 수 있습니다. 귀하는 Lambda 함수의 명칭을 제공합니다. 자세한 내용은 ID AWS Lambda 제공자를 통합하는 데 사용를 참조하세요.
- Amazon API Gateway를 사용하는 자격 증명 공급자 연결 - Lambda 함수가 지원하는 API Gateway 방법을 생성하여 자격 증명 공급자로 사용할 수 있습니다. 사용자는 Amazon API Gateway URL과 호출 역할을 제공합니다. 자세한 내용은 Amazon API Gateway를 ID 제공자 통합에 사용를 참조하세요.

어느 옵션이든 귀하가 인증 방법도 지정할 수 있습니다.

- 암호 또는 키 — 사용자는 암호 또는 키로 인증할 수 있습니다. 이것이 기본값입니다.
- 암호만 해당 — 연결하려면 사용자가 암호를 입력해야 합니다.
- 키만 해당 - 연결하려면 사용자가 개인 키를 제공해야 합니다.
- 암호 및 키 — 연결하려면 사용자가 개인 키와 암호를 모두 제공해야 합니다. 서버가 먼저 키를 확인한 다음 키가 유효하면 암호를 입력하라는 메시지가 표시됩니다. 제공된 프라이빗 키가 저장된 퍼블릭 키와 일치하지 않는 경우 인증이 실패합니다.

여러 인증 방법을 사용하여 사용자 지정 ID 공급자를 통해 인증하기

Transfer Family 서버는 여러 인증 방법을 사용할 때 AND 로직을 제어합니다. Transfer Family는 이를 사용자 지정 ID 공급자에게 보내는 두 개의 개별 요청으로 취급하지만 그 효과는 합산됩니다.

인증이 완료되려면 두 요청 모두 올바른 응답과 함께 성공적으로 반환되어야 합니다. Transfer Family를 사용하려면 두 개의 응답이 완료되어야 합니다. 즉, 응답에는 필수 요소 (스토리지로 Amazon EFS를 사용하는 경우 역할, 홈 디렉터리, 정책 및 POSIX 프로필) 가 모두 포함되어 있습니다. 또한 Transfer Family에서는 암호 응답에 공개 키가 포함되지 않도록 요구합니다.

공개 키 요청에는 ID 제공자로부터 별도의 응답이 있어야 합니다. 암호 OR 키 또는 암호 및 키를 사용할 때 이 동작은 변경되지 않습니다.

SSH/SFTP 프로토콜은 먼저 소프트웨어 클라이언트에 공개 키 인증을 요청한 다음 암호 인증을 요청합니다. 이 작업을 수행하려면 사용자가 인증을 완료하기 전에 두 가지를 모두 완료해야 합니다.

주제

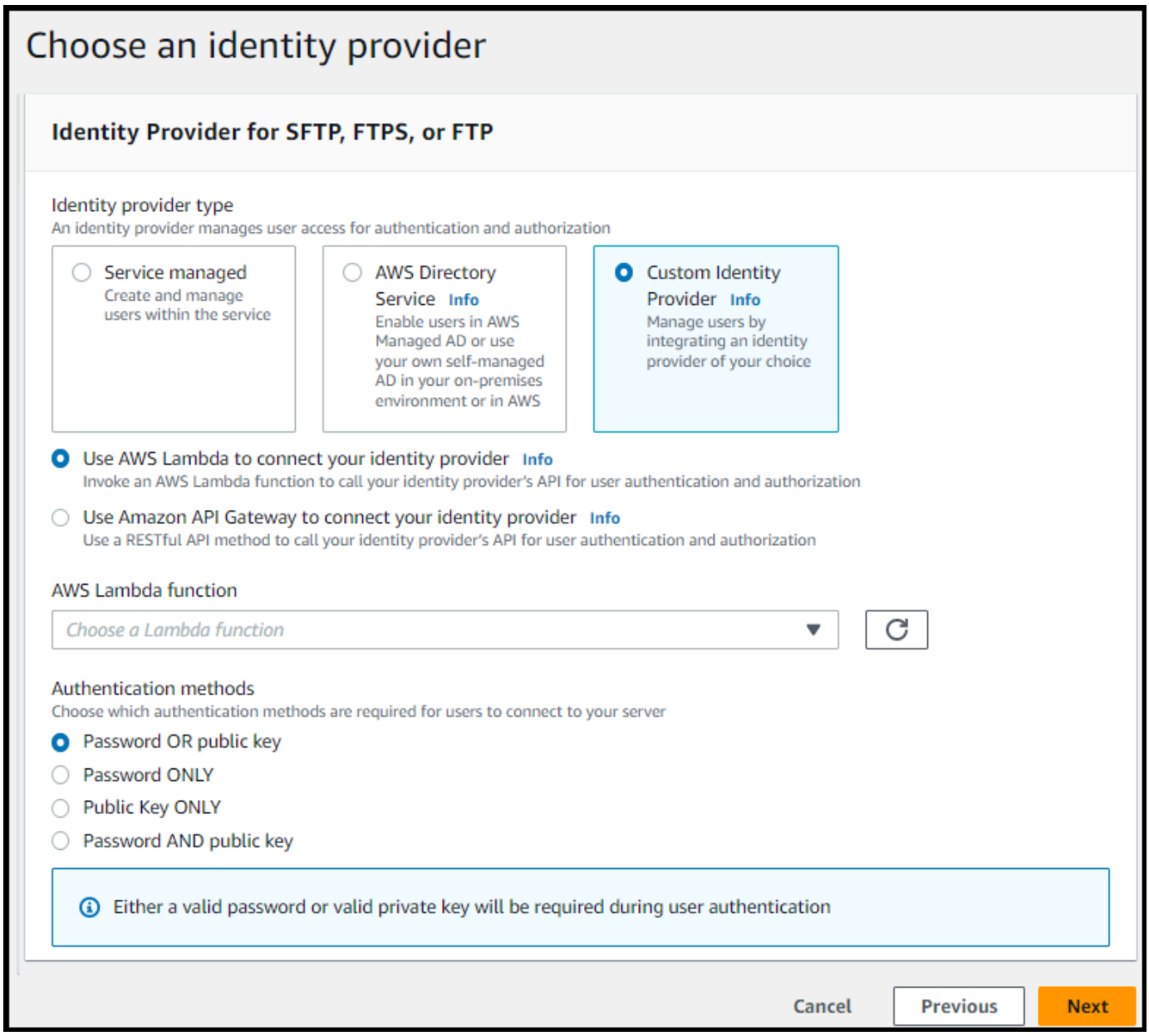
- ID AWS Lambda 제공자를 통합하는 데 사용
- Amazon API Gateway를 ID 제공자 통합에 사용

ID AWS Lambda 제공자를 통합하는 데 사용

사용자 지정 ID 공급자에 연결하는 AWS Lambda 함수를 만드세요. Okta, Secrets Manager와 같은 사용자 지정 ID 공급자를 사용하거나 권한 부여 및 인증 로직이 포함된 사용자 지정 데이터 저장소를 사용할 수 있습니다. OneLogin

> **Note**
>
> Lambda를 ID 공급자로 사용하는 Transfer Family 서버를 생성하기 전에 함수를 생성해야 합니다. Lambda 함수의 예는 예 Lambda 함수를 참조하세요. 또는 다음 중 하나를 사용하는 CloudFormation 스택을 배포할 수 있습니다. Lambda 함수 템플릿 또한 Lambda 함수가 Transfer Family를 신뢰하는 리소스 기반 정책을 사용하는지 확인하세요. 정책 예제는 Lambda 리소스 기반 정책을 참조하세요.

1. AWS Transfer Family 콘솔을 엽니다.
2. 서버 생성을 선택하여 서버 생성 페이지를 엽니다. ID 제공자 선택를 위해 다음 스크린샷과 같이 사용자 지정 ID 공급자를 선택합니다.



> **Note**
>
> 인증 방법은 SFTP를 Transfer Family 서버의 프로토콜 중 하나로 활성화한 경우에만 사용할 수 있습니다.

3. 기본값인 ID 제공자를 연결하는 AWS Lambda 데 사용이 선택되어 있는지 확인하십시오.
4. AWS Lambda 함수에서 Lambda 함수의 이름을 선택합니다.
5. 나머지 상자를 채운 다음 서버 만들기를 선택합니다. 서버를 만들기 위한 나머지 단계에 대한 자세한 내용은 SFTP, FTPS 또는 FTP 서버 엔드포인트 구성을 참조하세요.

Lambda 리소스 기반 정책

Transfer Family 서버 및 Lambda ARN을 참조하는 정책이 있어야 합니다. 예를 들어 ID 공급자에 연결되는 Lambda 함수와 함께 다음 정책을 사용할 수 있습니다. 정책은 JSON을 문자열로 이스케이프합니다.

```
"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**
>
> 상기 예 정책에서 각 *### ## ## ###*를 자신의 정보로 바꿉니다.

이벤트 메시지 구조

사용자 지정 IDP에 대해 권한 부여자 Lambda 함수로 전송되는 SFTP 서버의 이벤트 메시지 구조는 다음과 같습니다.

```
{
    "username": "value",
    "password": "value",
```

```
    "protocol": "SFTP",
    "serverId": "s-abcd123456",
    "sourceIp": "192.168.0.100"
}
```

서버로 전송되는 로그인 자격 증명의 값인 `username`과 `password`의 위치는 어디입니까?

예를 들어, 다음 명령을 입력해서 연결합니다.

```
sftp bobusa@server_hostname
```

새 암호를 두 번 입력하라는 메시지가 나타납니다.

```
Enter password:
    mysecretpassword
```

Lambda 함수 내에서 전달된 이벤트를 인쇄하여 Lambda 함수에서 이를 확인할 수 있습니다. 그것은 다음 텍스트 블록과 비슷하게 보여야 합니다.

```
{
    "username": "bobusa",
    "password": "mysecretpassword",
    "protocol": "SFTP",
    "serverId": "s-abcd123456",
    "sourceIp": "192.168.0.100"
}
```

이벤트 구조는 FTP와 FTPS와 비슷한데, 유일한 차이점은 SFTP라기 보다 오히려 아닌 해당 값이 `protocol`파라미터에 사용된다는 것입니다.

인증을 위한 Lambda 함수

다양한 인증 전략을 구현하려면 Lambda 함수를 편집하세요. 애플리케이션의 요구 사항을 충족하는 데 도움이 되도록 CloudFormation 스택을 배포할 수 있습니다. 자세한 내용은 개발자 안내서나 AWS Lambda Node.js로 Lambda 함수 구축을 참조하세요.

주제

- Lambda 함수 템플릿
- 유효 Lambda 값
- 예 Lambda 함수

- 구성 테스트

Lambda 함수 템플릿

인증을 위해 Lambda 함수를 사용하는 AWS CloudFormation 스택을 배포할 수 있습니다. 로그인 자격 증명을 사용하여 사용자를 인증하고 권한을 부여하는 여러 템플릿을 제공합니다. 이러한 템플릿 또는 AWS Lambda 코드를 수정하여 사용자 액세스를 추가로 사용자 지정할 수 있습니다.

> **Note**
>
> 템플릿에 FIPS 지원 보안 정책을 AWS CloudFormation 지정하여 FIPS 지원 AWS Transfer Family 서버를 만들 수 있습니다. 사용 가능한 보안 정책은 서버 보안 정책 AWS Transfer Family에 설명되어 있습니다.

인증에 사용할 스택을 만들려면 AWS CloudFormation

1. https://console.aws.amazon.com/cloudformation 에서 AWS CloudFormation 콘솔을 엽니다.
2. 사용 AWS CloudFormation 설명서의 AWS CloudFormation 스택 템플릿 선택에 나와 있는 기존 템플릿에서 스택을 배포하는 방법에 대한 지침을 따르십시오.
3. 다음 템플릿 중 하나를 사용하여 Transfer Family의 인증에 사용할 Lambda 함수를 생성합니다.

   - 클래식 (Amazon Cognito) 스택 템플릿

     에서 사용자 지정 ID 공급자로 사용할 템플릿을 만들기 AWS Lambda 위한 기본 템플릿입니다. AWS Transfer Family암호 기반 인증을 위해 Amazon Cognito에 대해 인증하며, 퍼블릭 키 기반 인증을 사용하는 경우 Amazon S3 버킷에서 퍼블릭 키가 반환됩니다. 배포 후에는 Lambda 함수 코드를 수정하여 다른 작업을 수행할 수 있습니다.

   - AWS Secrets Manager 스택 템플릿

     Secrets Manager를 ID 공급자로 통합하기 위해 AWS Transfer Family 서버와 AWS Lambda 함께 사용하는 기본 템플릿입니다. AWS Secrets Manager 형식의 `aws/transfer/`*`server-id`*`/`*`username`* 항목에 대해 인증합니다. 또한 암호에는 Transfer Family에 반환된 모든 사용자 속성에 대한 키-값 쌍이 들어 있어야 합니다. 배포 후에는 Lambda 함수 코드를 수정하여 다른 작업을 수행할 수 있습니다.

   - Okta 스택 템플릿: AWS Lambda AWS Transfer Family 서버와 함께 사용하여 Okta를 사용자 지정 ID 공급자로 통합하는 기본 템플릿입니다.

- Okta-MFA 스택 템플릿: AWS Lambda AWS Transfer Family 서버와 함께 사용하여 Okta를 MultiFactor 인증과 사용자 지정 ID 공급자로 통합하는 데 사용하는 기본 템플릿입니다.
- Azure Active Directory 템플릿: 이 스택에 대한 세부 정보는 Azure Active Directory를 사용한 인증 및 블로그 게시물에 설명되어 있습니다. AWS Transfer Family AWS Lambda

스택을 배포한 후에는 콘솔의 출력 탭에서 스택에 대한 세부 정보를 볼 수 있습니다. CloudFormation

사용자 지정 ID 공급자를 Transfer Family 워크플로에 통합하는 가장 쉬운 방법은 이러한 스택 중 하나를 배포하는 것입니다.

유효 Lambda 값

다음 표에는 사용자 지정 ID 공급자에 사용되는 Lambda 함수에 대해 Transfer Family가 허용하는 값에 대한 세부 정보가 설명되어 있습니다.

값	설명	필수
`Role`	Amazon S3 버킷 또는 Amazon EFS 파일 시스템에 대한 사용자 액세스를 제어하는 IAM 역할의 Amazon 리소스 이름(ARN)을 지정합니다. 이 역할에 연결된 정책은 Amazon S3 버킷 또는 Amazon EFS 파일 시스템에 대한 파일 송수신 시 사용자에게 제공할 액세스의 수준을 결정합니다. 또한 IAM 역할에는 사용자의 전송 요청을 처리할 때 서버가 해당 리소스에 액세스할 수 있도록 허용하는 신뢰 관계가 포함되어야 합니다.	필수

값	설명	필수
	신뢰 관계 설정에 대한 자세한 내용은 신뢰 관계를 구축하기 위해을 참조하세요.	
`PosixProfile`	Amazon EFS 파일 시스템에 대한 사용자의 액세스를 제어하는 사용자 ID(`Uid`), 그룹 ID(`Gid`) 및 보조 그룹 ID(`SecondaryGids`)를 포함한 전체 POSIX 자격 증명입니다. 파일 시스템의 파일 및 디렉터리에 설정된 POSIX 권한에 따라 Amazon EFS 파일 시스템에서 파일을 송수신할 때 사용자에게 제공되는 액세스 수준이 결정됩니다.	Amazon EFS 백업 스토리지에 필요
`PublicKeys`	이 사용자에게 유효한 SSH 퍼블릭 키 값 목록. 목록이 비어 있으면 유효한 로그인이 아님을 의미합니다. 암호 인증 중에는 반환할 수 없습니다.	선택 사항
`Policy`	여러 사용자에 대해 동일한 IAM 역할을 사용할 수 있도록 한 사용자에 대한 세션 정책입니다. 이 정책은 Amazon S3 버킷의 부분에 대한 사용자 액세스의 범위를 축소합니다.	선택 사항

값	설명	필수
HomeDirectoryType	사용자가 서버에 로그인하는 경우 홈 디렉터리가 될 랜딩 디렉터리(폴더) 타입입니다. • PATH로 설정하면, 사용자는 파일 전송 프로토콜 클라이언트에서와 같이 절대 Amazon S3 버킷 또는 Amazon EFS 경로를 볼 수 있습니다. • LOGICAL로 설정하면 HomeDirectoryDetails 파라미터에서 Amazon S3 또는 Amazon EFS 경로를 사용자에게 시각적으로 표시할 방법에 대한 매핑을 제공해야 합니다.	선택 사항
HomeDirectoryDetails	사용자에게 표시할 Amazon S3 또는 Amazon EFS 경로 및 키와 이러한 경로 및 키를 사용자에게 시각적으로 표시할 방법을 지정하는 논리적 디렉터리 매핑입니다. Entry 및 Target 쌍을 지정해야 합니다. 여기서 Entry는 경로가 표시되는 방식을 보여주고 Target는 실제 Amazon S3 경로입니다.	HomeDirectoryType 이 LOGICAL의 값을 가진 경우 필요
HomeDirectory	클라이언트를 사용하여 서버에 로그인하는 경우 랜딩 디렉터리(폴더)입니다.	선택 사항

> **Note**
>
> `HomeDirectoryDetails`는 JSON 맵의 문자열 표현입니다. 이는 실제 JSON 맵 객체인 `PosixProfile`과 문자열로 구성된 JSON 배열인 `PublicKeys`과 대조적입니다. 언어별 세부 정보는 코드 예를 참조하세요.

예 Lambda 함수

이 섹션에서는 NodeJS와 Python 모두에서 사용할 수 있는 Lambda 함수 몇 가지를 소개합니다.

> **Note**
>
> 이 예에서는 사용자, 역할, POSIX 프로필, 암호 및 홈 디렉토리 세부 정보가 모두 예시이므로 실제 값으로 바꿔야 합니다.

Logical home directory, NodeJS

다음 NodeJS 예 함수는 논리적 홈 디렉터리가 있는 사용자에 대한 세부 정보를 제공합니다.

```
// 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

다음 NodeJS 예 함수는 경로 기반의 홈 디렉터리가 있는 사용자에 대한 세부 정보를 제공합니다.

```
// 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

다음 Python 예 함수는 논리적 홈 디렉터리가 있는 사용자에 대한 세부 정보를 제공합니다.

```
# 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

다음 Python 예 함수는 경로 기반 홈 디렉터리가 있는 사용자에 대한 세부 정보를 제공합니다.

```
# 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
```

구성 테스트

사용자 지정 ID 공급자를 만든 후에는 구성을 테스트해야 합니다.

Console

AWS Transfer Family 콘솔을 사용하여 구성을 테스트하려면

1. AWS Transfer Family 콘솔을 엽니다.
2. 서버 페이지에서 새 서버를 선택하고 작업을 선택한 다음 테스트를 선택합니다.
3. AWS CloudFormation 스택을 배포할 때 설정한 사용자 이름 및 암호 텍스트를 입력합니다. 기본 옵션을 유지한 경우 사용자 이름은 `myuser` 이고 암호는 `MySuperSecretPassword`입니다.
4. 서버 프로토콜을 선택하고 소스 IP의 IP 주소 (AWS CloudFormation 스택을 배포할 때 설정한 경우) 를 입력합니다.

CLI

AWS CLI를 사용하여 구성을 테스트하려면

1. test-identity-provider 명령을 실행합니다. 다음 단계에서 설명하는 대로 *user input placeholder* 각각을 사용자 고유의 정보로 바꾸세요.

```
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. 서버 ID를 입력합니다.
3. AWS CloudFormation 스택을 배포할 때 설정한 사용자 이름과 비밀번호를 입력합니다. 기본 옵션을 유지한 경우 사용자 이름은 myuser 이고 암호는 MySuperSecretPassword입니다.
4. 서버 프로토콜과 소스 IP 주소 (AWS CloudFormation 스택을 배포할 때 설정한 경우) 를 입력합니다.

사용자 인증에 성공하면 테스트 결과 StatusCode: 200 HTTP 응답, 빈 문자열 Message: "" (그렇지 않으면 실패 이유가 포함됨) 및 Response 필드가 반환됩니다.

> **Note**
>
> 아래 응답 예에서 Response 필드는 '문자열화' (프로그램 내에서 사용할 수 있는 플랫 JSON 문자열로 변환) 된 JSON 객체이며, 사용자의 역할 및 권한에 대한 세부 정보를 포함합니다.

```
{
    "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": ""
}
```

Amazon API Gateway를 ID 제공자 통합에 사용

이 항목에서는 AWS Lambda 함수를 사용하여 API Gateway 메서드를 지원하는 방법을 설명합니다. ID 제공자를 통합하기 위해 RESTful API가 필요하거나 지역 차단 또는 속도 제한 요청에 RESTful API의 기능을 활용하는 AWS WAF 데 사용하려는 경우 이 옵션을 사용하십시오.

API Gateway를 사용하여 ID 공급자를 통합하는 경우의 제한

- 이 구성은 사용자 지정 도메인을 지원하지 않습니다.
- 이 구성은 프라이빗 API Gateway URL을 지원하지 않습니다.

둘 중 하나가 필요한 경우, API Gateway 없이 Lambda를 ID 공급자로서 사용할 수 있습니다. 자세한 내용은 ID AWS Lambda 제공자를 통합하는 데 사용를 참조하세요.

API Gateway 메서드를 사용하여 인증

Transfer Family의 ID 공급자로 사용할 API Gateway 메서드를 생성할 수 있습니다. 이 접근 방식은 API를 생성하고 제공할 수 있는 매우 안전한 방법을 제공합니다. API Gateway를 사용하면 HTTPS 엔드포인트를 생성하여 모든 수신 API 호출이 보다 안전하게 전송되도록 할 수 있습니다. API Gateway 서비스에 대한 자세한 내용은 API Gateway 개발자 안내서를 참조하세요.

API Gateway는 라는 권한 부여 방법을 제공하며`AWS_IAM`, 이 방법은 내부적으로 AWS 사용하는 것과 동일한 AWS Identity and Access Management IAM 기반 인증을 제공합니다. 로 `AWS_IAM` 인증을 활성화하면 API를 호출할 수 있는 명시적 권한을 가진 호출자만 해당 API의 API Gateway 메서드에 연결할 수 있습니다.

API Gateway 메서드를 Transfer Family의 사용자 지정 ID 공급자로 사용하려면 API Gateway 메서드에 IAM을 활성화해야 합니다. 이 프로세스의 일환으로 Transfer Family가 게이트웨이를 사용할 수 있는 권한을 IAM 역할에 제공합니다.

> **Note**
>
> 보안을 강화하기 위해 웹 애플리케이션 방화벽을 구성할 수 있습니다. AWS WAF 은(는) Amazon API Gateway에 전달되는 HTTP 및 HTTPS 요청을 모니터링할 수 있게 해주는 웹 애플리케이션 방화벽입니다. 자세한 내용은 웹 애플리케이션 방화벽 추가를 참조하세요.

Transfer Family를 통한 사용자 지정 인증에 API Gateway 방법을 사용하려면

1. 스택을 생성합니다. AWS CloudFormation 방법:

> **Note**
>
> 스택 템플릿은 Base64로 인코딩된 암호를 사용하도록 업데이트되었습니다. 자세한 내용은 을 참조하십시오. 템플릿 개선 사항 AWS CloudFormation

a. https://console.aws.amazon.com/cloudformation 에서 콘솔을 엽니다. AWS CloudFormation

b. 사용 AWS CloudFormation 설명서의 AWS CloudFormation 스택 템플릿 선택에 나와 있는 기존 템플릿에서 스택을 배포하는 방법에 대한 지침을 따르십시오.

c. 다음 기본 템플릿 중 하나를 사용하여 Transfer AWS Lambda Family에서 사용자 지정 ID 공급자로 사용할 지원 API Gateway 메서드를 만들 수 있습니다.

- 기본 스택 템플릿

  기본적으로 API Gateway 메서드는 하드 코딩된 SSH (Secure Shell) 키 또는 암호를 사용하여 단일 서버에서 단일 사용자를 인증하는 사용자 지정 ID 공급자로 사용됩니다. 배포 후에는 Lambda 함수 코드를 수정하여 다른 작업을 수행할 수 있습니다.

- AWS Secrets Manager 스택 템플릿

  기본적으로 API Gateway 메서드는 Secrets Manager에 있는 다음 형식의 `aws/transfer/server-id/username` 항목에 대해 인증합니다. 또한 암호에는 Transfer Family에 반환된 모든 사용자 속성에 대한 키-값 쌍이 들어 있어야 합니다. 배포 후에는 Lambda 함수 코드를 수정하여 다른 작업을 수행할 수 있습니다. 자세한 내용은 블로그 게시물 AWS Transfer Family 사용을 위한 암호 인증 활성화를 참조하십시오 AWS Secrets Manager.

- Okta 스택 템플릿

  API Gateway 메서드는 Transfer Family의 사용자 지정 ID 공급자로서 Okta와 통합됩니다. 자세한 내용은 블로그 게시물 Okta를 AWS Transfer Family를 사용한 ID 공급자로서 사용을 참조하세요.

사용자 지정 ID 공급자를 Transfer Family 워크플로에 통합하는 가장 쉬운 방법은 이러한 스택 중 하나를 배포하는 것입니다. 각 스택은 Lambda 함수를 사용하여 API Gateway에 기반한 API 메서드를 지원합니다. 그런 다음 Transfer Family에서 API 메서드를 사용자 지정 ID 공급자로 사용할 수 있습니다. 기본적으로 Lambda 함수는 암호로 `myuser` 호출된 단일 사용자를 인증합니

다.`MySuperSecretPassword` 배포 후에는보안 인증 정보를 편집하거나 Lambda 함수 코드를 업데이트하여 다른 작업을 수행할 수 있습니다.

> **Important**
>
> 기본 사용자 및 암호 보안 인증 정보를 편집하는 것이 좋습니다.

스택을 배포한 후에는 CloudFormation 콘솔의 출력 탭에서 스택에 대한 세부 정보를 볼 수 있습니다. 이러한 세부 정보에는 스택의 Amazon 리소스 이름(ARN), 스택이 생성한 IAM 역할의 ARN, 새 게이트웨이의 URL이 포함됩니다.

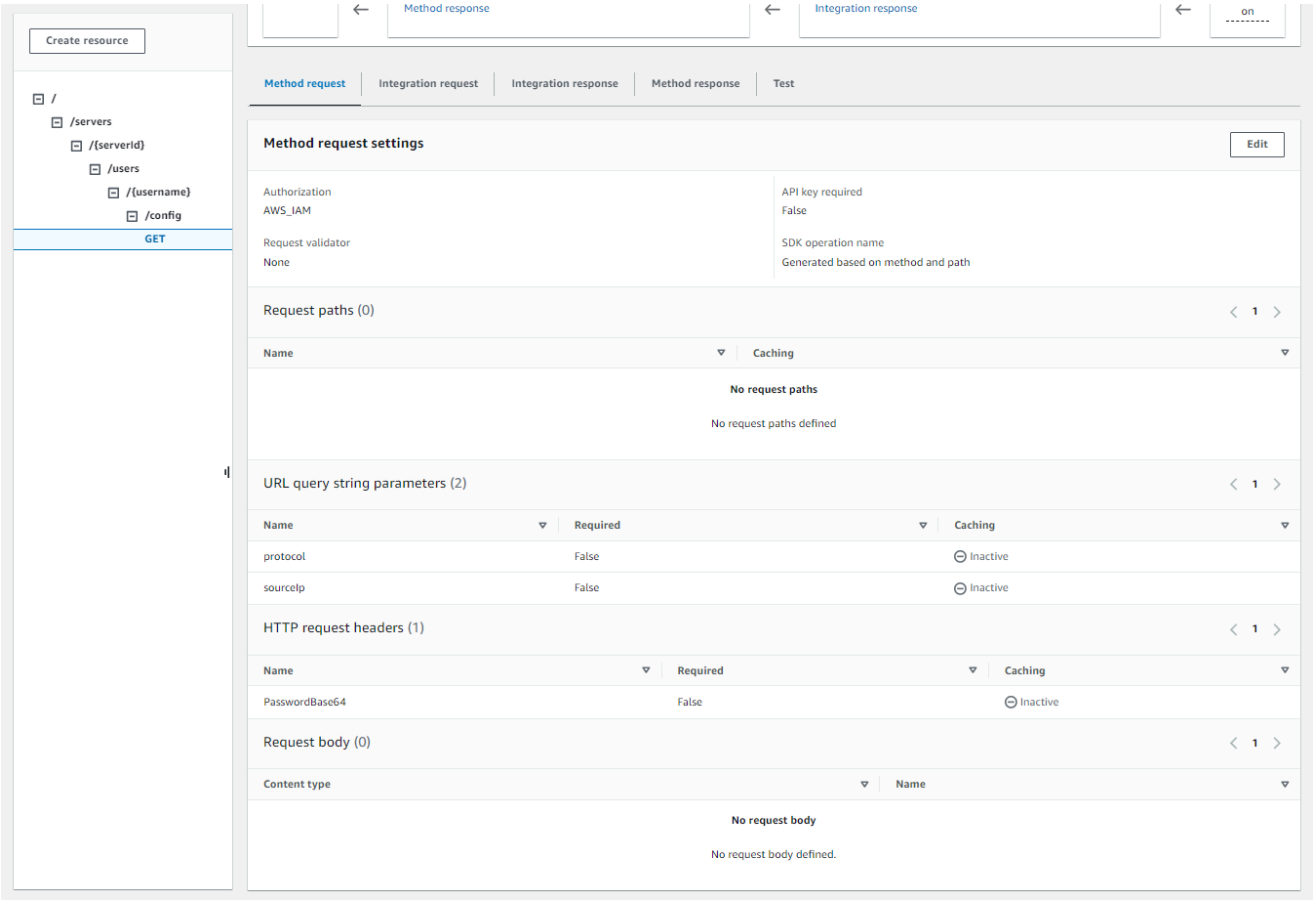
> **Note**
>
> 사용자 지정 ID 공급자 옵션을 사용하여 사용자를 위한 암호 기반 인증을 활성화하고 API Gateway에서 제공하는 요청 및 응답 로깅을 활성화하는 경우, API Gateway는 사용자의 암호를 Amazon Logs에 기록합니다. CloudWatch 프로덕션 환경에서는 이 로그를 사용하지 않는 것이 좋습니다. 자세한 내용은 API Gateway 개발자 안내서의 CloudWatch API Gateway에서 API 로깅 설정을 참조하십시오.

2. 서버의 API Gateway 메서드 구성을 확인합니다. 방법:

   a. https://console.aws.amazon.com/apigateway/에서 Amazon API Gateway 콘솔을 엽니다.

   b. 템플릿에서 생성한 사용자 지정 ID 제공자 이전 기본 AWS CloudFormation 템플릿 API를 선택합니다. 게이트웨이를 보려면 지역을 선택해야 할 수도 있습니다.

   c. 리소스 창에서 GET을 선택합니다. 다음 스크린샷은 올바름 방법 설정을 보여줍니다.

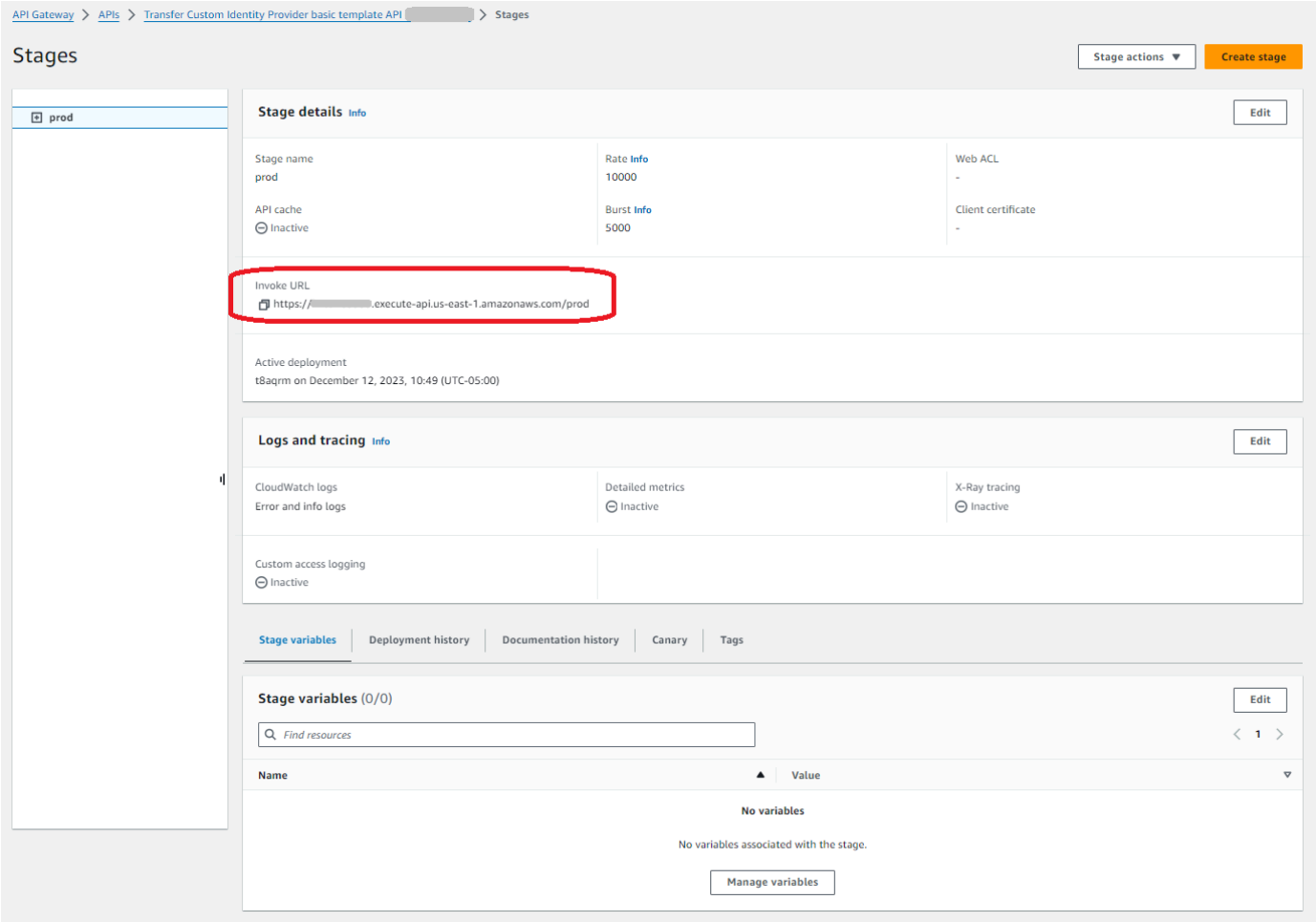


이 지점에서 API Gateway 를 배포할 준비가 완료됩니다.

3. 작업 및 API 배포를 선택합니다. 배포 단계에서 prod를 선택한 다음 배포를 선택합니다.

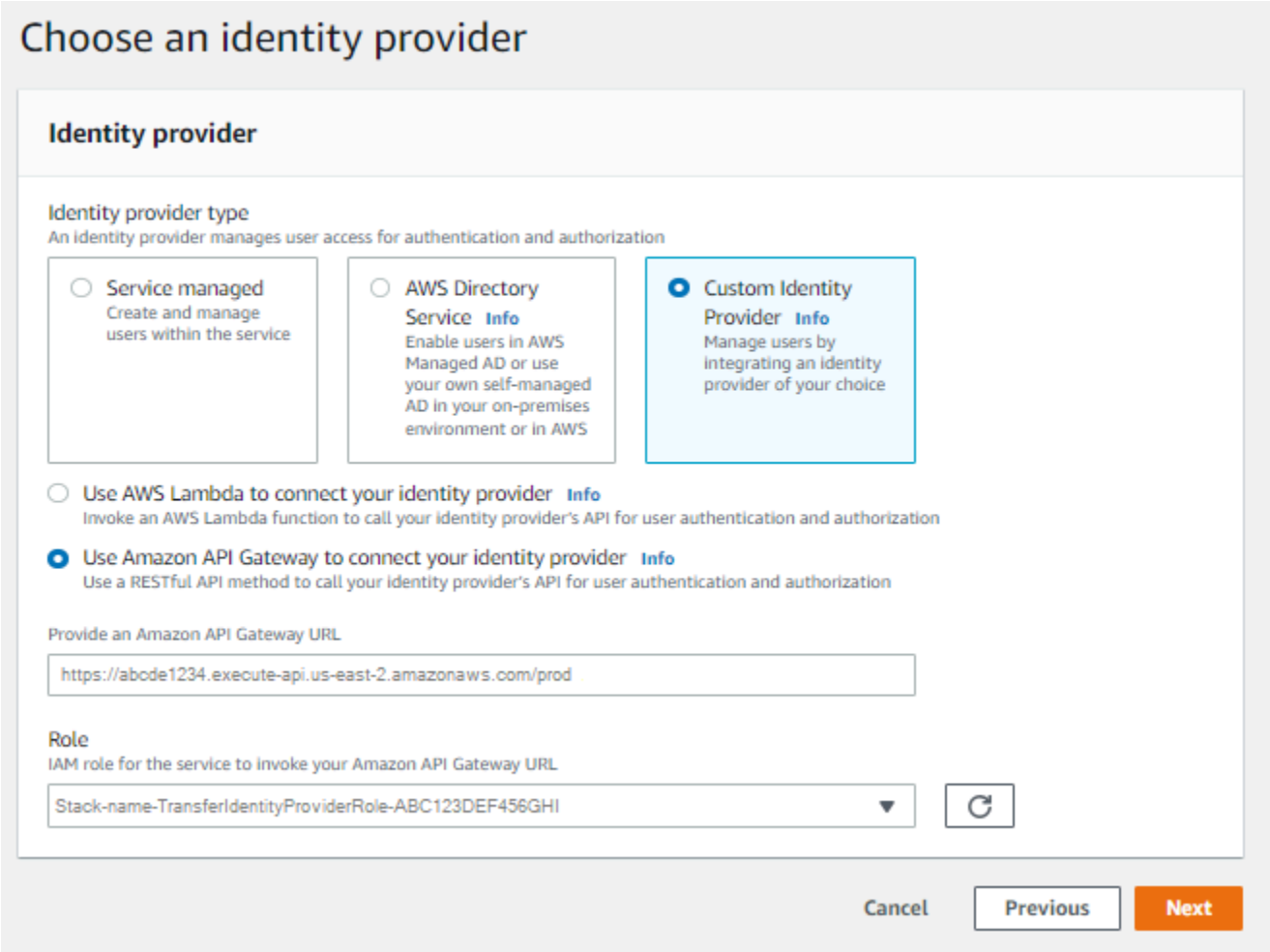
API Gateway 메서드를 성공적으로 배포한 후에는 다음 스크린샷과 같이 단계 > 단계 세부 정보에서 성능을 확인합니다.

> **Note**
>
> 화면 상단에 표시되는 URL 호출 주소를 복사합니다. 다음 단계에 필요할 수 있습니다.



4. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
5. 스택을 생성할 때 Transfer Family가 자동으로 생성되었어야 합니다. 그렇지 않은 경우 다음 단계를 사용하여 서버를 구성하십시오.

   a. 서버 생성을 선택하여 서버 생성 페이지를 엽니다. ID 공급자 선택에서 사용자 지정을 선택한 다음 Amazon API Gateway를 선택하여 다음 스크린샷과 같이 ID 공급자에 연결합니다.



b. Amazon API Gateway URL 제공 텍스트 상자에 이 절차의 3단계에서 생성한 API Gateway 엔드포인트의 호출 URL 주소를 붙여넣습니다.

c. 역할에서 AWS CloudFormation 템플릿으로 생성한 IAM 역할을 선택합니다. 이 역할을 통해 Transfer Family는 API 게이트웨이 메서드를 간접 호출할 수 있습니다.

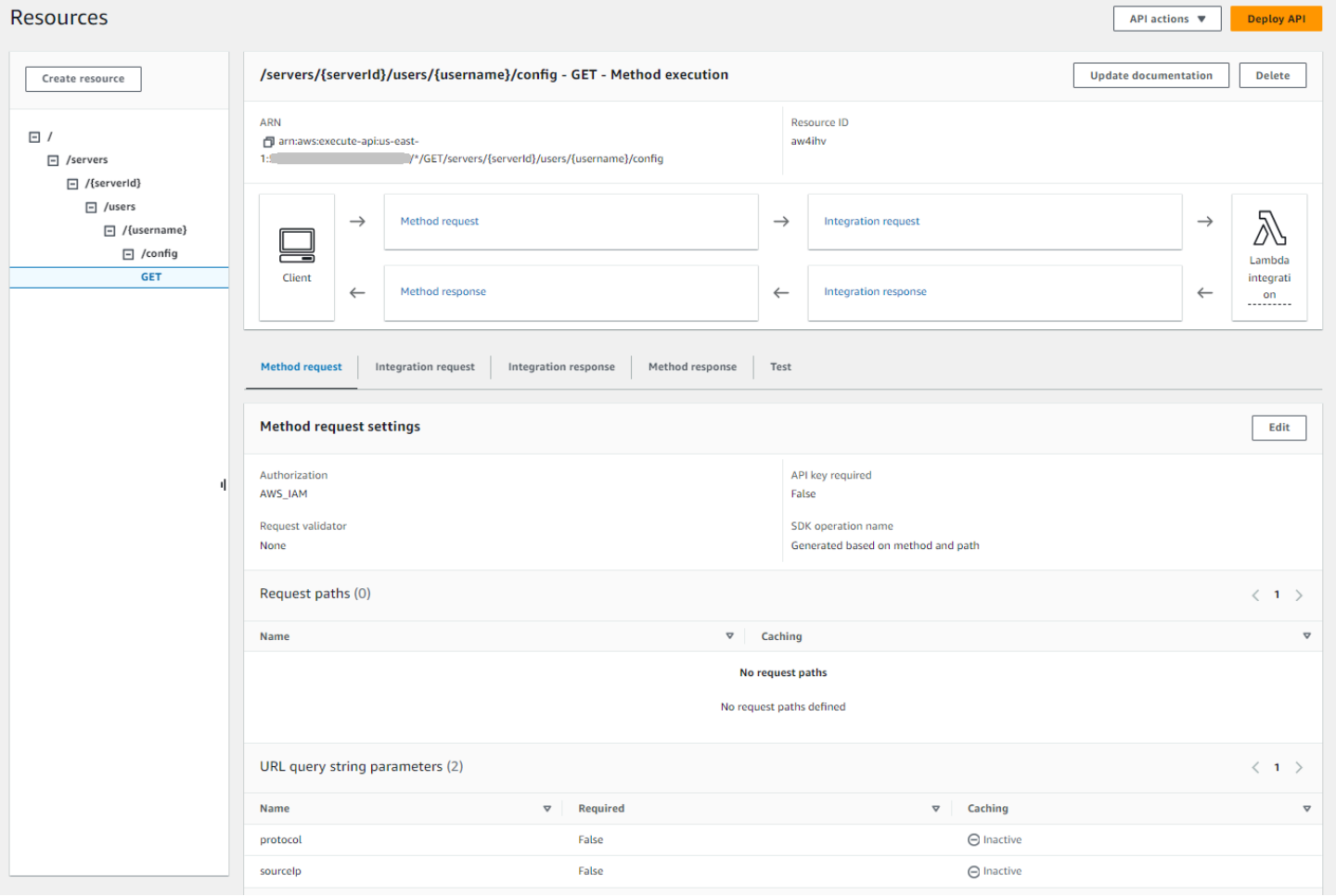
   호출 역할에는 1단계에서 생성한 AWS CloudFormation 스택에 대해 선택한 스택 이름이 포함됩니다. 형식은 다음과 같습니다. *`CloudFormation-stack-name`*-`TransferIdentityProviderRole-`*`ABC123DEF456GHI`*

d. 나머지 상자를 채운 다음 서버 만들기를 선택합니다. 서버를 만들기 위한 나머지 단계에 대한 자세한 내용은 SFTP, FTPS 또는 FTP 서버 엔드포인트 구성을 참조하세요.

API Gateway 메서드 구현

Transfer Family를 위한 사용자 지정 ID 공급자를 생성하려면 API Gateway 메서드에서 리소스 경로가 `/servers/`*`serverId`*`/users/`*`username`*`/config` 인 단일 메서드를 구현해야 합니다.

*serverId*와 *username* 값은 RESTful 리소스 경로에서 얻습니다. 또한 다음 이미지에서 보여지듯이 메서드 요청의 같이 URL 쿼리 문자열 파라미터로서 `sourceIp` 및 `protocol`를 추가합니다.



> **Note**
>
> 이 사용자 이름은 최소 3글자에서 최대 100자여야 합니다. 사용자 이름에는 다음 글자를 사용할 수 있습니다. a~z, A~Z, 0~9, 밑줄(_)과 하이픈(-), 마침표(.)과 골뱅이 사인(@)입니다. 그러나 사용자 이름은 하이픈, 마침표 및 @ 기호로 시작할 수 없습니다.

Transfer Family가 사용자를 대신해 암호 인증을 시도하면, 서비스는 `Password:` 헤더 필드를 공급합니다. `Password:` 헤더가 없는 경우, Transfer Family는 퍼블릭 키 인증을 시도하여 사용자를 인증합니다.

ID 공급자를 사용하여 최종 사용자를 인증하고 권한을 부여하는 경우 자격 증명을 검증하는 것 외에도 최종 사용자가 사용하는 클라이언트의 IP 주소를 기반으로 액세스 요청을 허용하거나 거부할 수 있습니다. 이 기능을 사용하면 S3 버킷 또는 Amazon EFS 파일 시스템에 저장된 데이터가 지원되는 프로토

콜을 통해 신뢰할 수 있는 것으로 지정한 IP 주소에서만 액세스할 수 있습니다. 이 기능을 활성화하려면 `sourceIp`를 쿼리 문자열에 포함해야 합니다.

서버에 여러 프로토콜을 사용하도록 설정한 상태에서 여러 프로토콜에서 동일한 사용자 이름을 사용하여 액세스를 제공하려는 경우 ID 공급자에 각 프로토콜별 자격 증명이 설정되어 있으면 그렇게 할 수 있습니다. 이 기능을 활성화하려면 RESTful 리소스 경로에 *`protocol`* 값을 포함해야 합니다.

API Gateway 메서드는 항상 HTTP 상태 코드를 반환해야 `200` 합니다. 모든 다른 HTTP 상태 코드는 API 액세스하는데 오류를 나타냅니다.

Amazon S3 응답 예

예 응답 본문은 Amazon S3용 다음 형식의 JSON 문서입니다.

```
{
 "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

정책은 JSON을 문자열로 이스케이프합니다. 예:

```
"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\"}]
}"
```

다음 예 응답은 사용자의 홈 디렉터리 타입이 논리적임을 보여줍니다.

```
{
   "Role": "arn:aws:iam::123456789012:role/transfer-access-role-s3",
   "HomeDirectoryType":"LOGICAL",
   "HomeDirectoryDetails":"[{\"Entry\":\"/\",\"Target\":\"/MY-HOME-BUCKET\"}]",
   "PublicKeys":[""]
}
```

Amazon EFS 예 응답

예 응답 본문은 Amazon EFS용 다음 형식의 JSON 문서입니다.

```
{
 "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` 필드는 인증에 성공했음을 나타냅니다. 암호 인증을 할 때(`Password:`헤더를 제공할 때), SSH 퍼블릭 키를 제공할 필요가 없습니다. 사용자를 인증할 수 없는 경우(예: 암호가 잘못된 경우) 메서드는 `Role` 설정되지 않은 응답을 반환해야 합니다. 이러한 응답의 예로는 빈 JSON 객체를 들 수 있습니다.

다음 예 응답은 사용자가 논리적인 홈 디렉터리 타입을 가졌다는 것을 보여줍니다.

```
{
    "Role": "arn:aws:iam::123456789012:role/transfer-access-role-efs",
    "HomeDirectoryType": "LOGICAL",
    "HomeDirectoryDetails":"[{\"Entry\":\"/\",\"Target\":\"/faa1a123\"}]",
    "PublicKeys":[""],
    "PosixProfile":{"Uid":65534,"Gid":65534}
}
```

Lambda 함수에 JSON 형식으로 사용자 정책을 포함할 수 있습니다. Transfer Family의 사용자 정책 구성에 대한 자세한 내용은 단원을 참조하세요액세스 통제 관리.

기본 Lambda 함수

다양한 인증 전략을 구현하려면, 사용자의 게이트웨이 사용하는 Lambda 함수를 편집하세요. 애플리케이션의 요구 사항을 충족하는 데 도움이 되려면 Node.js에서 다음 예 Lambda 함수를 사용할 수 있습니다. 자세한 내용은 개발자 안내서나 AWS Lambda Node.js로 Lambda 함수 구축을 참조하세요.

다음 예 Lambda 함수는 사용자 이름, 암호 (암호 인증을 수행하는 경우), 서버 ID, 프로토콜 및 클라이언트 IP 주소를 가져옵니다. 이러한 입력을 조합하여 ID 공급자를 조회하고 로그인을 허용할지 여부를 결정할 수 있습니다.

> **Note**
>
> 서버에 여러 프로토콜을 사용하도록 설정한 상태에서 여러 프로토콜에서 동일한 사용자 이름을 사용하여 액세스를 제공하려는 경우 ID 공급자에 각 프로토콜별 보안 인증이 설정되어 있으면 그렇게 할 수 있습니다.
> FTP (파일 전송 프로토콜)의 경우, Secure Shell(SSH) SFTP (파일 전송 프로토콜) 및 SSL을 통한 파일 전송 프로토콜 (FTPS)의 분리된 보아 인증을 유지하는 것이 좋습니다. SFTP 및 FTPS와 달리 FTP는 자격 증명을 일반 텍스트로 전송하므로 FTP에 대해 별도의 보안 인증을 유지하는 것이 좋습니다. FTP 자격 증명을 SFTP 또는 FTPS에서 분리하면 FTP 자격 증명이 공유되거나 노출되더라도 SFTP 또는 FTPS를 사용하는 워크로드의 보안을 유지할 수 있습니다.

이 예 함수는 퍼블릭 키 인증을 수행하는 경우 퍼블릭 키와 함께 역할 및 논리적 홈 디렉터리 세부 정보를 반환합니다.

서비스 관리 사용자를 생성할 때는 해당 사용자의 홈 디렉터리를 논리적 또는 물리적 디렉터리로 설정합니다. 마찬가지로 원하는 사용자의 물리적 또는 논리적 디렉터리 구조를 전달하려면 Lambda 함수 결과가 필요합니다. 설정하는 파라미터는 HomeDirectoryType필드 값에 따라 달라집니다.

- `HomeDirectoryType`을 `PATH`로 설정 — `HomeDirectory` 필드는 사용자에게 표시되는 절대 Amazon S3 버킷 접두사 또는 Amazon EFS 절대 경로여야 합니다.
- `HomeDirectoryType`을 `LOGICAL`로 설정 — `HomeDirectory` 필드를 설정하지 마세요. 대신 서비스 관리 사용자를 위한 HomeDirectoryDetails매개 변수에 설명된 값과 유사하게 원하는 항목/대상 매핑을 제공하는 `HomeDirectoryDetails` 필드를 설정합니다.

예 함수는 예 Lambda 함수에 나열되어 있습니다.

함께 사용하기 위한 Lambda 함수 AWS Secrets Manager

ID AWS Secrets Manager 공급자로 사용하려면 샘플 템플릿에서 Lambda 함수를 사용하면 됩니다. AWS CloudFormation Lambda 함수는 사용자 자격 증명으로 Secrets Manager 서비스를 쿼리하고, 성공하면 지정된 시크릿을 반환합니다. Secrets Manager 사용에 대한 자세한 내용은 AWS Secrets Manager 사용 설명서를 참조하세요.

이 Lambda 함수를 사용하는 샘플 AWS CloudFormation 템플릿을 다운로드하려면 에서 제공하는 Amazon S3 버킷으로 이동하십시오. AWS Transfer Family

템플릿 개선 사항 AWS CloudFormation

게시된 CloudFormation 템플릿의 API Gateway 인터페이스가 개선되었습니다. 이제 템플릿은 API Gateway와 함께 Base64로 인코딩된 암호를 사용합니다. 기존 배포는 이러한 개선 사항이 없어도 계속 작동하지만 기본 US-ASCII 문자 집합 이외의 문자를 포함하는 암호는 허용되지 않습니다.

이 기능을 활성화하는 템플릿의 변경 사항은 다음과 같습니다.

- `GetUserConfigRequest AWS::ApiGateway::Method`리소스에 이 `RequestTemplates` 코드가 있어야 합니다 (기울임꼴로 표시된 줄은 업데이트된 줄임).

```
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')"
}
```

- PasswordBase64헤더를 사용하려면 GetUserConfig 리소스의 내용을 변경해야 합니다 (기울임꼴로 표시된 줄은 업데이트된 줄임). RequestParameters

```
RequestParameters:
   method.request.header.PasswordBase64: false
   method.request.querystring.protocol: false
   method.request.querystring.sourceIp: false
```

스택용 템플릿이 최신인지 확인하려면

1. https://console.aws.amazon.com/cloudformation 에서 AWS CloudFormation 콘솔을 엽니다.
2. 스택 목록에서 스택을 선택합니다.
3. 디테일 패널에서 템플릿 탭을 선택합니다.
4. 다음을 찾아보세요.

   - RequestTemplates검색해서 다음 줄이 있는지 확인하세요.

   ```
   "password":
    "$util.escapeJavaScript($util.base64Decode($input.params('PasswordBase64'))).replaceAll(
   \'","''")",
   ```

   - RequestParameters검색해서 다음 줄이 있는지 확인하세요.

   ```
   method.request.header.PasswordBase64: false
   ```

업데이트된 라인이 보이지 않으면 스택을 편집하세요. AWS CloudFormation 스택을 업데이트하는 방법에 대한 자세한 내용은 AWS CloudFormation; 사용 설명서의 스택 템플릿 수정을 참조하십시오.

논리적 디렉터리를 사용하여 Transfer Family 디렉터리 구조를 단순화합니다.

AWS Transfer Family 서버 디렉터리 구조를 단순화하기 위해 논리적 디렉터리를 사용할 수 있습니다. 논리적 디렉터리를 사용하면 Amazon S3 버킷 또는 Amazon EFS 파일 시스템에 연결할 때 사용자가

탐색할 수 있는 사용자 친화적인 명칭을 사용하는 가상 디렉터리 구조를 만들 수 있습니다. 논리적 디렉터리를 사용하면 절대 디렉터리 경로, Amazon S3 버킷 명칭 및 EFS 파일 시스템 명칭을 최종 사용자에게 공개하지 않아도 됩니다.

> **Note**
>
> 최종 사용자가 허용한 작업만 수행할 수 있도록 세션 정책을 사용해야 합니다.
> 논리적 디렉터리를 사용하여 최종 사용자를 위한 사용자 친화적인 가상 디렉터리를 만들고 버킷 이름을 추상화해야 합니다. 논리적 디렉터리 매핑은 사용자가 지정된 논리적 경로와 하위 디렉터리에만 액세스할 수 있도록 허용하고 논리적 루트를 통과하는 상대 경로는 금지합니다. Transfer Family는 상대 요소를 포함할 수 있는 모든 경로를 검증하고 이러한 경로를 Amazon S3로 전달하기 전에 해당 경로가 확인되지 않도록 적극적으로 차단합니다. 이렇게 하면 사용자가 논리적 매핑을 넘어서는 것을 방지할 수 있습니다.
> Transfer Family를 사용하면 최종 사용자가 논리적 디렉터리 외부의 디렉터리에 액세스할 수 없지만 고유한 역할 또는 세션 정책을 사용하여 스토리지 수준에서 최소 권한을 적용하는 것이 좋습니다.

논리적 디렉터리를 사용하여 chroot 작업을 수행하여 스토리지 계층 내에서 사용자의 루트 디렉터리를 원하는 위치로 설정할 수 있습니다. 이 모드에서는 사용자가 구성한 홈 또는 루트 디렉터리 외부의 디렉터리를 탐색할 수 없습니다.

예를 들어 Amazon S3 사용자는 */mybucket/home/*`${transfer:UserName}`까지만 액세스하도록 제한되었지만 일부 클라이언트는 사용자가 폴더를 */mybucket/home*로 이동할 수 있도록 허용합니다. 이 경우 사용자는 Transfer Family 서버에서 로그아웃했다가 다시 로그인한 후에야 의도한 홈 디렉터리로 돌아갑니다. chroot 작업을 수행하면 이러한 상황이 발생하는 것을 방지할 수 있습니다.

버킷 및 접두사 전체에 고유한 디렉터리 구조를 만들 수 있습니다. 이 기능은 버킷 접두사를 통해 복제할 수 없는 특정 디렉터리 구조가 필요한 워크플로가 있는 경우에 유용합니다. 디렉터리 경로가 파일 시스템의 다른 위치를 참조하는 Linux 파일 시스템에 심볼릭 링크를 생성하는 것과 비슷하게 Amazon S3 내의 여러 비연속 위치에 연결할 수도 있습니다.

> **논리적 디렉터리 파일 매핑**
>
> 이제 `HomeDirectoryMapEntry` 데이터 유형에 파라미터가 포함됩니다. `Type` 이 매개 변수가 존재하기 전에는 대상이 파일인 논리적 디렉터리 매핑을 만들 수 있었습니다. 이전에 이러한 종류의 논리적 디렉터리 매핑을 만든 적이 있는 경우 이를 명시적으로 로 설정해야 합니다. 그렇지 않으면 앞으로 매핑이 제대로 작동하지 않을 수 있습니다. `Type FILE`

이 작업을 수행하는 한 가지 방법은 UpdateUser API를 호출하고 기존 매핑에 대해 를 로 설정하는 것입니다. Type FILE

논리적 디렉터리 사용 규칙

논리적 디렉터리 매핑을 구축하기 전에 다음 규칙을 이해해야 합니다.

- Entry이(가) "/"인 경우 경로가 겹칠 수 없으므로 매핑을 하나만 사용할 수 있습니다.
- 논리적 디렉터리는 최대 2.1MB의 매핑을 지원합니다 (서비스 관리 사용자의 경우 이 제한은 항목 2,000개). 즉, 매핑이 포함된 데이터 구조의 최대 크기는 2.1MB입니다. 매핑이 많은 경우 다음과 같이 매핑 크기를 계산할 수 있습니다.
  1. 일반적인 매핑을 사용할 실제 값의 *entry-path* 위치와 *target-path* 위치 {"Entry":"/*entry-path*","Target":"/*target-path*"} 등의 형식으로 작성하십시오.
  2. 해당 문자열의 문자 수를 세고 1을 더하십시오.
  3. 이 숫자에 서버에 대해 가지고 있는 대략적인 매핑 수를 곱하십시오.

  3단계에서 예상한 수가 2.1MB 미만이면 매핑이 허용 가능한 한도 내에 있는 것입니다.
- 버킷 또는 파일 시스템 경로가 사용자 이름을 기반으로 파라미터화된 경우 타겟은 ${transfer:UserName} 변수를 사용할 수 있습니다.
- 대상은 다른 버킷 또는 파일 시스템의 경로일 수 있지만 매핑된 AWS Identity and Access Management (IAM) 역할 (응답의 Role 매개변수) 이 해당 버킷 또는 파일 시스템에 대한 액세스를 제공하는지 확인해야 합니다.
- HomeDirectory파라미터 값을 사용할 때 Entry Target 두 쌍이 이 값을 암시하므로 파라미터를 지정하지 마세요. LOGICAL HomeDirectoryType
- 대상은 순방향 슬래시 (/) 문자로 시작해야 하지만 를 지정할 때는 후행 슬래시 (/) 를 사용하지 마십시오. Target 예를 들어 /DOC-EXAMPLE-BUCKET/images 는 허용되지만 DOC-EXAMPLE-BUCKET/images 허용되지는 않습니다. /DOC-EXAMPLE-BUCKET/images/
- Amazon S3는 객체 저장소입니다. 즉, 폴더는 가상 개념이며 실제 디렉터리 계층 구조가 없습니다. 애플리케이션이 클라이언트로부터 stat 작업을 실행하는 경우 Amazon S3를 스토리지로 사용하면 모든 것이 파일로 분류됩니다. 이 동작은 Amazon 심플 스토리지 서비스 사용 설명서의 폴더를 사용하여 Amazon S3 콘솔에서 객체를 구성하는 방법에 설명되어 있습니다. 애플리케이션에서 어떤 것이 파일인지 폴더인지 stat 정확하게 표시해야 하는 경우, Amazon Elastic File System (Amazon EFS) 을 Transfer Family 서버의 스토리지 옵션으로 사용할 수 있습니다.

- 사용자의 논리적 디렉터리 값을 지정하는 경우 사용하는 파라미터는 사용자 유형에 따라 달라집니다.
  - 서비스 관리 사용자의 경우 HomeDirectoryMappings에서 논리적 디렉터리 값을 제공하세요.
  - 사용자 지정 ID 제공자 사용자의 경우 에서 논리적 디렉터리 값을 제공하십시오HomeDirectoryDetails.

> **Important**
>
> Amazon S3 디렉터리의 성능을 최적화하도록 선택하지 않는 한 (서버를 생성하거나 업데이트할 때) 시작 시 루트 디렉터리가 있어야 합니다. Amazon S3의 경우 이는 순방향 슬래시 (/) 로 끝나는 0바이트 객체를 이미 생성해야 루트 폴더를 생성할 수 있음을 의미합니다. 이 문제를 피하는 것이 Amazon S3 성능 최적화를 고려해야 하는 이유입니다.

논리적 디렉터리 구현 및 chroot

논리적 디렉터리와 chroot 기능을 사용하려면 다음을 수행해야 합니다:

각 사용자에 대해 논리적 디렉터리를 활성화하세요. 사용자를 만들거나 업데이트할 때 HomeDirectoryType 파라미터를 LOGICAL로 설정하면 됩니다.

```
"HomeDirectoryType": "LOGICAL"
```

chroot

chroot의 경우, 각 사용자에 대한 단일 Entry 및 Target 쌍으로 구성된 디렉터리 구조를 만드세요. 루트 폴더는 Entry 지점이고 매핑할 버킷 또는 파일 시스템의 Target 위치입니다.

Example for Amazon S3

```
[{"Entry": "/", "Target": "/mybucket/jane"}]
```

Example for Amazon EFS

```
[{"Entry": "/", "Target": "/fs-faa1a123/jane"}]
```

이전 예와 같이 절대 경로를 사용하거나, 다음 예와 같이 사용자 이름을 동적으로 `${transfer:UserName}`로 대체할 수 있습니다.

```
[{"Entry": "/", "Target":
"/mybucket/${transfer:UserName}"}]
```

위 예에서 사용자는 루트 디렉터리에 잠겨 있어 계층 구조에서 상위 위치로 이동할 수 없습니다.

가상 디렉터리 구조

가상 디렉터리 구조의 경우, 사용자의 IAM 역할 매핑에 액세스 권한이 있는 한, 여러 버킷 또는 파일 시스템을 포함하여 S3 버킷 또는 EFS 파일 시스템 어디에나 대상을 사용하여 여러 `Entry` `Target` 쌍을 생성할 수 있습니다.

다음 가상 구조 예제에서 사용자는 AWS SFTP에 로그인할 때,, 및 하위 디렉터리가 있는 루트 디렉터리에 있습니다. `/pics` `/doc` `/reporting` `/anotherpath/subpath/financials`

> **Note**
>
> Amazon S3 디렉터리의 성능을 최적화하도록 선택하지 않는 한 (서버를 생성하거나 업데이트할 때), 디렉터리가 아직 존재하지 않는 경우 사용자나 관리자가 해당 디렉터리를 생성해야 합니다. 이 문제를 피하는 것이 Amazon S3 성능 최적화를 고려해야 하는 이유입니다.
> Amazon EFS의 경우 여전히 관리자가 논리적 매핑 또는 디렉터리를 생성해야 합니다. /

```
[
{"Entry": "/pics", "Target": "/bucket1/pics"},
{"Entry": "/doc", "Target": "/bucket1/anotherpath/docs"},
{"Entry": "/reporting", "Target": "/reportingbucket/Q1"},
{"Entry": "/anotherpath/subpath/financials", "Target": "/reportingbucket/financials"}]
```

> **Note**
>
> 매핑한 특정 폴더에만 파일을 업로드할 수 있습니다. 즉, 이전 예에서는 `/anotherpath` 또는 `anotherpath/subpath` 디렉터리에는 업로드할 수 없고; `anotherpath/subpath/financials`에만 업로드할 수 있습니다. 또한 경로가 겹칠 수 없으므로 이러한 경로에 직접 매핑할 수 없습니다.
> 예를 들어, 다음 매핑을 형성한다고 가정하겠습니다:

```
{
   "Entry": "/pics",
   "Target": "/mybucket/pics"
},
{
   "Entry": "/doc",
   "Target": "/mybucket/mydocs"
},
{
   "Entry": "/temp",
   "Target": "/mybucket"
}
```

해당 버킷에만 파일을 업로드할 수 있습니다. `sftp`를 통해 처음 연결하면 루트 디렉터리인 `/`(으)로 이동됩니다. 해당 디렉터리에 파일을 업로드하려고 하면 업로드가 실패합니다. 다음 명령은 예 순서를 나타냅니다:

```
sftp> pwd
Remote working directory: /
sftp> put file
Uploading file to /file
remote open("/file"): No such file or directory
```

임의의 `directory/sub-directory` 위치에 업로드하려면 경로를 `sub-directory`에 명시적으로 매핑해야 합니다.

다운로드하여 사용할 수 있는 AWS CloudFormation 템플릿을 포함하여 논리적 디렉터리 구성 및 사용자를 chroot 위한 자세한 내용은 스토리지 블로그의 chroot 및 논리적 디렉터리를 사용한 AWS SFTP 구조 단순화를 참조하십시오. AWS

논리적 디렉터리 구성 예

이 예에서는 사용자를 생성하고 두 개의 논리적 디렉터리를 할당합니다. 다음 명령을 실행하면 (기존 Transfer Family 서버의 경우) 논리적 디렉터리 `pics` 및 `doc` 를 사용하여 새 사용자가 만들어집니다.

```
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
```

marymajor가 기존 사용자이고 홈 디렉터리 타입이 PATH인 경우 이전 사용자와 비슷한 명령을 사용하여 그것을 LOGICAL로 변경할 수 있습니다.

```
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\"}]"
```

유념할 사항:

- 디렉터리 /DOC-EXAMPLE-BUCKET1/pics 및 /DOC-EXAMPLE-BUCKET2/test/mydocs 없는 경우 사용자 (또는 관리자) 가 해당 디렉터리를 만들어야 합니다.
- **marymajor**이 서버에 연결되어 ls -l 명령을 실행하면 다음과 같은 내용이 표시됩니다:

  ```
  drwxr--r--   1        -        -        0 Mar 17 15:42 doc
  drwxr--r--   1        -        -        0 Mar 17 16:04 pics
  ```

- **marymajor**은(는) 이 수준에서는 파일이나 디렉터리를 만들 수 없습니다. 하지만 pics 및 doc 내에서 하위 디렉토리를 추가할 수는 있습니다.
- pics 및 doc에 추가되는 파일은 각각 Amazon S3 경로 /DOC-EXAMPLE-BUCKET1/pics 및 /DOC-EXAMPLE-BUCKET2/test/mydocs에 추가됩니다.
- 이 예에서는 이러한 가능성을 설명하기 위해 서로 다른 두 개의 버킷을 지정합니다. 하지만 사용자에 대해 지정한 여러 논리 디렉터리 또는 모든 논리적 디렉터리에 동일한 버킷을 사용할 수 있습니다.

Amazon EFS를 위한 논리적 디렉터리 구성

Transfer Family 서버에서 Amazon EFS를 사용하는 경우, 사용자가 논리적 홈 디렉터리에서 작업할 수 있으려면 먼저 읽기 및 쓰기 권한이 있는 사용자의 홈 디렉터리를 생성해야 합니다. 사용자는 논리적 홈 디렉터리 상의 mkdir에 대한 권한이 없기 때문에 이 디렉터리를 직접 만들 수는 없습니다.

사용자의 홈 디렉터리가 존재하지 않고 사용자가 ls 명령을 실행하면 시스템은 다음과 같이 응답합니다:

```
sftp> ls
remote readdir ("/"): No such file or directory
```

상위 디렉터리에 대한 관리 액세스 권한이 있는 사용자는 사용자의 논리적 홈 디렉터리를 생성해야 합니다.

사용자 지정 응답 AWS Lambda

맞춤 ID 공급자에 연결하는 Lambda 함수와 함께 논리적 디렉터리를 사용할 수 있습니다. 이렇게 하려면 Lambda 함수에서 `HomeDirectoryType`를 **`LOGICAL`**(으)로 지정하고, `HomeDirectoryDetails` 파라미터에 `Entry`와(과) `Target`를 추가합니다. 예:

```
HomeDirectoryType: "LOGICAL"
HomeDirectoryDetails: "[{\"Entry\": \"/\", \"Target\": \"/DOC-EXAMPLE-BUCKET/
theRealFolder"}]"
```

다음 코드는 맞춤 Lambda 인증 호출로부터의 응답 성공의 예입니다.

```
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**
>
> 이 `"Url":` 행은 API Gateway 메서드를 맞춤 ID 공급자로 사용하는 경우에만 반환됩니다.

AWS Transfer Family SFTP 커넥터

AWS Transfer Family SFTP 커넥터는 SFTP 프로토콜을 사용하여 Amazon 스토리지와 외부 파트너 간에 파일 및 메시지를 전송하기 위한 관계를 설정합니다. Amazon S3에서 파트너 소유의 외부 대상으로 파일을 보낼 수 있습니다. SFTP 커넥터를 사용하여 파트너의 SFTP 서버에서 파일을 검색할 수도 있습니다.

> **Note**
>
> 현재 SFTP 커넥터는 인터넷 액세스 가능 엔드포인트를 제공하는 원격 SFTP 서버에 연결하는 데만 사용할 수 있습니다.

다음 블로그 게시물은 SFTP 커넥터를 사용하여 원격 SFTP 서버로 파일을 전송하기 전에 PGP를 사용하여 파일을 암호화하는 것을 포함하여 SFTP 커넥터를 사용하여 MFT 워크플로를 구축하는 참조 아키텍처를 제공합니다. SFTP 커넥터 및 PGP 암호화로 안전하고 규정을 준수하는 관리형 파일 전송 설계 AWS Transfer Family

Transfer Family AWS Transfer Family SFTP 커넥터에 대한 간략한 소개는 SFTP 커넥터를 참조하세요.

주제

- SFTP 커넥터 구성
- SFTP 커넥터를 사용하여 파일을 보내고 검색합니다.
- 원격 디렉터리 콘텐츠 목록
- SFTP 커넥터 관리

SFTP 커넥터 구성

이 항목에서는 SFTP 커넥터를 만드는 방법, 커넥터와 관련된 보안 알고리즘, 자격 증명을 보관할 암호를 저장하는 방법, 개인 키 형식 지정에 대한 세부 정보, 커넥터 테스트 지침을 설명합니다.

주제

- SFTP 생성
- SFTP 커넥터와 함께 사용할 암호를 저장합니다.

- SFTP 커넥터 개인 키 생성 및 형식 지정
- SFTP 커넥터를 테스트합니다.

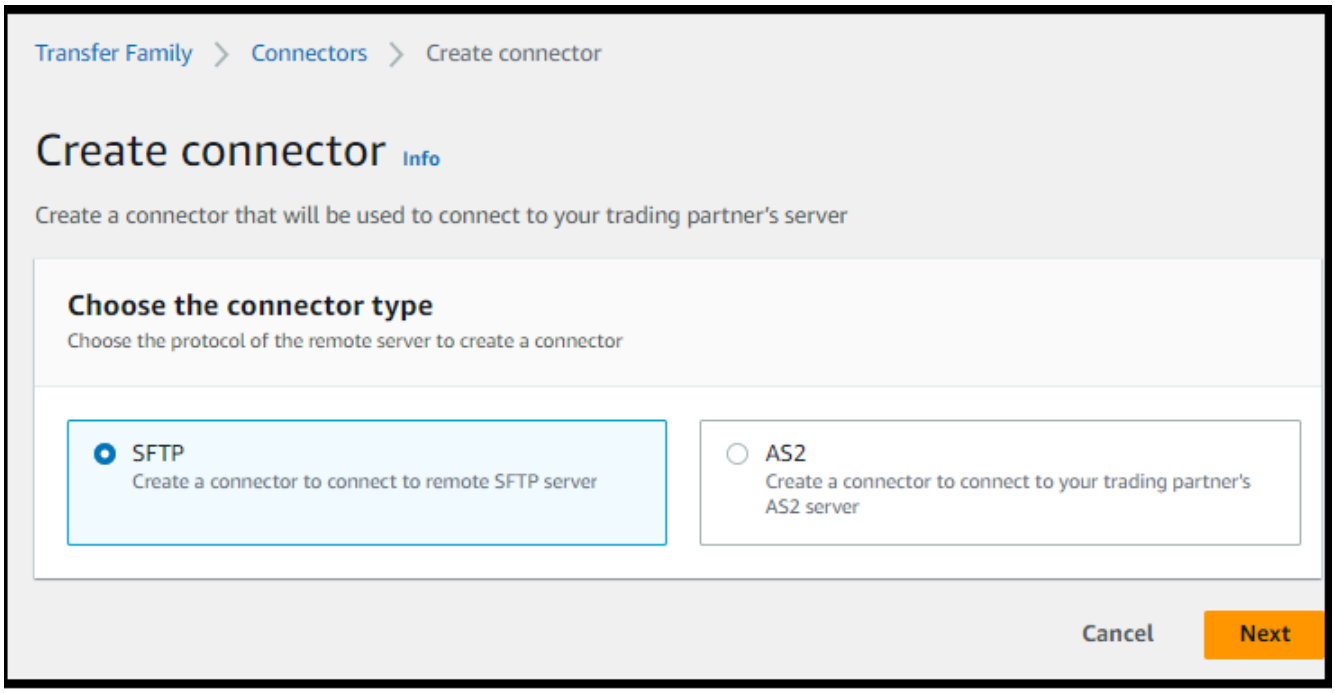
SFTP 생성

이 절차에서는 AWS Transfer Family 콘솔 또는 를 사용하여 SFTP 커넥터를 만드는 방법을 설명합니다. AWS CLI

Console

SFTP 커넥터를 생성하려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 커넥터를 선택한 다음 커넥터 생성을 선택합니다.
3. 커넥터 타입으로 SFTP를 선택하여 SFTP 커넥터를 생성한 후 다음을 선택합니다.



4. 커넥터 구성 섹션에서 다음 정보를 제공합니다:
   - URL에는 원격 SFTP 서버의 URL을 입력합니다. 이 URL은 `sftp://`*`partner-SFTP-server-url`*와 같은 형식(예: `sftp://AnyCompany.com`)이어야 합니다.

> **Note**
>
> 경우에 따라 URL에 포트 번호를 제공할 수 있습니다. 형식은 `sftp://partner-SFTP-server-url:port-number`입니다. 기본 포트 번호(지정된 포트가 없는 경우)는 포트 22입니다.

- 액세스 역할에서 사용할 (IAM) 역할의 Amazon 리소스 이름 AWS Identity and Access Management (ARN) 을 선택합니다.
  - 이 역할이 `StartFileTransfer` 요청에서 사용된 파일 위치의 상위 디렉터리에 대한 읽기 및 쓰기 액세스 권한을 이 역할이 제공하는지 확인하세요.
  - 이 역할이 `secretsmanager:GetSecretValue`에 대해 암호에 액세스할 수 있는 권한을 제공하는지 확인하세요.

    > **Note**
    >
    > 정책에서 비밀번호의 ARN을 지정해야 합니다. ARN에는 비밀 이름이 포함되지만 이름은 무작위 영숫자 6자로 추가됩니다. 시크릿에 대한 ARN의 형식은 다음과 같습니다.
    >
    > ```
    > arn:aws:secretsmanager:region:account-id:secret:aws/
    > transfer/SecretName-6RandomCharacters
    > ```

  - 사용자의 이전 요청을 처리할 때 커넥터가 리소스에 액세스할 수 있도록 허용하는 신뢰 관계가 이 역할에 포함되어 있는지 확인하세요. 신뢰 관계 설정에 대한 자세한 내용은 신뢰 관계를 구축하기 위해을 참조하세요.

다음 예제는 Amazon S3의 *DOC-EXAMPLE-BUCKET#* 액세스하는 데 필요한 권한을 부여하고 Secrets Manager에 저장된 지정된 시크릿을 부여합니다.

```
{
  "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

액세스 역할의 경우, 예는 단일 암호에 대한 액세스 권한을 부여합니다. 하지만 와일드카드 문자를 사용하면 여러 사용자 및 암호에 대해 동일한 IAM 역할을 재사용하려는 경우 작업을 줄일 수 있습니다. 예컨대, 다음 리소스 명령문은 aws/transfer으로 시작하는 이름을 가진 모든 암호에 대한 권한을 부여합니다.

```
"Resource": "arn:aws:secretsmanager:region:account-id:secret:aws/
transfer/*"
```

SFTP 자격 증명이 포함된 암호를 다른 AWS 계정에 저장할 수도 있습니다. 계정 간 보안 액세스를 활성화하는 방법에 대한 자세한 내용은 다른 계정의 사용자에 대한 보안 권한 AWS Secrets Manager 섹션을 참조하십시오.

- (선택 사항) Logging 역할의 경우 커넥터가 이벤트를 로그로 푸시하는 데 사용할 IAM 역할을 선택합니다. CloudWatch 다음 예제 정책은 SFTP 커넥터의 이벤트를 기록하는 데 필요한 권한을 나열합니다.

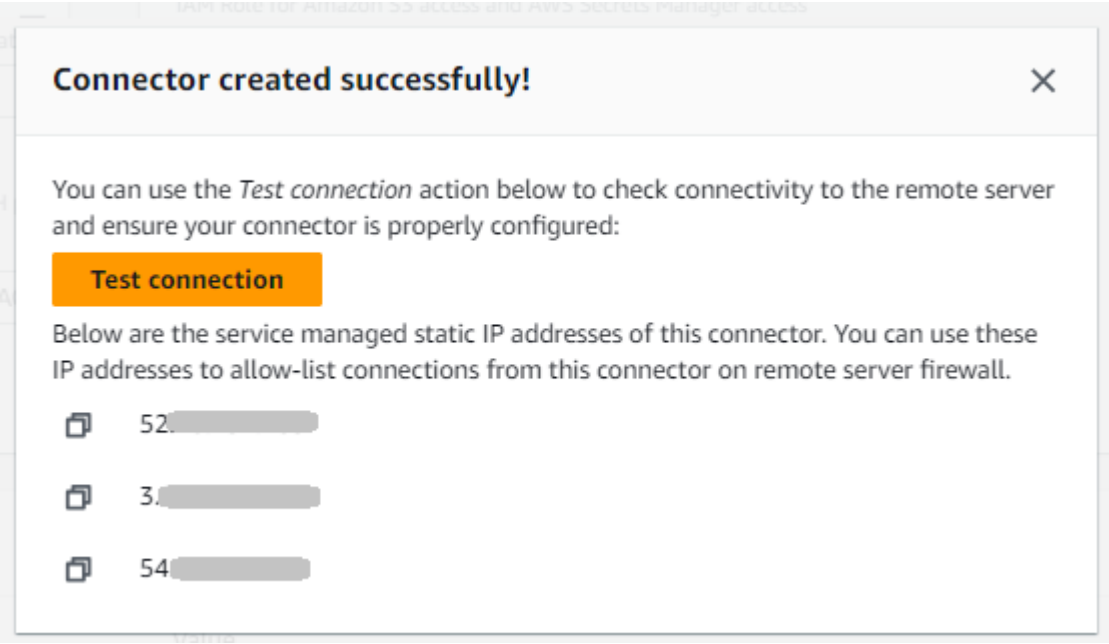
```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "SFTPConnectorPermissions",
        "Effect": "Allow",
        "Action": [
            "logs:CreateLogStream",
            "logs:DescribeLogStreams",
            "logs:CreateLogGroup",
            "logs:PutLogEvents"
        ],
        "Resource": [
            "arn:aws:logs:*:*:log-group:/aws/transfer/*"
        ]
    }]
}
```

5. SFTP 구성 섹션에 다음 정보를 제공합니다.

   - 커넥터 자격 증명의 경우 드롭다운 목록에서 SFTP 사용자의 개인 키 또는 AWS Secrets Manager 암호가 포함된 암호의 이름을 선택합니다. 암호를 생성하여 특정 방식으로 저장해야 합니다. 자세한 내용은 SFTP 커넥터와 함께 사용할 암호를 저장합니다.를 참조하세요.
   - 신뢰할 수 있는 호스트 키의 경우 외부 서버를 식별하는 데 사용되는 호스트 키의 공개 부분을 붙여넣습니다. 신뢰 호스트 키 추가를 선택하여 키를 더 추가하면 키를 두 개 이상 추가할 수 있습니다. SFTP 서버에 대해 `ssh-keyscan` 명령을 사용하여 필요한 키를 검색할 수 있습니다. Transfer Family가 지원하는 신뢰 호스트 키의 형식 및 타입에 대한 자세한 설명은 SFTPConnectorConfig 섹션을 참조하세요.

6. 암호화 알고리즘 옵션 섹션에서 보안 정책 필드의 드롭다운 목록에서 보안 정책을 선택합니다. 보안 정책을 통해 커넥터가 지원하는 암호화 알고리즘을 선택할 수 있습니다. 사용 가능한 보안 정책 및 알고리즘에 대한 자세한 내용은 을 참조하십시오SFTP AWS Transfer Family 커넥터에 대한 보안 정책.

7. (옵션) 태그 섹션에서 키 및 값에 하나 이상의 태그를 키-값 쌍을 입력합니다.
8. 모든 설정을 확인한 후 커넥터 생성을 선택하여 SFTP 커넥터를 생성합니다. 커넥터가 성공적으로 생성되면 할당된 고정 IP 주소 목록과 연결 테스트 버튼이 있는 화면이 나타납니다. 버튼을 사용하여 새 커넥터의 구성을 테스트할 수 있습니다.



새 SFTP 커넥터의 ID가 목록에 추가된 커넥터 페이지가 나타납니다. 커넥터에 대한 세부 정보를 보려면 SFTP 커넥터 세부 정보 보기 섹션을 참조하세요.

CLI

커넥터를 생성할 때는 create-connector 명령을 사용합니다. 이 명령을 사용하여 SFTP 커넥터를 생성하려면 다음 정보를 제공해야 합니다.

- 원격 SFTP 서버의 URL입니다. 이 URL은 `sftp://`*`partner-SFTP-server-url`*와 같은 형식(예: `sftp://AnyCompany.com`)이어야 합니다.
- 액세스 역할입니다. 사용할 AWS Identity and Access Management (IAM) 역할의 Amazon 리소스 이름(ARN)을 선택합니다.
  - 이 역할이 `StartFileTransfer` 요청에서 사용된 파일 위치의 상위 디렉터리에 대한 읽기 및 쓰기 액세스 권한을 이 역할이 제공하는지 확인하세요.
  - 이 역할이 `secretsmanager:GetSecretValue`에 대해 암호에 액세스할 수 있는 권한을 제공하는지 확인하세요.

> **Note**
>
> 정책에서 비밀번호의 ARN을 지정해야 합니다. ARN에는 비밀 이름이 포함되지만 이름은 무작위 영숫자 6자로 추가됩니다. 시크릿에 대한 ARN의 형식은 다음과 같습니다.
>
> ```
> arn:aws:secretsmanager:region:account-id:secret:aws/
> transfer/SecretName-6RandomCharacters
> ```

- 사용자의 이전 요청을 처리할 때 커넥터가 리소스에 액세스할 수 있도록 허용하는 신뢰 관계가 이 역할에 포함되어 있는지 확인하세요. 신뢰 관계 설정에 대한 자세한 내용은 신뢰 관계를 구축하기 위해을 참조하세요.

다음 예제는 Amazon S3의 *DOC-EXAMPLE-BUCKET#* 액세스하는 데 필요한 권한을 부여하고 Secrets Manager에 저장된 지정된 시크릿을 부여합니다.

```
{
  "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**
>
> 액세스 역할의 경우, 예는 단일 암호에 대한 액세스 권한을 부여합니다. 하지만 와일드카드 문자를 사용하면 여러 사용자 및 암호에 대해 동일한 IAM 역할을 재사용하려는 경우 작업을 줄일 수 있습니다. 예컨대, 다음 리소스 명령문은 aws/transfer으로 시작하는 이름을 가진 모든 암호에 대한 권한을 부여합니다.
>
> ```
> "Resource": "arn:aws:secretsmanager:region:account-id:secret:aws/transfer/
> *"
> ```
>
> SFTP 자격 증명이 포함된 암호를 다른 AWS 계정에 저장할 수도 있습니다. 계정 간 보안 액세스를 활성화하는 방법에 대한 자세한 내용은 다른 계정의 사용자에 대한 보안 권한 AWS Secrets Manager 섹션을 참조하십시오.

- (선택 사항) 이벤트를 로그로 푸시하는 데 사용할 커넥터의 IAM 역할을 선택합니다. CloudWatch 다음 예제 정책은 SFTP 커넥터의 이벤트를 기록하는 데 필요한 권한을 나열합니다.

```
{
    "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/*"
        ]
    }]
}
```

- 다음 SFTP 구성 정보를 제공하세요.
  - SFTP 사용자의 개인 키 또는 AWS Secrets Manager 비밀번호가 포함된 비밀번호의 ARN입니다.
  - 외부 서버를 식별하는 데 사용되는 호스트 키의 공개 부분입니다. 원하는 경우 신뢰 호스트 키를 여러 개 제공할 수 있습니다.

  SFTP 정보를 제공하는 가장 쉬운 방법은 SFTP 정보를 파일에 저장하는 것입니다. 예컨대, 다음 예 텍스트를 `testSFTPConfig.json`이라는 파일에 복사합니다.

```
// 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="
   ]
}
```

- 보안 정책 이름을 입력하여 커넥터의 보안 정책을 지정합니다.

> **Note**
>
> `SecretId`는 전체 ARN이거나 암호 이름(이전 목록의 *example-username-key*)일 수 있습니다.

이제 다음 명령을 실행하여 커넥터를 생성합니다.

```
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
```

SFTP 커넥터와 함께 사용할 암호를 저장합니다.

Secrets Manager를 사용하여 SFTP 커넥터의 사용자 자격 증명을 저장할 수 있습니다. 암호를 생성할 때 사용자 이름을 입력해야 합니다. 또한 암호, 프라이빗 키 또는 둘 다 제공할 수 있습니다. 자세한 내용은 Transfer for SFTP 커넥터 할당량단원을 참조하세요.

> **Note**
>
> Secrets Manager에 비밀을 저장하면 AWS 계정 요금이 발생합니다. 요금에 대한 자세한 내용은 AWS Secrets Manager 요금을 참조하세요.

SFTP 커넥터의 사용자 자격 증명을 Secrets Manager에 저장하려면

1. https://console.aws.amazon.com/secretsmanager/ 에서 AWS Management Console 로그인하고 AWS Secrets Manager 콘솔을 여십시오.
2. 왼쪽 탐색 창에서 암호를 선택합니다.
3. 암호 페이지에서 새 암호 저장을 선택합니다.
4. 암호 선택 페이지의 암호 타입에서 다른 타입의 암호를 선택합니다.
5. 키/값 쌍 섹션에서 키/값 탭을 선택합니다.
   - 키 - **Username**를 입력합니다.
   - 값 - 파트너 서버에 연결할 권한이 있는 사용자의 이름을 입력합니다.
6. 암호를 제공하려면 행 추가를 선택하고 키/값 쌍 섹션에서 키/값 탭을 선택합니다.

   행 추가를 선택하고 키/값 쌍 섹션에서 키/값 탭을 선택합니다.
   - 키 - **Password**를 입력합니다.
   - 값 – 사용자의 암호를 입력합니다.
7. 프라이빗 키를 제공하려면 프라이빗 키 데이터를 입력하는 방법에 대해 설명하는 SFTP 커넥터 개인 키 생성 및 형식 지정 섹션을 참조하세요.

> **Note**
>
> 입력하는 프라이빗 키 데이터는 원격 SFTP 서버에 이 사용자에 대해 저장된 퍼블릭 키와 일치해야 합니다.

8. 다음을 선택합니다.
9. 암호 구성 페이지에 암호를 위한 명칭과 설명을 입력합니다. 이름에 **aws/transfer/**이라는 접두사를 사용하는 것이 좋습니다. 예컨대, 암호 **aws/transfer/connector-1**을 지정할 수 있습니다.
10. 다음을 선택한 후 교체 구성 페이지의 기본값을 그대로 사용합니다. 그 다음 다음을 선택합니다.
11. 검토 페이지에서 저장을 선택하여 암호를 만들고 저장합니다.

SFTP 커넥터 개인 키 생성 및 형식 지정

공개/개인 키 쌍 생성에 대한 자세한 내용은 에 설명되어 있습니다. macOS, Linux 또는 Unix에서 SSH 키 생성

예를 들어, SFTP 커넥터에 사용할 프라이빗 키를 생성하려면 다음 샘플 명령을 사용하여 올바른 유형의 키를 생성합니다 (*key_name#* 키 쌍의 실제 파일 이름으로 대체).

```
ssh-keygen -t rsa -b 4096 -m PEM -f key_name -N ""
```

> **Note**
>
> SFTP 커넥터와 함께 사용할 키 페어를 생성할 때는 패스프레이즈를 사용하지 마십시오. SFTP 구성이 제대로 작동하려면 빈 암호가 필요합니다.

이 명령은 키 크기가 4096비트인 RSA 키 쌍을 만듭니다. 키는 레거시 PEM 형식으로 생성되며, 이 형식은 Transfer Family에서 SFTP 커넥터 암호와 함께 사용하기 위해 필요합니다. 키는 현재 디렉터리, 즉 명령을 실행하는 디렉터리의 *key_name key_name*.pub (개인 키) 및 (공개 키) 에 저장됩니다. ssh-keygen

> **Note**
>
> Transfer Family는 SFTP 커넥터에 사용되는 키에 대해 OpenSSH 형식 `-----BEGIN OPENSSH PRIVATE KEY-----` () 을 지원하지 않습니다. 키는 레거시 PEM 형식(`-----BEGIN RSA PRIVATE KEY-----` 또는 `-----BEGIN EC PRIVATE KEY-----`)이어야 합니다. ssh-keygen 도구를 사용하여 명령을 실행할 때 `-m PEM` 옵션을 제공하여 키를 변환할 수 있습니다.

키를 생성한 후에는 개인 키의 형식이 JSON 형식의 개행 문자 (”\ n “) 로 지정되었는지 확인해야 합니다.

명령을 사용하여 기존 개인 키를 올바른 형식 (개행 문자가 포함된 JSON 형식) 으로 변환하십시오. 여기서는 Powershell의 예를 제공합니다. `jq` 개인 키를 줄 바꿈 문자가 포함된 JSON 형식으로 변환하려는 도구 또는 명령을 사용할 수 있습니다.

jq command

이 예제에서는 Download jq에서 다운로드할 수 있는 `jq` 명령을 사용합니다.

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

예를 들어 프라이빗 키 파일이 에 `~/.ssh/my_private_key` 있는 경우 명령은 다음과 같습니다.

```
jq -sR . ~/.ssh/my_private_key
```

그러면 키가 올바른 형식 (포함된 줄 바꿈 문자 포함) 으로 표준 출력에 출력됩니다.

PowerShell

Windows를 사용하는 경우 키를 올바른 형식으로 PowerShell 변환하는 데 사용할 수 있습니다. 다음 Powershell 명령은 개인 키를 올바른 형식으로 변환합니다.

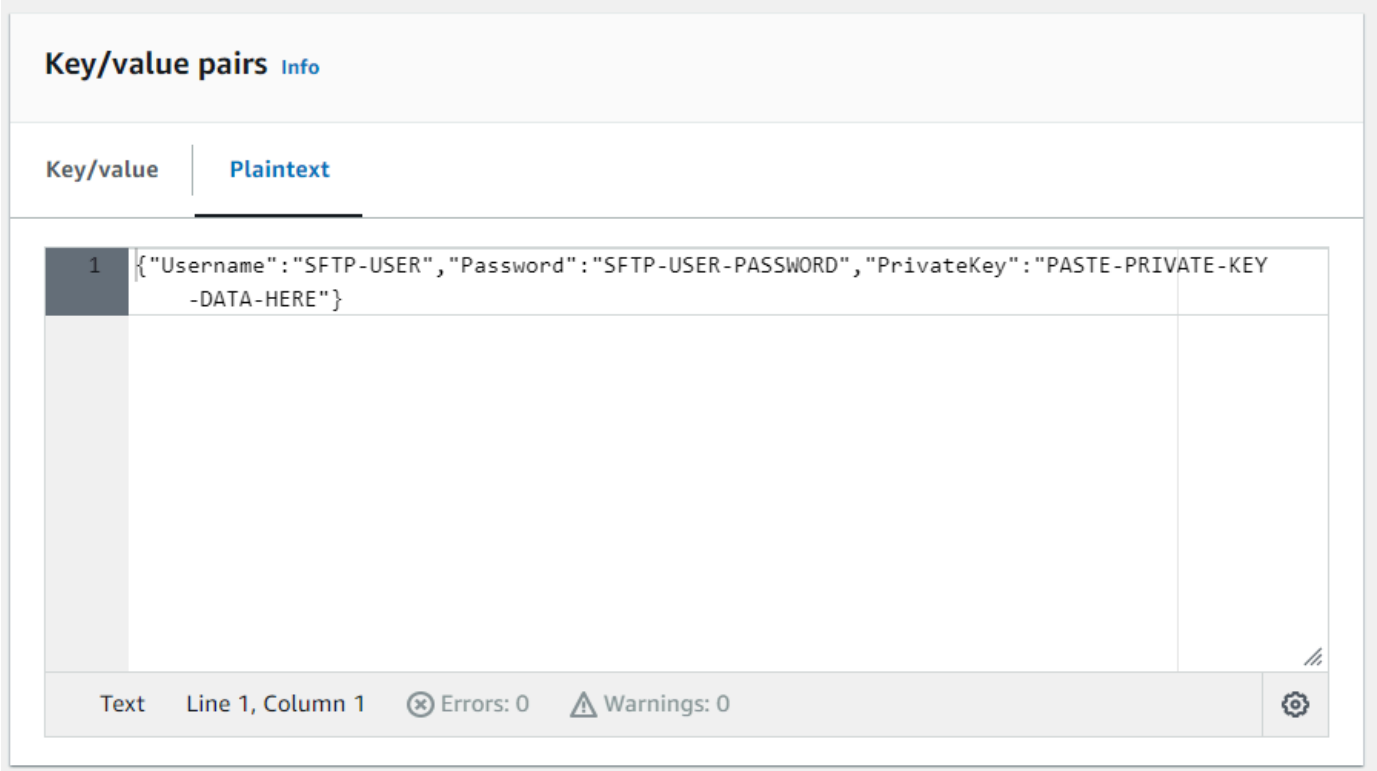
```
Get-Content -Raw path-to-private-key-file | ConvertTo-Json
```

SFTP 커넥터와 함께 사용할 프라이빗 키 데이터를 암호에 추가하려면

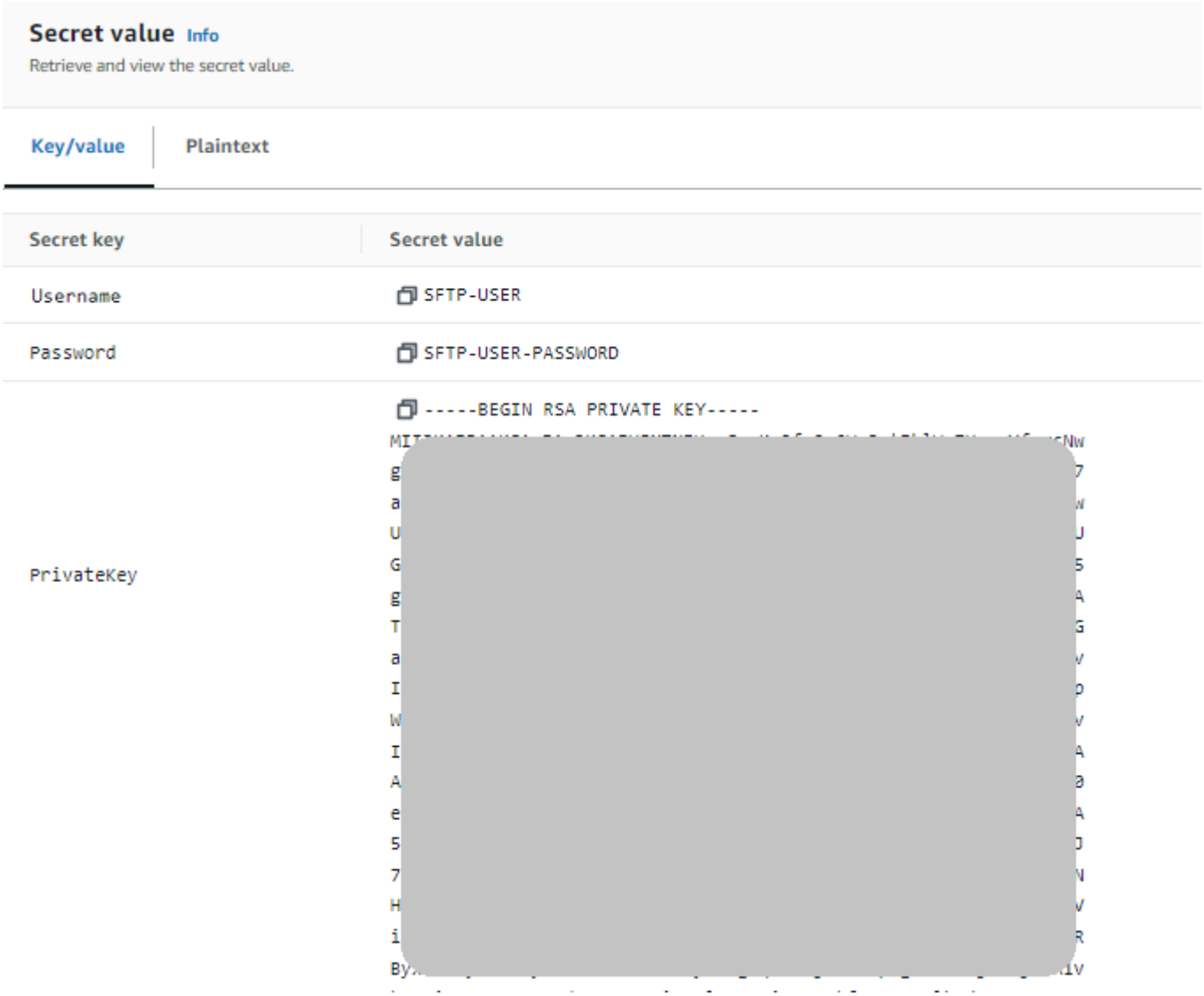
1. Secrets Manager 콘솔에서 기타 타입의 암호를 저장할 때 일반 텍스트 탭을 선택합니다. 텍스트는 비어 있어야 하며 여는 중괄호와 닫는 중괄호({})만 있어야 합니다.

2. 다음 형식을 사용하여 사용자 이름, 프라이빗 키 데이터 및/또는 암호를 붙여 넣습니다. 프라이빗 키 데이터의 경우 1단계에서 실행한 명령의 출력을 붙여 넣습니다.

```
{"Username":"SFTP-USER","Password":"SFTP-USER-PASSWORD","PrivateKey":"PASTE-PRIVATE-KEY-DATA-HERE"}
```



프라이빗 키 데이터를 올바르게 붙여 넣으면 키/값 탭을 선택할 때 다음과 같은 내용이 표시됩니다. 개인 키 데이터는 연속적인 텍스트 문자열이 아니라 표시되는 line-by-line 것을 알 수 있습니다.



3. 8단계의 SFTP 커넥터와 함께 사용할 암호를 저장합니다.의 절차를 계속하고 끝까지 해당 절차를 따르세요.

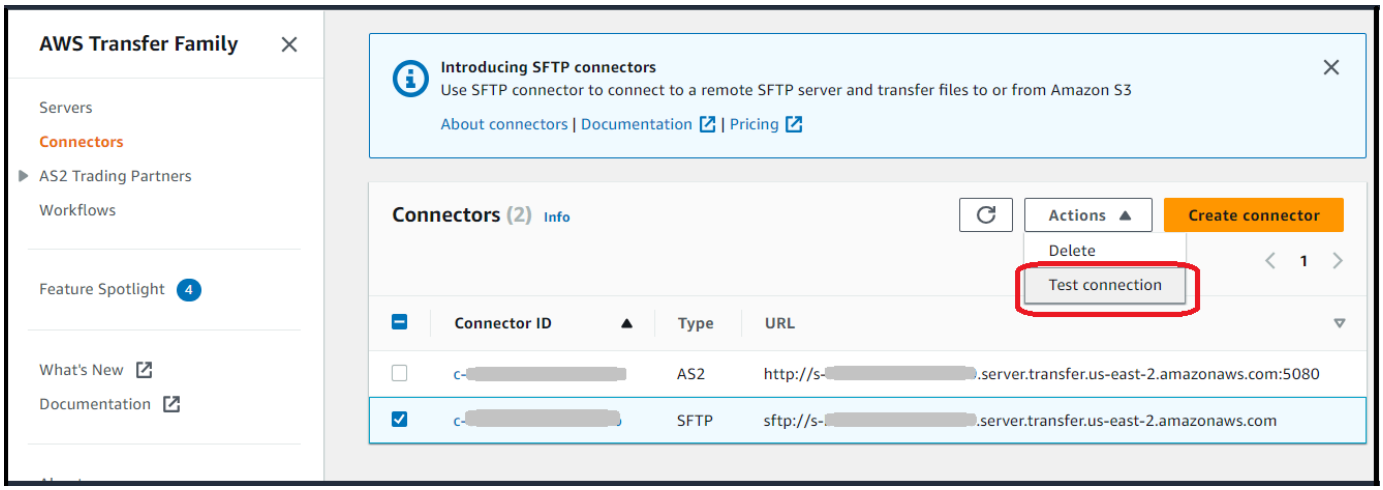
SFTP 커넥터를 테스트합니다.

SFTP 커넥터를 만든 후에는 새 커넥터를 사용하여 파일을 전송하기 전에 테스트해 보는 것이 좋습니다.

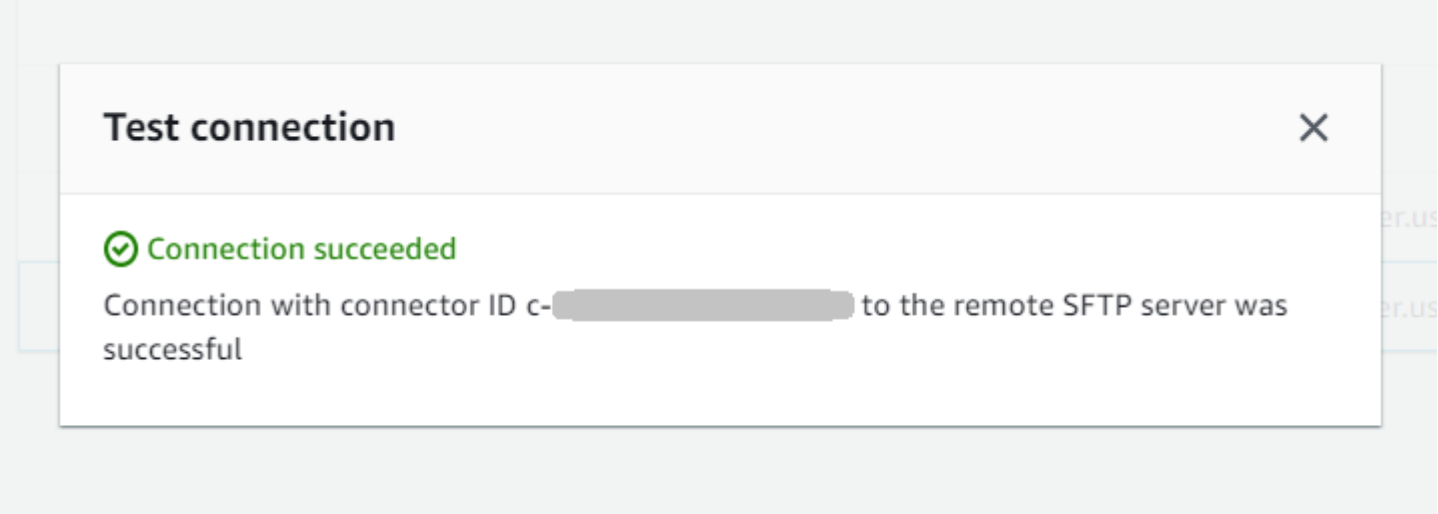
SFTP 커넥터를 테스트하려면

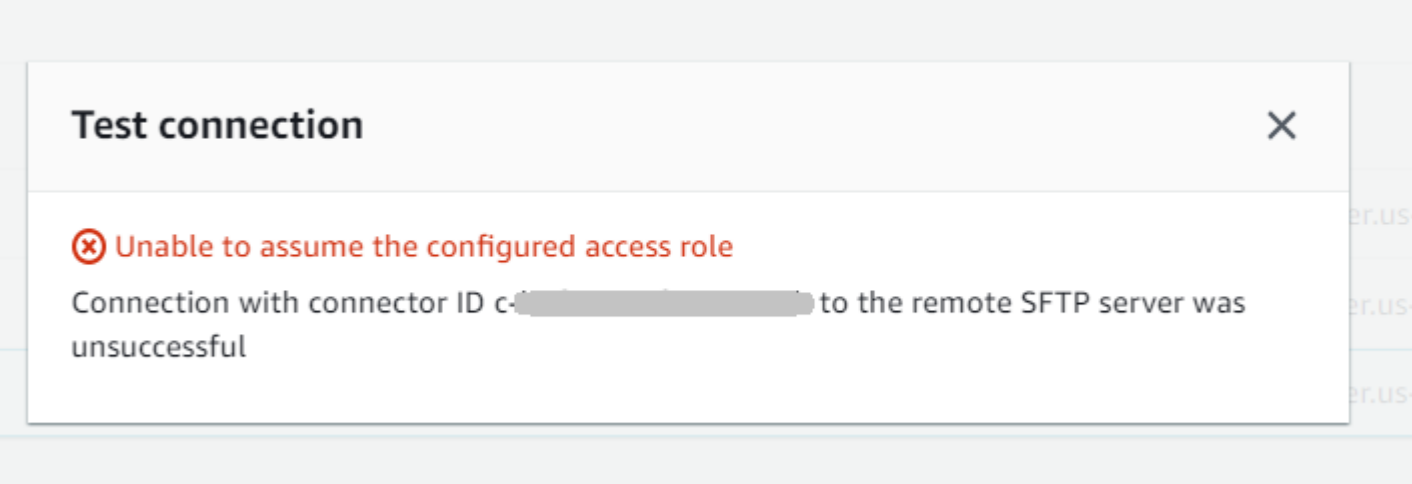
1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 여십시오.
2. 왼쪽 탐색 창에서 커넥터를 선택하고 커넥터를 선택합니다.

3. 작업 메뉴에서 테스트 연결을 선택합니다.



시스템은 테스트의 통과 또는 실패 여부를 나타내는 메시지를 반환합니다. 테스트가 실패하면 시스템은 테스트 실패 이유를 기반으로 오류 메시지를 제공합니다.





> **Note**
>
> API를 사용하여 커넥터를 테스트하려면 TestConnection API 설명서를 참조하세요.

SFTP 커넥터를 사용하여 파일을 보내고 검색합니다.

SFTP 커넥터는 클라우드와 온프레미스 모두에서 원격 서버와 AWS Transfer Family 통신할 수 있도록 기능을 확장합니다. 원격 소스에서 생성 및 저장된 데이터를 분석, 비즈니스 애플리케이션, 보고 및 감사를 위해 AWS 호스팅된 데이터 웨어하우스와 통합할 수 있습니다.

원격 SFTP 서버로 파일 전송을 시작하려면 SFTP 커넥터를 사용하여 전송을 수행하는 StartFileTransfer API 작업을 사용합니다. 각 `StartFileTransfer` 요청에는 10개의 고유한 경로가 포함될 수 있습니다.

서버 로그를 확인하여 파일 전송을 모니터링할 수 있습니다. 커넥터 활동은 `aws/transfer/`*`connector-id`* 형식(예: `aws/transfer/c-1234567890abcdef0`)의 로그 스트림에 기록됩니다. 커넥터에 대한 로그가 표시되지 않는 경우 커넥터에 대한 올바른 권한을 가진 로깅 역할을 지정했는지 확인하세요.

커넥터 생성에 대한 자세한 설명은 SFTP 커넥터 구성 섹션을 참조하세요.

SFTP 커넥터를 사용하여 파일을 보내고 검색하려면 () 명령을 사용합니다. `start-file-transfer` AWS Command Line Interface AWS CLI파일 전송(아웃바운드 전송) 또는 파일 수신(인바운드 전송) 여부에 따라 다음 매개 변수를 지정합니다.

- 아웃바운드 전송
  - `send-file-paths`에는 파트너의 SFTP 서버로 전송할 파일을 위한 1~10개의 소스 파일 경로가 포함됩니다.
  - `remote-directory-path`는 고객의 SFTP 서버에서 파일을 전송할 원격 경로입니다.
- 인바운드 전송
  - `retrieve-file-paths`에는 1~10개의 원격 경로가 포함됩니다. 각 경로는 파트너의 SFTP 서버에서 Transfer Family 서버로 파일을 전송할 위치를 지정합니다.
  - `local-directory-path`는 파일이 저장되는 Amazon S3 위치 (버킷 및 선택적 접두사) 입니다.

파일을 보내려면 `send-file-paths` 및 `remote-directory-path` 파라미터를 지정합니다. 최대 10개의 파일을 `send-file-paths` 파라미터에 지정할 수 있습니다. 다음 예 명령은 Amazon S3 스토

리지에 있는 /DOC-EXAMPLE-SOURCE-BUCKET/file1.txt 및 /DOC-EXAMPLE-SOURCE-BUCKET/file2.txt라는 파일을 파트너의 SFTP 서버의 /tmp 디렉터리로 보냅니다. 이 예 명령을 사용하려면 *DOC-EXAMPLE-SOURCE-BUCKET*를 실제 버킷으로 대체하세요.

```
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
```

파일을 수신하려면 retrieve-file-paths 및 local-directory-path 파라미터를 지정합니다. *## ### #### SFTP #### ### /my/remote/file1.txt #### Amazon S3 ## /DOC-EXAMPLE-BUCKET/ #### #####. /my/remote/file2.txt* 이 예 명령을 사용하려면 *user input placeholders*를 실제 정보로 대체하세요.

```
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
```

이전 예는 SFTP 서버의 절대 경로를 지정합니다. 상대 경로, 즉 SFTP 사용자의 홈 디렉터리를 기준으로 하는 경로를 사용할 수도 있습니다. 예컨대, SFTP 사용자가 marymajor이고 SFTP 서버의 홈 디렉터리가 /users/marymajor/인 경우 다음 명령은 /DOC-EXAMPLE-SOURCE-BUCKET/file1.txt을 /users/marymajor/test-connectors/file1.txt으로 전송합니다.

```
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
```

원격 디렉터리 콘텐츠 목록

원격 SFTP 서버에서 파일을 검색하기 전에 원격 SFTP 서버에 있는 디렉토리의 내용을 검색할 수 있습니다. 이 작업을 수행하려면 StartDirectoryListingAPI 호출을 사용합니다.

다음 예제는 커넥터의 컨피그레이션에 지정된 원격 SFTP 서버의 home 폴더 내용을 나열합니다. 결과는 Amazon S3 위치와 /DOC-EXAMPLE-BUCKET/connector-files 이름이 지정된 파일에 저장됩니다c-AAAA1111BBBB2222C-6666abcd-11aa-22bb-cc33-0000aaaa3333.json.

```
aws transfer start-directory-listing  \
```

```
    --connector-id c-AAAA1111BBBB2222C  \
    --output-directory-path /DOC-EXAMPLE-BUCKET/example/connector-files  \
    --remote-directory-path /home
```

이 AWS CLI 명령은 목록 ID와 결과를 포함하는 파일 이름을 반환합니다.

```
{
    "ListingId": "6666abcd-11aa-22bb-cc33-0000aaaa3333",
    "OutputFileName": "c-AAAA1111BBBB2222C-6666abcd-11aa-22bb-cc33-0000aaaa3333.json"
}
```

> **Note**
>
> 출력 파일의 명명 규칙은 다음과 *connector-ID*-*listing-ID*.json 같습니다.

JSON 파일에는 다음 정보가 포함됩니다.

- `filePath`: 원격 서버의 SFTP 커넥터에 대한 목록 요청 디렉토리를 기준으로 한 원격 파일의 전체 경로입니다.
- `modifiedTimestamp`: 파일이 마지막으로 수정된 시간 (초), 협정 세계시 (UTC) 형식. 이 필드는 선택 사항입니다. 원격 파일 속성에 타임스탬프가 없는 경우 파일 목록에서 제외됩니다.
- `size`: 파일 크기 (바이트). 이 필드는 선택 사항입니다. 원격 파일 속성에 파일 크기가 포함되지 않은 경우 파일 목록에서 제외됩니다.
- `path`: 원격 서버의 SFTP 커넥터에 대한 목록 작성 요청 디렉터리를 기준으로 한 원격 디렉터리의 전체 경로입니다.
- `truncated`: 목록 출력에 원격 디렉터리에 포함된 모든 항목이 포함되는지 여부를 나타내는 플래그입니다. `truncated`출력 값이 true인 경우 선택적 `max-items` 입력 속성에 제공된 값을 늘려 더 많은 항목을 나열할 수 있습니다 (최대 허용 목록 크기인 10,000개 항목).

다음은 출력 파일 (`c-AAAA1111BBBB2222C-6666abcd-11aa-22bb-cc33-0000aaaa3333.json`) 의 내용 예제입니다. 여기서 원격 디렉터리에는 두 개의 파일과 두 개의 하위 디렉터리 (경로) 가 있습니다.

```
{
    "files": [
        {
```

```
            "filePath": "/home/what.txt",
            "modifiedTimestamp": "2024-01-30T20:34:54Z",
            "size" : 2323
        },
        {
            "filePath": "/home/how.pgp",
            "modifiedTimestamp": "2024-01-30T20:34:54Z",
            "size" : 4691
        }
    ],
    "paths": [
        {
            "path": "/home/magic"
        },
        {
            "path": "/home/aws"
        },
    ],
    "truncated": "false"
}
```

SFTP 커넥터 관리

이 항목에서는 SFTP 커넥터를 보고 업데이트하는 방법을 설명하고 SFTP 커넥터와 관련된 할당량을 나열합니다.

> **Note**
>
> 각 커넥터에는 커넥터의 수명 기간 동안 변경되지 않는 고정 IP 주소가 자동으로 할당됩니다. 이렇게 하면 알려진 IP 주소의 인바운드 연결만 허용하는 원격 SFTP 서버에 연결할 수 있습니다. 커넥터에는 동일한 프로토콜 (SFTP 또는 AS2) 을 사용하는 모든 커넥터가 공유하는 고정 IP 주소 세트가 할당됩니다. AWS 계정

주제

- SFTP 커넥터 업데이트
- SFTP 커넥터 세부 정보 보기
- Transfer for SFTP 커넥터 할당량

SFTP 커넥터 업데이트

커넥터의 기존 파라미터 값을 변경하려면 `update-connector` 명령을 실행하면 됩니다. 다음 명령은 지역의 커넥터 *`connector-id`*의 암호를 *`region-id`*에서 *`secret-ARN`*로 업데이트합니다. 이 예 명령을 사용하려면 *`user input placeholders`*를 실제 정보로 대체하세요.

```
aws transfer update-connector --sftp-config '{"UserSecretId":"secret-ARN"}' \
   --connector-id connector-id --region region-id
```

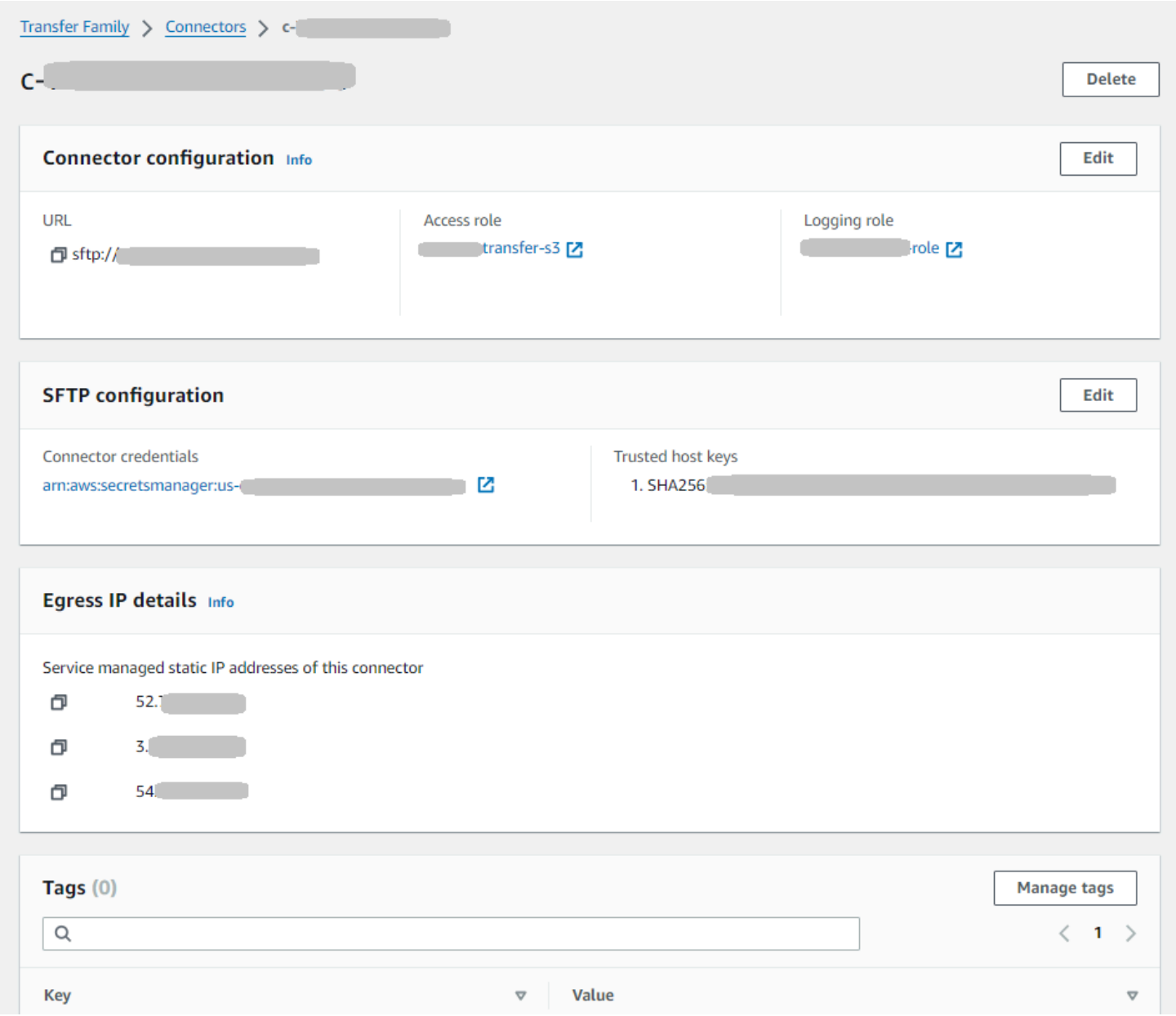
SFTP 커넥터 세부 정보 보기

AWS Transfer Family 콘솔에서 SFTP 커넥터의 세부 정보 및 속성 목록을 찾을 수 있습니다.

커넥터 세부 정보를 보려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 커넥터를 선택합니다.
3. 커넥터 ID 열에서 식별자를 선택하면 선택한 커넥터의 세부 정보 페이지가 표시됩니다.

커넥터 세부 정보 페이지에서 편집을 선택하여 SFTP 커넥터의 속성을 변경할 수 있습니다.



> **Note**
>
> 형식은 다르지만 다음 AWS Command Line Interface (AWS CLI) 명령을 실행하면 이 정보의 대부분을 얻을 수 있습니다. 이 예 명령을 사용하려면 *user input placeholders*를 실제 정보로 대체하세요.
>
> ```
> aws transfer describe-connector --connector-id your-connector-id
> ```
>
> 자세한 내용은 API 참조의 DescribeConnector를 참조하세요.

Transfer for SFTP 커넥터 할당량

SFTP 커넥터에는 다음과 같은 할당량이 적용됩니다.

> **Note**
>
> SFTP 커넥터에 대한 추가 서비스 할당량은 의 AWS Transfer Family 엔드포인트와 할당량에 나열되어 있습니다. Amazon Web Services 일반 참조

SFTP 커넥터 할당량

명칭	기본값	조정 가능
초당 최대 테스트 연결 트랜잭션(TPS)	계정 한 개에 대해 초당 1개의 요청	아니요
보류 중인 파일 전송의 최대 대기열 크기	1000	아니요
최대 파일 크기	50기가바이트 (GiB)	아니요
파일당 최대 전송 시간	6시간	아니요
파일당 최대 요청 대기 시간	6시간	아니요
계정당 커넥터의 최대 대역폭(SFTP 및 AS2 커넥터 모두 이 값에 영향을 미침)	50MBps	아니요

SFTP 커넥터의 자격 증명을 저장하기 위해 각 Secrets Manager 암호와 관련된 할당량이 있습니다. 여러 목적으로 동일한 암호를 사용하여 여러 유형의 키를 저장하는 경우 이러한 할당량이 발생할 수 있습니다.

- 단일 비밀의 총 길이: 12,000자
- **`Password`**문자열의 최대 길이: 1024자
- **`PrivateKey`**문자열의 최대 길이: 8192자
- **`Username`**문자열의 최대 길이: 100자

AWS Transfer Family AS2의 경우

적용성 보고서 2(AS2)는 강력한 메시지 보호 및 검증 메커니즘을 포함하는 RFC 정의 파일 전송 사양입니다. AS2 프로토콜은 프로토콜에 내장된 데이터 보호 및 보안 기능을 필요로 하는 규정 준수 요구 사항이 있는 워크플로에 매우 중요합니다.

> **Note**
>
> Transfer Family용 AS2는 Drummond 인증을 받았습니다.

공급망, 물류 및 결제 워크플로에 AS2를 사용하는 소매, 생명 과학, 제조, 금융 서비스, 유틸리티 등의 산업 분야의 고객은 AWS Transfer Family AS2 엔드포인트를 사용하여 비즈니스 파트너와 안전하게 거래할 수 있습니다. 트랜잭션 데이터는 기본적으로 처리, 분석 및 기계 AWS 학습을 위해 액세스할 수 있습니다. 이 데이터는 AWS에서 실행되는 ERP(전사적 자원 관리) 및 CRM(고객 관계 관리) 시스템과의 통합에도 사용할 수 있습니다. AS2를 통해 고객은 기존 비즈니스 파트너 통합 및 규정 준수를 AWS 유지하면서 규모에 맞게 business-to-business (B2B) 거래를 실행할 수 있습니다.

구성된 AS2 지원 서버를 보유한 파트너와 파일을 교환하려는 Transfer Family 고객의 경우 설정에는 암호화를 위한 퍼블릭-프라이빗 키 쌍과 파트너와의 서명 및 퍼블릭 키 교환을 위한 다른 한 쌍을 생성하는 작업이 포함됩니다.

Transfer Family는 AS2가 활성화된 Transfer Family 엔드포인트와 Transfer Family AS2 커넥터를 구성할 수 있는 워크숍을 제공합니다. 이 워크숍에 대한 세부 정보는 여기에서 확인할 수 있습니다.

전송 중인 AS2 페이로드를 보호하려면 일반적으로 CMS(Cryptographic Message Syntax)를 사용하며, 일반적으로 암호화와 디지털 서명을 사용하여 데이터 보호 및 피어 인증을 제공합니다. 서명된 메시지 처리 알림(MDN) 응답 페이로드는 메시지가 수신되고 성공적으로 복호화되었다는 확인(부인 방지)을 제공합니다.

이러한 CMS 페이로드 및 MDN 응답의 전송은 HTTP를 통해 이루어집니다.

> **Note**
>
> HTTPS AS2 서버 엔드포인트는 현재 지원되지 않습니다. TLS 종료는 현재 고객의 책임입니다.

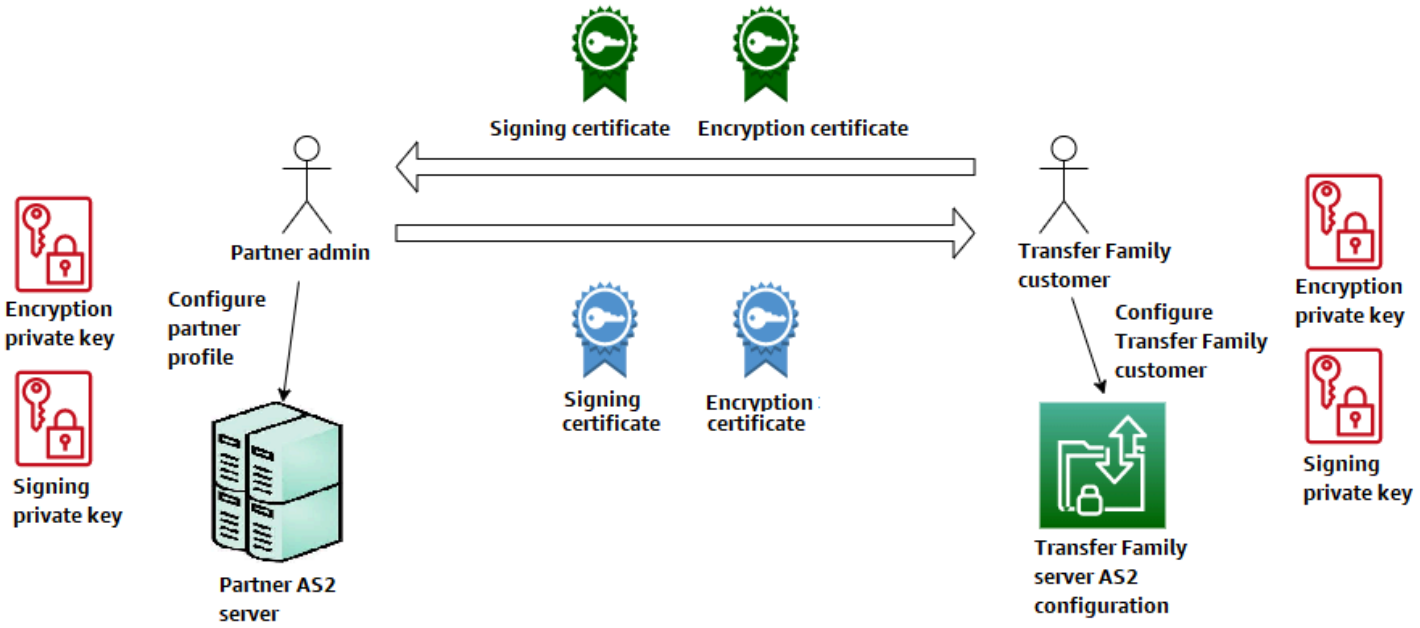
AS2 (Applibility step-by-step Statement 2) 구성 설정에 대한 자세한 내용은 자습서를 참조하십시오. AS2 구성 설정

주제

- AS2 사용 사례
- AS2 구성
- AS2 커넥터 구성
- AS2 파트너 관리
- AS2 메시지 전송 및 수신
- AS2 사용량 모니터링

AS2 사용 사례

AS2 서버를 구성한 파트너와 파일을 교환하려는 AWS Transfer Family 고객의 경우 설정의 가장 복잡한 부분은 암호화를 위한 공개-개인 키 쌍과 파트너와의 서명 및 공개 키 교환을 위한 다른 한 쌍을 생성하는 것입니다.



AS2와 함께 사용할 때는 다음과 같은 변형을 고려해 보십시오. AWS Transfer Family

> **Note**
>
> 거래 파트너는 해당 파트너 프로필과 관련된 파트너입니다.

다음 표의 MDN에 대한 모든 언급은 서명된 MDN을 가정합니다.

AS2 사용 사례

인바운드 전용 사용 사례

- 거래 파트너의 암호화된 AS2 메시지를 Transfer Family 서버로 전송합니다.

  이 경우 다음과 같이 합니다:

  1. 거래 파트너와 자신을 위한 프로필을 만드세요.
  2. AS2 프로토콜을 사용하는 Transfer Family 서버를 생성합니다.
  3. 계약서를 작성하여 서버에 추가합니다.
  4. 개인 키가 포함된 인증서를 가져와 프로필에 추가한 다음 공개 키를 파트너 프로필로 가져와 암호화하십시오.
  5. 이러한 항목을 확보한 후에는 인증서용 퍼블릭 키를 거래 파트너에게 보내세요.

  이제 파트너가 암호화된 메시지를 보내고 사용자는 이를 복호화하여 Amazon S3 버킷에 저장할 수 있습니다.
- 거래 파트너의 암호화된 AS2 메시지를 Transfer Family 서버로 전송하고 서명을 추가합니다.

  이 시나리오에서는 여전히 인바운드 전송만 하고 있지만, 이제는 파트너가 보내는 메시지에 서명하도록 하는 것이 좋습니다. 이 경우 거래 파트너의 서명 공개 키 (파트너 프로필에 추가된 서명 인증서) 를 가져오십시오.
- 거래 파트너의 암호화된 AS2 메시지를 Transfer Family 서버로 전송하고 서명을 추가하고 MDN 응답을 보냅니다.

  이 시나리오에서는 여전히 인바운드 전송만 하고 있지만, 이제 거래 파트너는 서명된 페이로드를 받는 것 외에도 서명된 MDN 응답을 받기를 원합니다.

  1. 공개 및 비공개 서명 키(프로필의 서명 인증서로 사용)를 가져오세요.
  2. 공개 서명 키를 거래 파트너에게 보내십시오.

아웃바운드 전용 사용 사례

- 암호화된 AS2 메시지를 Transfer Family 서버에서 거래 파트너로 전송합니다.

이 경우는 AS2 서버에 계약을 추가하는 대신 커넥터를 만든다는 점을 제외하면 인바운드 전용 전송 사용 사례와 비슷합니다. 이 경우 거래 파트너의 공개 키를 거래 파트너의 프로필로 가져옵니다.

- Transfer Family 서버에서 거래 파트너로 암호화된 AS2 메시지를 전송하고 서명을 추가합니다.

  여전히 아웃바운드 전송만 하고 있지만, 이제 거래 파트너는 전송한 메시지에 서명하기를 원합니다.

  1. 서명 프라이빗 키(프로필에 추가된 서명 인증서)를 가져오세요.
  2. 거래 파트너에게 공개 키를 보내세요.
- Transfer Family 서버에서 암호화된 AS2 메시지를 거래 파트너로 전송하고 서명을 추가하고 MDN 응답을 보냅니다.

  여전히 아웃바운드 전송만 하고 있지만, 이제는 서명된 페이로드를 보내는 것 외에도 거래 파트너로부터 서명된 MDN 응답을 받고 싶을 것입니다.

  1. 거래 파트너가 공개 서명 키를 보내드립니다.
  2. 거래 파트너의 공개 키 (파트너 프로필에 추가된 서명 인증서) 를 가져오세요.

인바운드 및 아웃바운드 사용 사례

- Transfer Family 서버와 거래 파트너 간에 암호화된 AS2 메시지를 양방향으로 전송합니다.

  이 경우 다음과 같이 합니다:

  1. 거래 파트너와 자신을 위한 프로필을 만드세요.
  2. AS2 프로토콜을 사용하는 Transfer Family 서버를 생성합니다.
  3. 계약서를 작성하여 서버에 추가합니다.
  4. 커넥터를 생성합니다.
  5. 개인 키가 포함된 인증서를 가져와 프로필에 추가한 다음 공개 키를 파트너 프로필로 가져와 암호화하십시오.
  6. 거래 파트너로부터 공개 키를 받고 이를 거래 파트너의 프로필에 추가하여 암호화하세요.
  7. 이러한 항목을 확보한 후에는 인증서용 퍼블릭 키를 거래 파트너에게 보내세요.

  이제 거래 파트너와 암호화된 메시지를 교환할 수 있으며 둘 다 암호를 복호화할 수 있습니다. 사용자는 Amazon S3 버킷에 수신한 메시지를 저장할 수 있으며, 파트너는 사용자가 전송한 메시지를 복호화하여 저장할 수 있습니다.
- Transfer Family 서버와 거래 파트너 간에 양방향으로 암호화된 AS2 메시지를 전송하고 서명을 추가합니다.

  이제 당신과 당신의 파트너는 서명된 메시지를 원합니다.

  1. 서명 프라이빗 키(프로필에 추가된 서명 인증서)를 가져오세요.
  2. 거래 파트너에게 공개 키를 보내십시오.
  3. 거래 파트너의 서명 공개 키를 가져와 프로필에 추가하세요.
- Transfer Family 서버와 거래 파트너 간에 암호화된 AS2 메시지를 양방향으로 전송하고 서명을 추가하고 MDN 응답을 보냅니다.

  이제 서명된 페이로드를 교환하고 싶은데, 거래 파트너와 거래 파트너 모두 MDN 응답을 원합니다.

  1. 거래 파트너가 공개 서명 키를 보내드립니다.
  2. 거래 파트너의 공개 키 (파트너 프로필의 서명 인증서) 를 가져오세요.
  3. 거래 파트너에게 공개 키를 보내십시오.

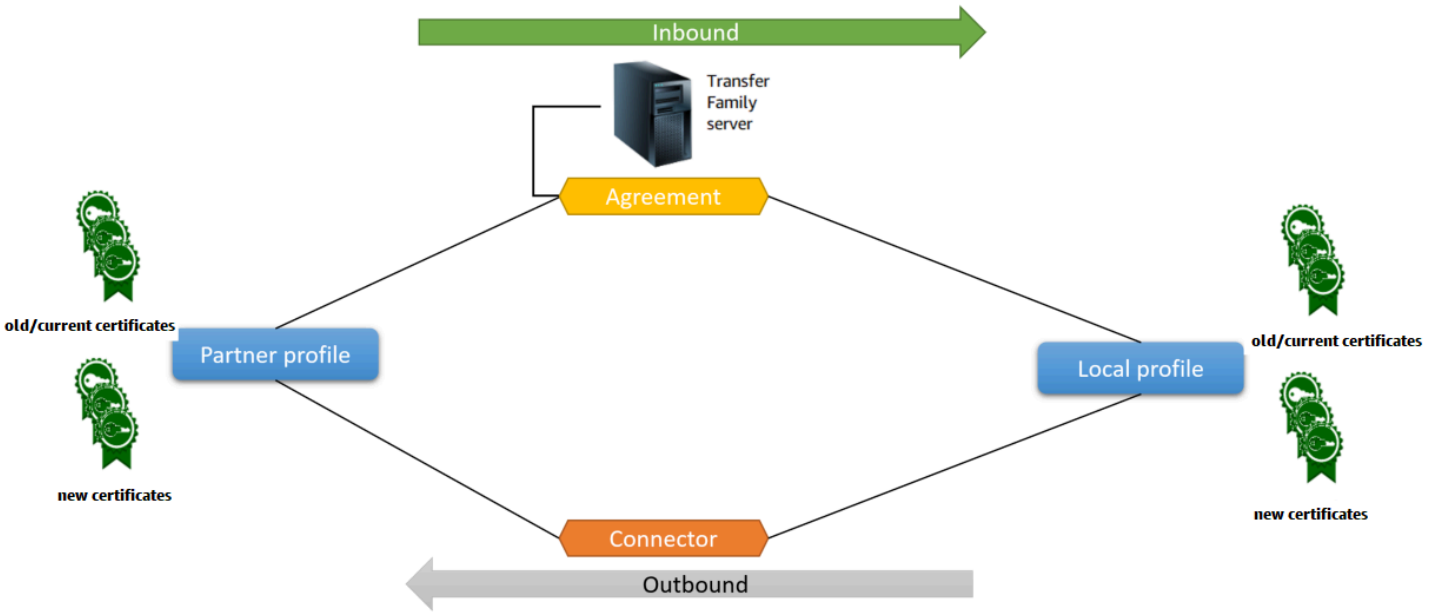
AS2 구성

AS2 지원 서버를 생성하려면 다음 구성 요소도 지정해야 합니다:

- 계약 - 양자간 거래 파트너 계약 또는 파트너십, 메시지(파일)를 교환하는 두 당사자들 사이의 관계를 정의함. 계약을 정의하기 위해 Transfer Family는 서버, 로컬 프로필, 파트너 프로필, 인증서 정보를 결합합니다. Transfer Family AS2-인바운드 프로세스는 계약을 사용합니다.
- 인증서 - 퍼블릭 키(X.509) 인증서는 AS2 통신에서 메시지 암호화 및 확인을 위해 사용됩니다. 인증서는 커넥터 엔드포인트에도 사용됩니다.
- 로컬 프로필 및 파트너 프로필 - 로컬 프로필은 로컬(AS2 지원 Transfer Family 서버) 조직 또는 "당사자"를 정의합니다. 마찬가지로 파트너 프로필은 Transfer Family 외부의 원격 파트너 조직을 정의합니다.

모든 AS2 지원 서버에 필수는 아니지만 아웃바운드 전송의 경우 커넥터가 필요합니다. 커넥터는 아웃바운드 연결에 대한 파라미터를 캡처합니다. 커넥터는 고객의 서버가 아닌 외부 서버로 파일을 전송하는 데 필요합니다. AWS

다음 다이어그램은 인바운드 프로세스와 아웃바운드 프로세스에 관련된 AS2 객체 간의 관계를 나타낸 것입니다.



AS2 구성의 end-to-end 예는 을 참조하십시오AS2 구성 설정.

주제

- Transfer Family 콘솔을 사용하여 AS2 서버 생성
- 템플릿을 사용하여 데모 Transfer Family AS2 스택을 생성하세요.
- AS2 구성 및 할당량
- AS2 특성 및 기능

Transfer Family 콘솔을 사용하여 AS2 서버 생성

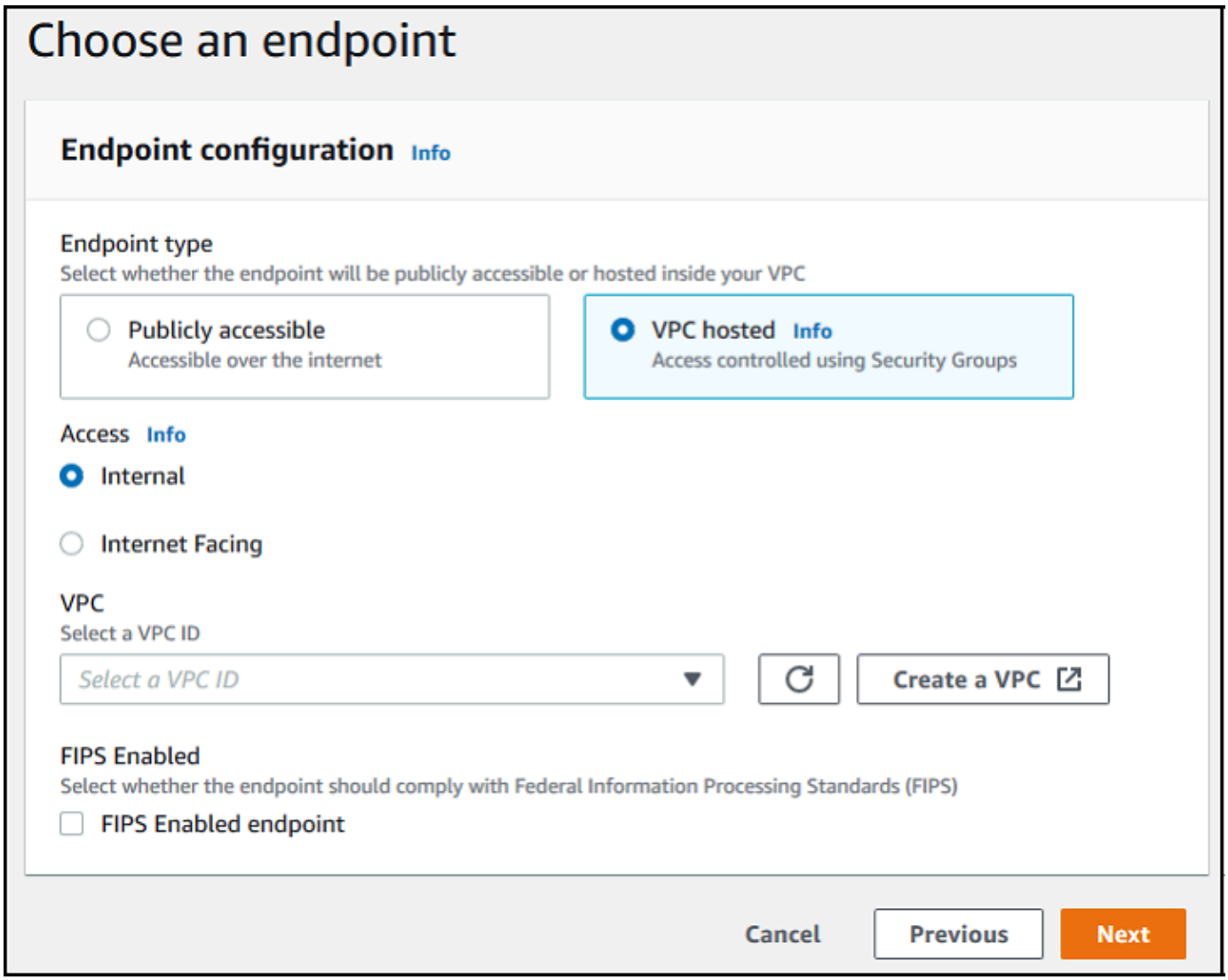
이 절차에서는 Transfer Family 콘솔을 사용하여 AS2 지원 서버를 생성하는 방법을 설명합니다. AWS CLI 대신 사용하려면 을 참조하십시오. the section called “2단계: AS2 프로토콜을 사용하는 Transfer Family 서버 생성”

AS2 지원 서버를 만들려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 탐색 창에서 서버를 선택한 다음 서버 생성을 선택합니다.
3. 프로토콜 선택 페이지에서 AS2(적용 가능성 선언문 2)를 선택하고 다음을 선택합니다.
4. 자격 증명 제공자 선택 페이지에서 다음을 선택합니다.

   > **Note**
   >
   > AS2의 경우 AS2 프로토콜에 대해 기본 인증이 지원되지 않기 때문에 ID 제공자를 선택할 수 없습니다. 대신 Virtual Private Cloud(VPC) 보안 그룹을 통해 액세스를 통제합니다.

5. 엔드포인트 선택 페이지에서 다음을 수행합니다.



a. 엔드포인트 타입에서 서버의 엔드포인트를 호스팅할 VPC 호스팅을 선택합니다. VPC 호스팅 엔드포인트를 설정하는 자세한 설명은 Virtual Private Cloud(VPC)에 서버 생성 섹션을 참조하세요.

> **Note**
>
> 공개적으로 액세스할 수 있는 엔드포인트는 AS2 프로토콜에서 지원되지 않습니다. 인터넷을 통해 VPC 엔드포인트에 액세스할 수 있게 하려면 액세스에서 인터넷 연결을 선택한 다음 탄력적 IP 주소를 입력합니다.

b. 액세스에서 다음 옵션 중 하나를 선택합니다.

- 내부 - 이 옵션을 선택하면 VPC 및 VPC로 연결된 환경(예: AWS Direct Connect 또는 VPN을 통한 온프레미스 데이터 센터 등) 내에서 액세스할 수 있습니다.
- 인터넷 연결 — 인터넷을 통해, 그리고 VPC 및 VPC로 연결된 환경 (예: 온프레미스 데이터 센터 또는 VPN) 내에서 액세스를 제공하려면 이 옵션을 선택합니다. AWS Direct Connect

  인터넷 연결을 선택하는 경우 메시지가 표시되면 탄력적 IP 주소를 제공하세요.

c. VPC의 경우 기존 VPC를 선택하거나 VPC 생성을 선택하여 새 VPC를 생성합니다.

d. FIPS 지원의 경우 FIPS 지원 엔드포인트 확인란의 선택을 취소한 상태로 유지합니다.

> **Note**
>
> FIPS 지원 엔드포인트는 AS2 프로토콜에서 지원되지 않습니다.

e. 다음을 선택합니다.

6. 도메인 선택 페이지에서 선택한 프로토콜을 사용하여 파일을 객체로 저장하고 액세스하려면 Amazon S3를 선택합니다.

   다음을 선택합니다.

7. 추가 세부 정보 구성 페이지에서 필요한 설정을 선택합니다.

> **Note**
>
> AS2와 함께 다른 프로토콜을 구성하는 경우 모든 추가 세부 설정이 적용됩니다. 하지만 AS2 프로토콜의 경우 로깅 및 태그 섹션의 설정만 적용됩니다. CloudWatch CloudWatch 로깅 역할 설정은 선택 사항이지만 메시지 상태를 확인하고 구성 문제를 해결할 수 있도록 로깅 역할을 설정하는 것이 좋습니다.

8. 검토 및 생성 페이지에서 옵션을 검토하여 올바른지 확인하세요.

- 설정을 편집하려면 변경하려는 단계 옆의 편집을 선택합니다.

> **Note**
>
> 단계를 편집하는 경우 편집하기로 선택한 단계 이후의 각 단계를 검토하는 것이 좋습니다.

- 변경 사항이 없는 경우 서버 생성을 선택하여 서버를 생성하세요. 서버 페이지로 이동하고, 새 서버가 나열되는 다음 화면이 표시됩니다.

  새 서버 상태가 온라인으로 변경되기까지 몇 분 정도 걸릴 수 있습니다. 이때부터 서버는 사용자의 파일 작업을 수행할 수 있습니다.

템플릿을 사용하여 데모 Transfer Family AS2 스택을 생성하세요.

AS2 지원 Transfer Family 서버를 신속하게 생성할 수 있는 독립형 AWS CloudFormation 템플릿을 제공합니다. 템플릿은 퍼블릭 Amazon VPC 엔드포인트, 인증서, 로컬 및 파트너 프로필, 계약, 커넥터로 서버를 구성합니다.

이 템플릿을 사용하기 전에 다음 사항에 유의하세요.

- 이 템플릿에서 스택을 생성할 경우, 사용한 AWS 리소스에 대한 요금이 청구됩니다.
- 템플릿은 여러 인증서를 생성하여 안전하게 보관할 수 AWS Secrets Manager 있도록 보관합니다. 이 서비스를 사용하면 요금이 부과되므로 원하는 경우 Secrets Manager에서 이러한 인증서를 삭제할 수 있습니다. Secrets Manager에서 이러한 인증서를 삭제해도 Transfer Family 서버에서는 삭제되지 않습니다. 따라서 데모 스택의 기능은 영향을 받지 않습니다. 하지만 프로덕션 AS2 서버에서 사용할 인증서의 경우 Secrets Manager를 사용하여 저장된 인증서를 관리하고 주기적으로 교체하는 것이 좋습니다.
- 템플릿은 기본 용도로만 사용하고 주로 데모 용도로만 사용하는 것이 좋습니다. 프로덕션 환경에서 이 데모 스택을 사용하려는 경우 템플릿의 YAML 코드를 수정하여 보다 강력한 스택을 만드는 것이 좋습니다. 예를 들어 프로덕션 수준의 인증서를 만들고 프로덕션에서 사용할 수 있는 AWS Lambda 함수를 만들어 보세요.

템플릿에서 AS2 지원 Transfer Family 서버를 만들려면 CloudFormation

1. https://console.aws.amazon.com/cloudformation 에서 AWS CloudFormation 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 스택을 선택합니다.
3. 스택 생성을 선택한 다음 새 리소스 사용(표준)을 선택합니다.
4. 사전 조건 - 템플릿 준비섹션에서 템플릿 준비 완료를 선택합니다.
5. AS2 데모 템플릿인 이 링크를 복사하여 Amazon S3 URL 필드에 붙여넣습니다.
6. 다음을 선택합니다.
7. 스택 세부 정보 지정 페이지에서 스택의 이름을 지정한 다음 파라미터를 지정합니다.

- AS2에서 로컬 AS2 ID 및 파트너 AS2 ID의 값을 입력하거나 기본값인 `local` 및 `partner`를 각각 적용합니다.
- 네트워크에서 보안 그룹 인그레스 CIDR IP에 값을 입력하거나 기본값 `0.0.0.0/0`을 그대로 사용합니다.

> **Note**
>
> 이 값은 CIDR 형식으로 AS2 서버로 들어오는 트래픽에 허용되는 IP 주소를 지정합니다. 기본값인 `0.0.0.0/0`은 모든 IP 주소를 허용합니다.

- 일반에서 접두사에 값을 입력하거나 기본값인 `transfer-as2`을 그대로 사용합니다. 이 접두사는 스택에서 생성되는 모든 리소스 이름 앞에 위치합니다. 예를 들어, 기본 접두사를 사용하는 경우 Amazon S3 버킷의 `transfer-as2-`*`TransferS3BucketName`*라는 이름이 지정됩니다.

8. 다음을 선택합니다. 스택 옵션 구성페이지에서 다음을 다시 선택합니다.
9. 생성 중인 스택의 세부 정보를 검토한 다음 스택 생성을 선택합니다.

> **Note**
>
> 페이지 하단의 기능에서 AWS Identity and Access Management (IAM) 리소스를 생성할 AWS CloudFormation 수 있음을 확인해야 합니다.

스택이 생성되면 AWS Command Line Interface (AWS CLI) 를 사용하여 파트너 서버에서 로컬 Transfer Family 서버로 테스트 AS2 메시지를 보낼 수 있습니다. 테스트 메시지를 보내기 위한 샘플 AWS CLI 명령이 스택의 다른 모든 리소스와 함께 생성됩니다.

이 샘플 명령어를 사용하려면 스택의 Outputs 탭으로 이동하여 TransferExampleAs2Command를 복사하십시오. 그런 다음 AWS CLI를 사용하여 명령을 실행할 수 있습니다. 를 아직 설치하지 않은 경우 AWS Command Line Interface 사용 설명서의 최신 버전 설치 또는 업데이트를 참조하십시오. AWS CLI AWS CLI

샘플 명령의 형식은 다음과 같습니다.

```
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**
>
> 이 명령의 버전에는 스택에 있는 *TransferS3BucketName* 및 *TransferConnectorId* 리소스의 실제 값이 포함되어 있습니다.

이 샘플 명령은 && 문자열을 사용하여 서로 연결된 두 개의 개별 명령으로 구성되어 있습니다.

첫 번째 명령은 버킷에 비어 있는 새 텍스트 파일을 만듭니다.

```
aws s3api put-object --bucket TransferS3BucketName --key test.txt
```

그런 다음 두 번째 명령은 커넥터를 사용하여 파트너 프로필에서 로컬 프로필로 파일을 보냅니다. Transfer Family 서버에는 로컬 프로필이 파트너 프로필의 메시지를 수락하도록 허용하는 계약이 설정되어 있습니다.

```
aws transfer start-file-transfer --region aws-region --connector-id TransferConnectorId
 --send-file-paths /TransferS3BucketName/test.txt
```

명령을 실행한 후 Amazon S3 버킷(*TransferS3BucketName*)으로 이동하여 콘텐츠를 볼 수 있습니다. 명령이 제대로 실행되면 버킷에 다음과 같은 객체가 표시되어야 합니다.

- `processed/`— 이 폴더에는 전송된 파일과 MDN 응답을 설명하는 JSON 파일이 들어 있습니다.
- `processing/`— 이 폴더에는 처리 중인 파일이 일시적으로 포함되지만 전송이 완료된 후에는 이 폴더가 비어 있어야 합니다.
- *server-id*/— 이 폴더의 이름은 Transfer Family 서버 ID에 따라 붙여집니다. 여기에는 `from-`*partner* (이 폴더의 이름은 파트너의 AS2 ID에 따라 동적으로 붙여짐)이고, 이것은 그 자체가 `failed/processed/`, 및 `processing/` 폴더가 들어 있습니다. /*server-id*/`from-`*partner*`/processed/` 폴더에는 전송된 텍스트 파일의 사본과 해당 JSON 및 MDN 파일이 들어 있습니다.
- `test.txt`— 이 객체는 전송된 (빈) 파일입니다.

AS2 구성 및 할당량

이 항목에서는 허용되는 암호 및 다이제스트를 포함하여 적용성 보고서 2(AS2) 프로토콜을 사용하는 전송에 대해 지원되는 구성, 기능 및 기능에 대해 설명합니다. 또한 이 섹션에서는 AS2 전송의 제한 및 알려진 문제에 대해서도 설명합니다.

주제

AS2 지원 구성

서명, 암호화, 압축, MDN

인바운드 전송과 아웃바운드 전송 모두에서 다음 항목은 필수 또는 옵션입니다.

- 암호화 - 필수 (현재 지원되는 유일한 전송 방법인 HTTP 전송의 경우). 암호화되지 않은 메시지는 ALB (Application Load Balancer) 와 같은 TLS 종료 프록시에 의해 전달되고 `X-Forwarded-Proto: https`헤더가 있는 경우에만 수락됩니다.
- 서명 — 옵션
- 압축 - 옵션(현재 지원되는 유일한 압축 알고리즘은 ZLIB)
- 메시지 처리 알림 (MDN) — 옵션

암호(Ciphers)

인바운드 전송과 아웃바운드 전송 모두에 지원되는 암호는 다음과 같습니다.

- AES128_CBC
- AES192_CBC
- AES256_CBC
- 3DES (이전 버전과의 호환성에만 해당)

Digests

다음 데이터 유형이 지원됩니다.

- 인바운드 서명 및 MDN — SHA1, SHA256, SHA384, SHA512
- 아웃바운드 서명 및 MDN – SHA1, SHA256, SHA384, SHA512

MDN

MDN 응답의 경우 다음과 같은 특정 타입이 지원됩니다.

- 인바운드 전송 — 동기 및 비동기
- 아웃바운드 전송— 동기식 전송 전용
- Simple Mail Transfer Protocol (SMTP) (이메일 MDN) — 지원되지 않음

Transports

- 인바운드 전송 - HTTP는 현재 지원되는 유일한 전송이므로 HTTP를 명시적으로 지정해야 합니다.

> **Note**
>
> 인바운드 전송에 HTTPS를 사용해야 하는 경우 Application Load Balancer 또는 Network Load Balancer에서 TLS를 종료할 수 있습니다. 이에 대한 설명은 HTTPS를 통해 AS2 메시지를 받습니다.에 나와 있습니다.

- 아웃바운드 전송 - HTTP URL을 제공하는 경우 암호화 알고리즘도 지정해야 합니다. HTTPS URL을 제공하는 경우 암호화 알고리즘에 NONE을 지정할 수 있습니다.

AS2 할당량 및 제한사항

이 섹션에서는 AS2의 할당량 및 제한 사항에 대해 설명합니다.

주제

- AS2 할당량
- 비밀 처리를 위한 할당량
- 알려진 제한 사항

AS2 할당량

AS2 파일 전송에는 다음과 같은 할당량이 적용됩니다. 조정 가능한 할당량 증가를 요청하려면 AWS 일반 참조의 AWS 서비스 할당량을 참조하세요.

AS2 할당량

명칭	기본값	조정 가능
초당 수신되는 최대 인바운드 파일 수	100	아니요

명칭	기본값	조정 가능
초당 전송되는 최대 아웃바운드 파일 수	100	아니요
최대 동시 인바운드 파일 수	400	아니요
최대 동시 아웃바운드 파일 수	400	아니요
인바운드 파일의 최대 크기 (비압축)	1GB	아니요
아웃바운드 파일의 최대 크기 (비압축)	1GB	아니요
아웃바운드 요청당 최대 파일 수	10	아니요
초당 최대 아웃바운드 요청 수	100	아니요
초당 최대 인바운드 요청 수	100	아니요
계정당 최대 아웃바운드 대역폭 (아웃바운드 SFTP 및 AS2 요청 모두 이 값에 영향을 미침)	초당 50MB	아니요
서버당 최대 계약 수	100	예
계정당 최대 커넥터 수 (SFTP 및 AS2 커넥터 모두 이 제한에 영향을 미침)	100	예
파트너 프로필당 최대 인증서 수	10	아니요
계정당 최대 인증서 수	1000	예
계정당 최대 파트너 프로필 개수	1000	예

비밀 처리를 위한 할당량

AWS Transfer Family 기본 인증을 사용하는 AS2 고객을 AWS Secrets Manager 대신하여 전화를 겁니다. 또한 Secrets Manager는 다음과 같은 전화를 걸 수 AWS KMS있습니다.

> **Note**
>
> 이러한 할당량은 Transfer Family의 비밀 사용에만 국한되는 것이 아니라 Transfer Family의 모든 서비스에서 공유됩니다. AWS 계정

Secrets `GetSecretValue`Manager의 경우 적용되는 할당량은 할당량에 설명된 대로 DescribeSecret 및 GetSecretValue API 요청을 합친 비율입니다.AWS Secrets Manager

Secrets Manager **`GetSecretValue`**

이름	값	설명
DescribeSecret 및 GetSecret Value API 요청의 합산 요금	지원되는 각 리전: 초당 10,000개	DescribeSecret 및 GetSecret Value API 요청을 합친 초당 최대 트랜잭션 수입니다.

에는 다음과 같은 할당량이 적용됩니다. AWS KMS`Decrypt` 자세한 내용은 각 API 작업에 대한 요청 할당량을 참조하십시오. AWS KMS

AWS KMS **`Decrypt`**

할당량 이름	기본값(초당 요청 수)

암호화 작업(대칭) 요청 빈도	이러한 공유 할당량은 요청에 사용된 AWS KMS 키 유형 AWS 리전 및 키에 따라 달라집니다. 각 할당량은 별도로 계산됩니다.
	• 5500(공유)
	• 다음 리전에서는 10000(공유):
	  • 미국 동부(오하이오) us-east-2
	  • 아시아 태평양(싱가포르) ap-southeast-1
	  • 아시아 태평양(시드니) ap-southeast-2

할당량 이름	기본값(초당 요청 수)
	• 아시아 태평양(도쿄) ap-northeast-1 • EU(프랑크푸르트) eu-central-1 • EU(런던) eu-west-2 • 다음 리전에서는 50000(공유): • 미국 동부(버지니아 북부), us-east-1 • 미국 서부(오리건), us-west-2 • 유럽(아일랜드) eu-west-1
사용자 지정 키 스토어 요청 할당량 Note 이 할당량은 외부 키 저장소를 사용하는 경우에만 적용됩니다.	사용자 지정 키 저장소 요청 할당량은 각 사용자 지정 키 저장소에 대해 별도로 계산됩니다. • 각 키 스토어당 1,800개 (공유) AWS CloudHSM • 각 외부 키 저장소에 대해 1800회(공유)

알려진 제한 사항

- 서버측 TCP keep-Alive는 지원되지 않습니다. 클라이언트가 keep-alive 패킷을 전송하지 않는 한 350초 동안 사용하지 않으면 연결 제한 시간이 초과됩니다.
- 서비스에서 활성 계약을 수락하고 Amazon CloudWatch 로그에 나타나려면 메시지에 유효한 AS2 헤더가 포함되어야 합니다.
- AWS Transfer Family AS2용 메시지를 수신하는 서버는 RFC 6211에 정의된 대로 메시지 서명을 검증하기 위한 CMS (암호화 메시지 구문) 알고리즘 보호 속성을 지원해야 합니다. 이 속성은 일부 구형 IBM Sterling 제품에서는 지원되지 않습니다.
- 메시지 ID가 중복되면 처리됨/경고: 문서 중복 메시지가 표시됩니다.
- AS2 인증서의 키 길이는 최소 2,048비트, 최대 4,096비트여야 합니다.
- AS2 메시지 또는 비동기 mDN을 거래 파트너의 HTTPS 엔드포인트로 보내는 경우 메시지 또는 mDN은 공개적으로 신뢰할 수 있는 인증 기관 (CA) 에서 서명한 유효한 SSL 인증서를 사용해야 합니다. 자체 서명된 인증서는 현재 아웃바운드 전송에만 지원됩니다.
- 엔드포인트는 TLS 버전 1.2 프로토콜과 보안 정책에서 허용하는 암호화 알고리즘을 (서버 보안 정책 AWS Transfer Family에서 설명한 것과 같이) 지원해야 합니다.
- AS2 버전 1.2의 다중 첨부 파일 및 CEM (인증서 교환 메시징) 은 현재 지원되지 않습니다.

- 기본 인증은 현재 아웃바운드 메시지에만 지원됩니다.

AS2 특성 및 기능

다음 표에는 AS2를 사용하는 Transfer Family 리소스에 사용할 수 있는 기능이 나열되어 있습니다.

AS2 특성

Transfer Family는 AS2에 다음과 같은 특성을 제공합니다.

기능	지원 대상: AWS Transfer Family
드럼몬드 인증	예
AWS CloudFormation 지원	예
아마존 CloudWatch 메트릭스	예
SHA-2 암호화 알고리즘	예
아마존 S3 지원	예
Amazon EFS 지원	아니요
예약 메시지	예 [1]
AWS Transfer Family 관리형 워크플로우	아니요
인증서 교환 메시징 (CEM)	아니요
상호 TLS (MTL)	아니요
자체 서명 인증서 지원	예

1. Amazon을 사용한 예약 AWS Lambda 기능을 통해 사용할 수 있는 아웃바운드 예약 메시지 EventBridge

AS2 송신 및 수신 기능

다음 표는 AWS Transfer Family AS2 송신 및 수신 기능 목록을 제공합니다.

기능	인바운드: 서버를 통한 수신	아웃바운드: 커넥터를 사용하여 전송
TLS 암호화 전송 (HTTPS)	예 [1]	예
비 TLS 전송 (HTTP)	예	예 [2]
동기식 MDN	예	예
메시지 압축	예	예
비동기식 MDN	예	아니요
고정 IP 주소	예	예
자체 IP 주소 가져오기	예	아니요
여러 파일 첨부	아니요	아니요
기본 인증	아니요	예
AS2 재시작	해당 사항 없음	아니요
AS2 신뢰성	아니요	아니요
메시지별 사용자 지정 제목	해당 사항 없음	아니요

1. Network Load Balancer (NLB)와 함께 사용 가능한 인바운드 TLS 암호화 전송

2. 아웃바운드 비 TLS 전송은 암호화가 활성화된 경우에만 사용할 수 있습니다.

AS2 커넥터 구성

커넥터의 목적은 아웃바운드 전송을 위해 거래 파트너들 사이이의 관계를 설정하는 것입니다. 즉, Transfer Family 서버에서 파트너 소유의 외부 대상으로 AS2 파일을 보내는 것입니다. 커넥터의 경우 로컬 및 파트너 프로필을 생성하여 로컬 당사자, 원격 파트너 및 인증서를 지정합니다.

커넥터를 설치한 후에는 거래 파트너에게 정보를 전송할 수 있습니다. 각 AS2 서버에는 세 개의 고정 IP 주소가 할당됩니다. AS2 커넥터는 이러한 IP 주소를 사용하여 AS2를 통해 거래 파트너에게 비동기 mDN을 전송합니다.

> **Note**
>
> 거래 파트너가 수신한 메시지 크기는 Amazon S3의 객체 크기와 일치하지 않습니다. AS2 메시지가 파일을 보내기 전에 봉투에 싸서 보내기 때문에 이러한 불일치가 발생합니다. 따라서 파일을 압축하여 전송하더라도 파일 크기가 커질 수 있습니다. 따라서 거래 파트너의 최대 파일 크기가 보내는 파일 크기보다 커야 합니다.

AS2 커넥터 생성

이 절차에서는 콘솔을 사용하여 AS2 커넥터를 생성하는 방법을 설명합니다. AWS Transfer Family 를 AWS CLI 대신 사용하려면 을 참조하십시오the section called “6단계: 사용자와 사용자의 거래 파트너 간의 커넥터를 생성합니다.”.

AS2 커넥터를 생성하려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 커넥터를 선택한 다음 커넥터 생성을 선택합니다.
3. 커넥터 구성 섹션에서 다음 정보를 지정합니다:

   - URL - 아웃바운드 연결의 URL을 입력합니다.
   - 액세스 역할 — 사용할 (IAM) 역할의 Amazon 리소스 이름 AWS Identity and Access Management (ARN) 을 선택합니다. 이 역할이 `StartFileTransfer` 요청에서 사용된 파일 위치의 상위 디렉터리에 대한 읽기 및 쓰기 액세스 권한을 제공하는지 확인하세요. 또한 그 역할이 `StartFileTransfer`와 함께 전송하려는 파일의 상위 디렉터리에 대한 읽기 및 쓰기 액세스 권한을 제공하는 지 확인하세요.

     > **Note**
     >
     > 커넥터에 기본 인증을 사용하는 경우 액세스 역할에는 암호에 대한 `secretsmanager:GetSecretValue` 권한이 필요합니다. AWS 관리형 키 in AWS Secrets Manager대신 고객 관리 키를 사용하여 암호를 암호화하는 경우 역할에도 해당 키에 대한 `kms:Decrypt` 권한이 필요합니다. 접두사 `aws/transfer/`를 사용하여 암호의 명칭을 지정하는 경우 암호 생성 권한 예에 나와 있는 것처럼 와일드카드 문자(*)를 사용하여 필요한 권한을 추가할 수 있습니다.

   - 로깅 역할 (선택 사항) - 이벤트를 CloudWatch 로그로 푸시하는 데 사용할 커넥터의 IAM 역할을 선택합니다.

4. AS2 구성 섹션에서 로컬 및 파트너 프로필, 암호화 및 서명 알고리즘, 전송된 정보의 압축 여부를 선택합니다. 유의할 사항:

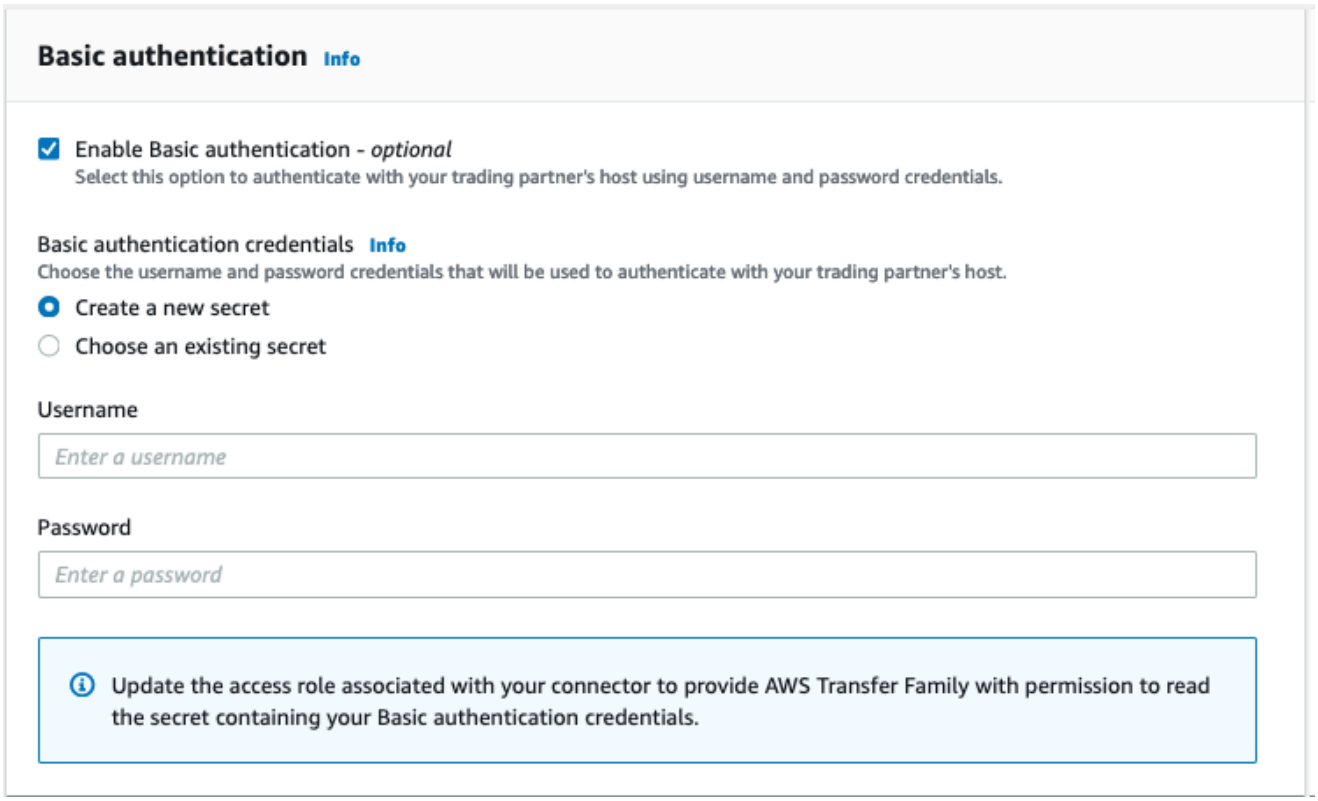
   - 암호화 알고리즘은 암호화 알고리즘이 취약하기 때문에 암호화 알고리즘이 필요한 레거시 클라이언트를 지원해야 하는 DES_EDE3_CBC 경우가 아니면 선택하지 마십시오.
   - 주제는 커넥터와 함께 전송되는 AS2 메시지의 subject HTTP 헤더 속성으로 사용됩니다.
   - 암호화 알고리즘 없이 커넥터를 생성하려는 경우 HTTPS 프로토콜로 지정해야 합니다.

5. MDN 구성 섹션에서 다음 정보를 지정합니다.

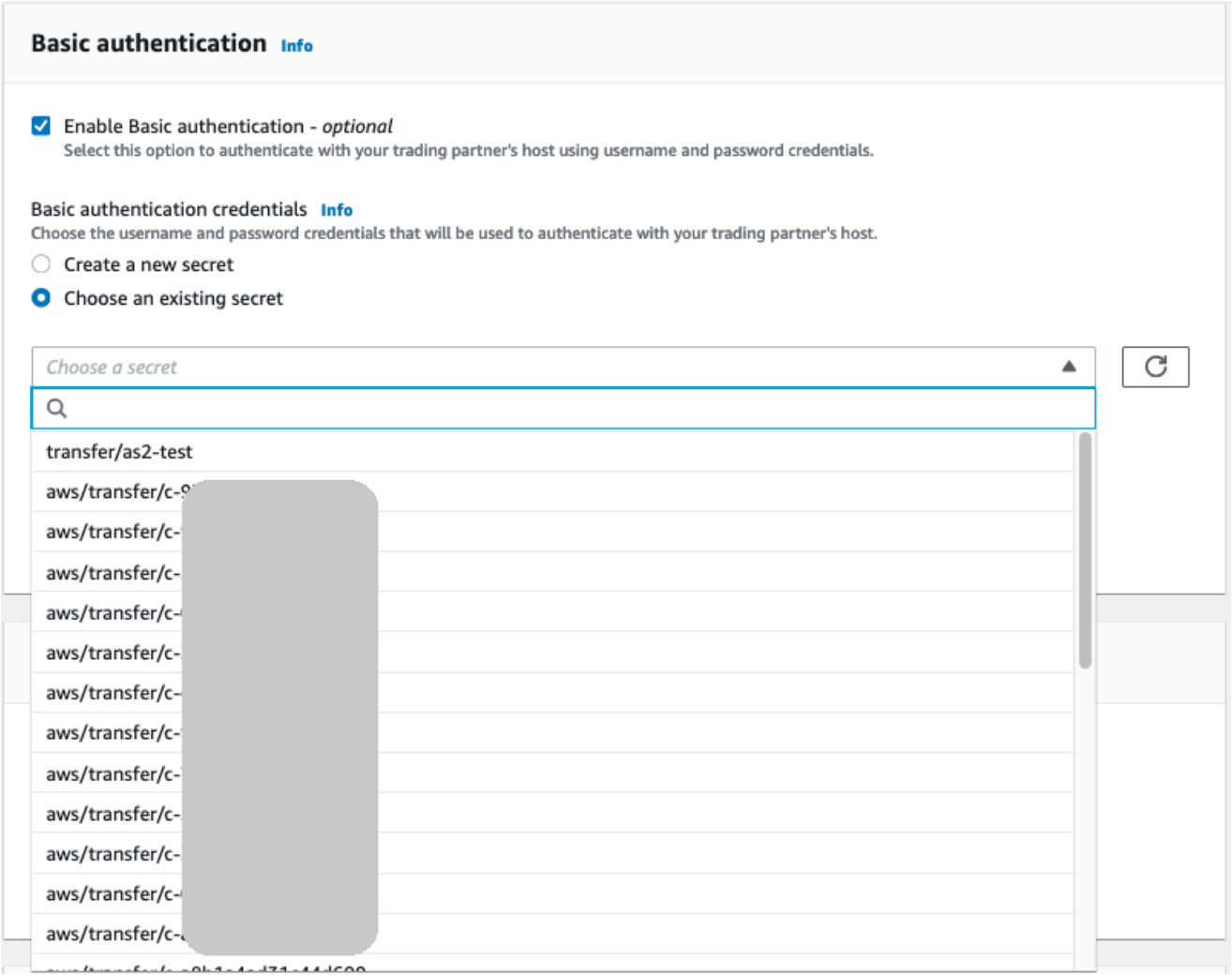
   - MDN 요청 - 거래 파트너가 AS2를 통해 메시지를 성공적으로 수신한 후 MDN을 보내도록 요청할 수 있습니다.
   - 서명된 MDN - MDN에 서명을 요구하는 옵션이 있습니다. 이 옵션은 MDN 요청을 선택한 경우에만 사용할 수 있습니다.

6. 기본 인증 섹션에서 다음 정보를 지정합니다.

   - 아웃바운드 메시지와 함께 로그온 자격 증명을 보내려면 기본 인증 활성화를 선택합니다. 아웃바운드 메시지와 함께 자격 증명을 보내지 않으려면 기본 인증 활성화를 선택 취소하세요.
   - 인증을 사용하는 경우 암호를 선택하거나 생성하세요.
     - 새 암호를 만들려면 새 암호 생성을 선택한 다음 사용자 이름과 암호를 입력합니다. 이러한 자격 증명은 파트너의 엔드포인트에 연결하는 사용자와 일치해야 합니다.



- 암호를 사용하려면 기존 암호 선택을 선택한 후 드롭다운 메뉴에서 암호를 선택합니다. Secrets Manager에서 올바른 형식의 암호를 만드는 방법에 대한 자세한 설명은 AS2 커넥터에 대한 기본 인증을 활성화합니다. 섹션을 참조하세요.



7. 모든 설정을 확인한 후 커넥터 생성을 선택하여 커넥터를 생성합니다.

새 커넥터의 ID가 목록에 추가된 커넥터 페이지가 나타납니다. 커넥터에 대한 세부 정보를 보려면 AS2 커넥터 세부 정보 보기 섹션을 참조하세요.

AS2 커넥터 알고리즘

AS2 커넥터를 생성하면 다음 보안 알고리즘이 커넥터에 연결됩니다.

유형	알고리즘
TLS 암호	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256

유형	알고리즘
	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

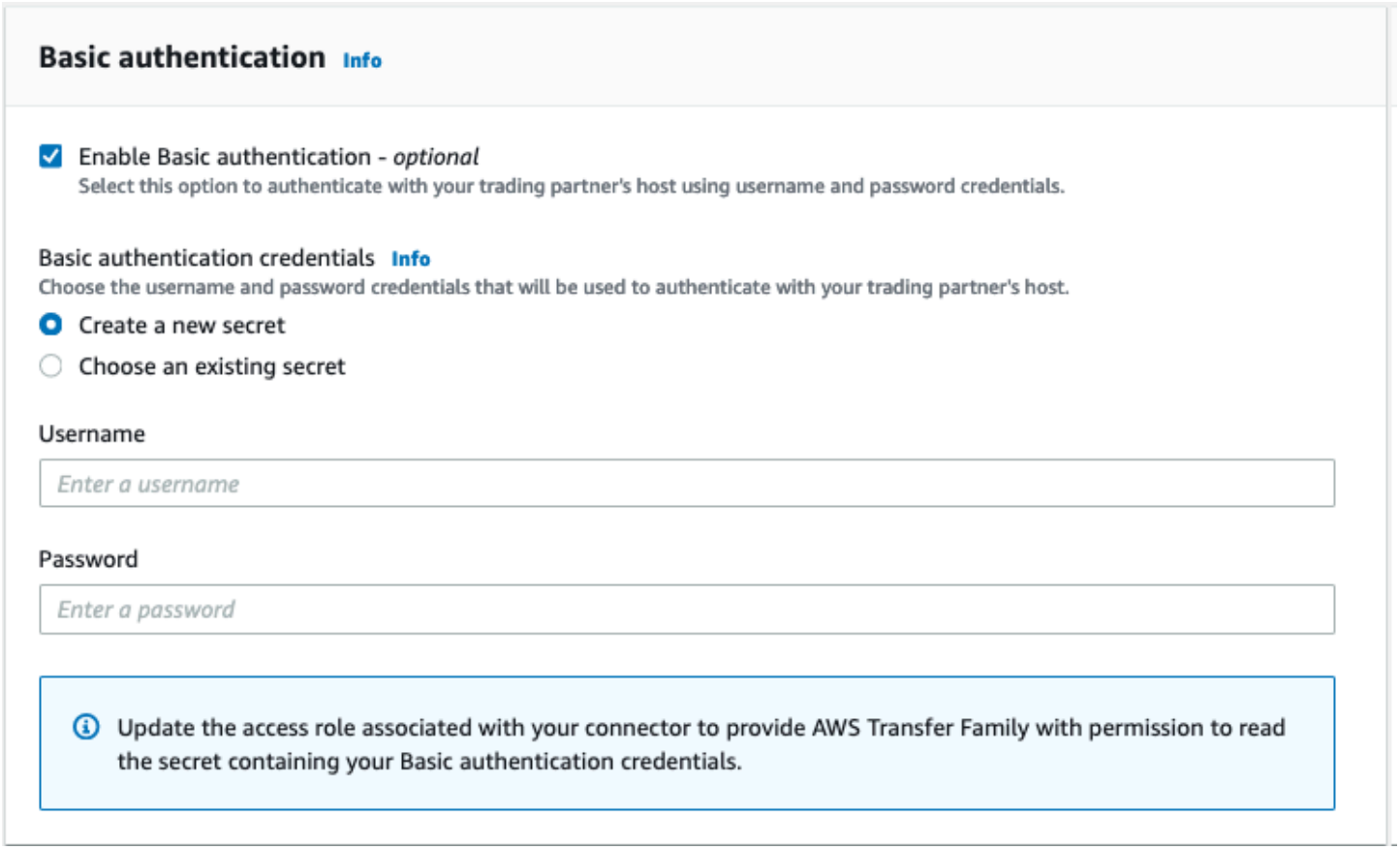
AS2 커넥터의 기본 인증

AS2 프로토콜을 사용하는 Transfer Family 서버를 만들거나 업데이트할 때 아웃바운드 메시지에 대한 기본 인증을 추가할 수 있습니다. 커넥터에 인증 정보를 추가하여 이 작업을 수행합니다.

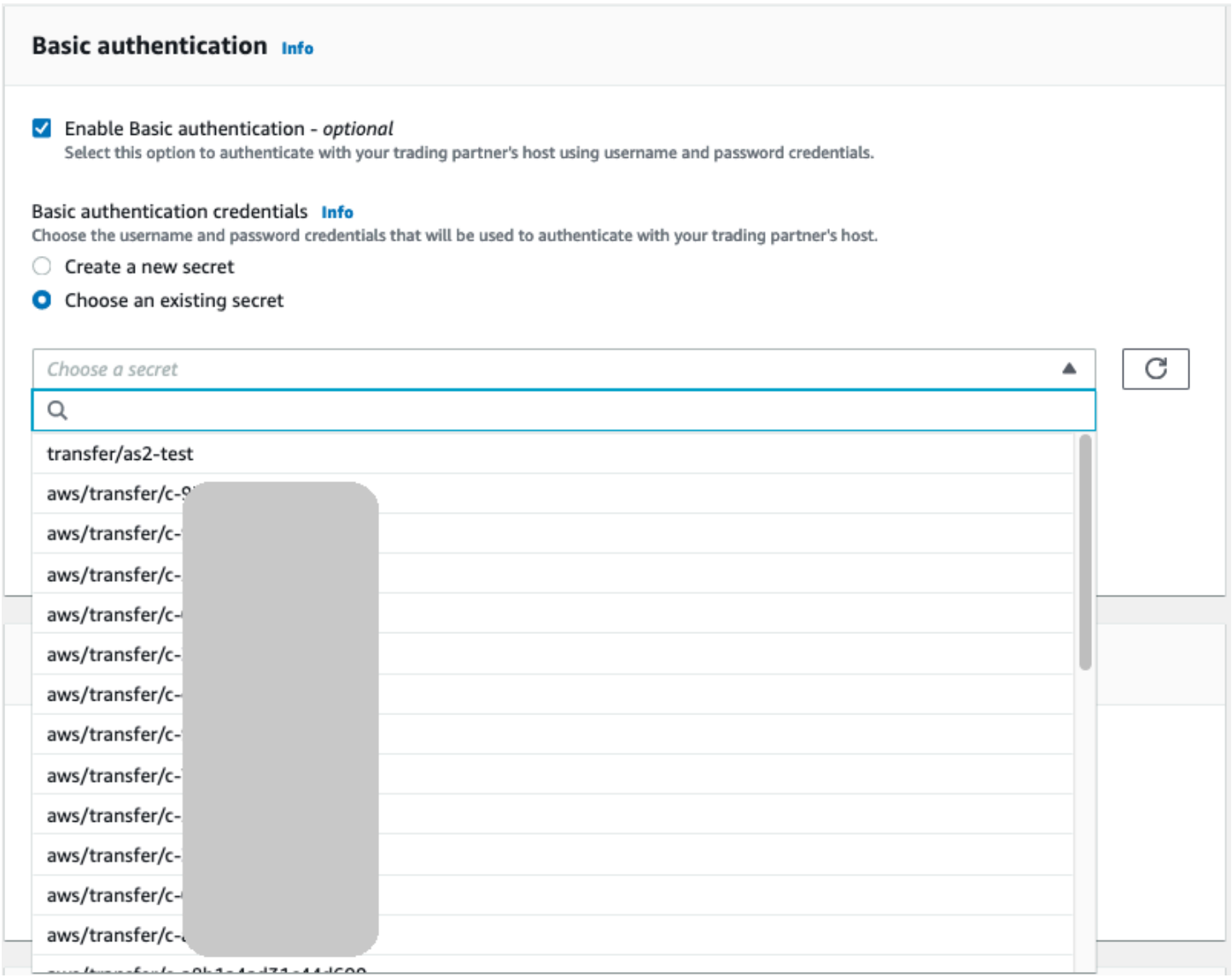
> **Note**
>
> 기본 인증은 HTTPS를 사용하는 경우에만 사용할 수 있습니다.

커넥터에 인증을 사용하려면 기본 인증 섹션에서 기본 인증 활성화를 선택합니다. 기본 인증을 활성화한 후 새 비밀을 생성하거나 기존 비밀을 사용할 수 있습니다. 어느 경우든 비밀의 자격 증명은 이 커넥터를 사용하는 아웃바운드 메시지와 함께 전송됩니다. 자격 증명은 거래 파트너의 원격 엔드포인트에 연결하려는 사용자와 일치해야 합니다.

다음 스크린샷은 기본 인증 활성화를 선택하고 새 비밀 만들기를 선택한 것을 보여줍니다. 이러한 선택을 한 후 비밀에 대한 사용자 이름과 암호를 입력할 수 있습니다.



다음 스크린샷은 기본 인증 활성화를 선택하고 기존 비밀을 선택한 것을 보여줍니다. 비밀은 AS2 커넥터에 대한 기본 인증을 활성화합니다.에 설명된 대로 올바른 형식이어야 합니다.



AS2 커넥터에 대한 기본 인증을 활성화합니다.

AS2 커넥터에 대한 기본 인증을 활성화하면 Transfer Family 콘솔에서 새 암호를 만들거나 AWS Secrets Manager에서 만든 암호를 사용할 수 있습니다. 어느 경우든 암호는 Secrets Manager에 저장됩니다.

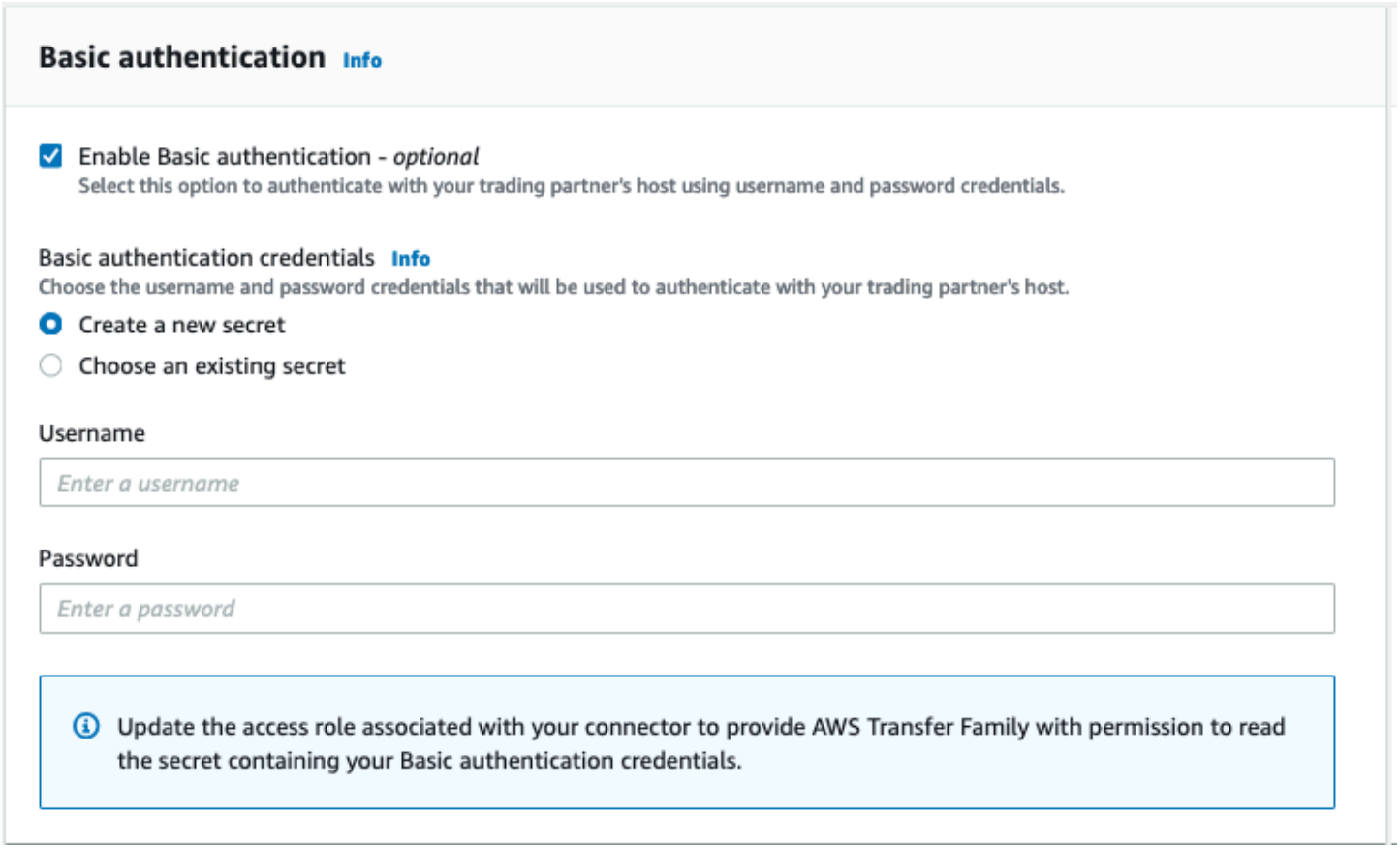
주제

- 콘솔을 사용하여 새 암호 생성
- 기존 암호 사용
- 에서 시크릿 생성 AWS Secrets Manager

콘솔을 사용하여 새 암호 생성

콘솔에서 커넥터를 만들 때 새 암호를 생성할 수 있습니다.

새 암호를 만들려면 새 암호 생성을 선택한 다음 사용자 이름과 암호를 입력합니다. 이러한 자격 증명은 파트너의 엔드포인트에 연결하는 사용자와 일치해야 합니다.

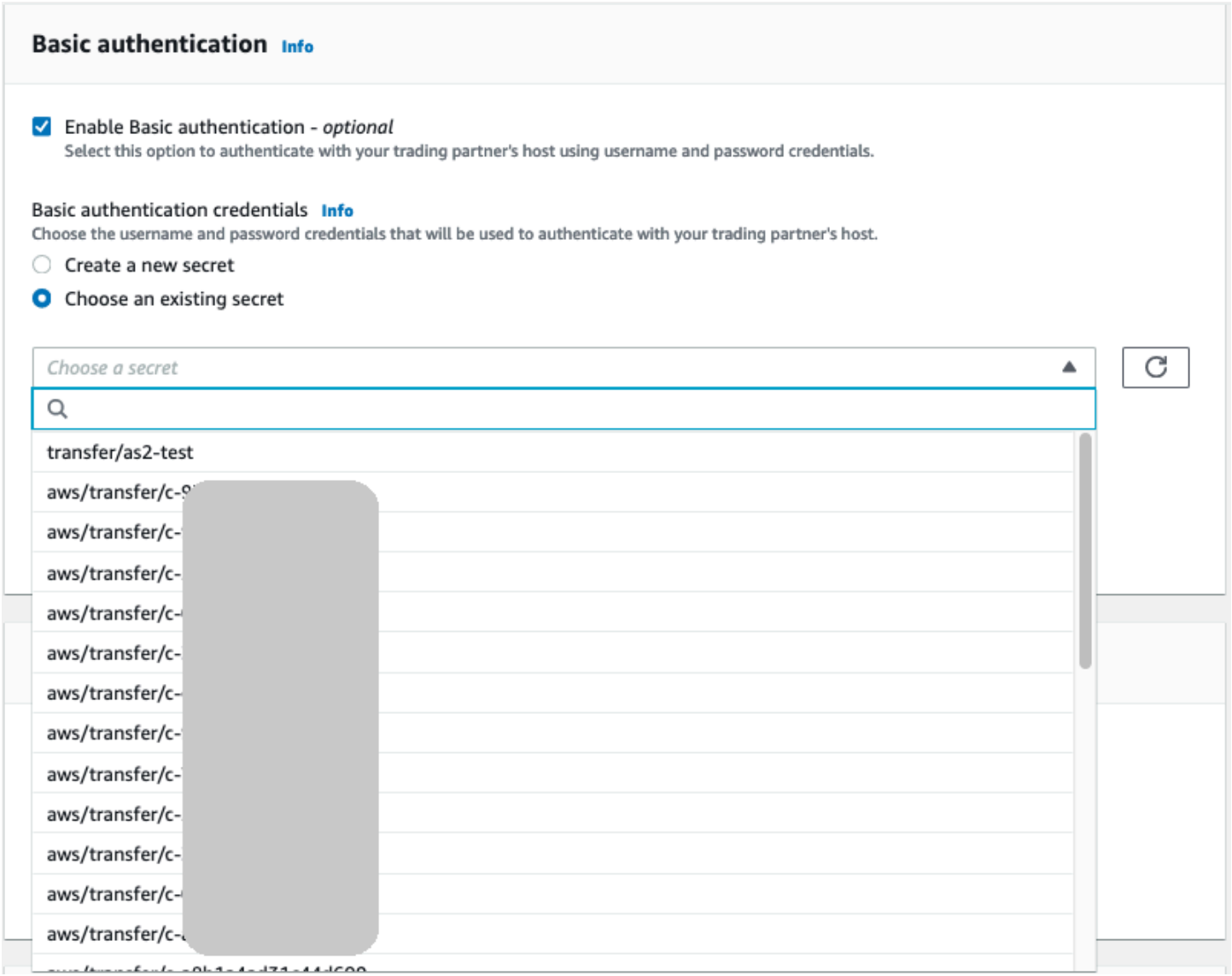


> **Note**
>
> 콘솔에서 새 암호를 만들면 암호의 명칭은 **/aws/transfer/*connector-id***과 같은 명명 규칙을 따릅니다. 여기서 *connector-id*는 만들려는 커넥터의 ID입니다. AWS Secrets Manager에서 암호를 찾으려고 할 때는 이 점을 고려하세요.

기존 암호 사용

콘솔에서 커넥터를 만들 때 기존 암호를 지정할 수 있습니다.

암호를 사용하려면 기존 암호 선택을 선택한 후 드롭다운 메뉴에서 암호를 선택합니다. Secrets Manager에서 올바른 형식의 암호를 만드는 방법에 대한 자세한 설명은 에서 시크릿 생성 AWS Secrets Manager 섹션을 참조하세요.



에서 시크릿 생성 AWS Secrets Manager

다음 절차는 AS2 커넥터에 사용할 적절한 암호를 만드는 방법을 설명합니다.

> **Note**
>
> 기본 인증은 HTTPS를 사용하는 경우에만 사용할 수 있습니다.

AS2 Basic 인증을 위해 Secrets Manager에 사용자 자격 증명을 저장하려면

1. https://console.aws.amazon.com/secretsmanager/ 에서 AWS Management Console 로그인하고 AWS Secrets Manager 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 암호를 선택합니다.
3. 암호 페이지에서 새 암호 저장을 선택합니다.
4. 암호 선택 페이지의 암호 타입에서 다른 타입의 암호를 선택합니다.
5. 키/값 쌍 섹션에서 키/값 탭을 선택합니다.

   - 키 - **`Username`**를 입력합니다.
   - 값 - 파트너 서버에 연결할 권한이 있는 사용자의 이름을 입력합니다.
6. 암호를 제공하려면 행 추가를 선택하고 키/값 쌍 섹션에서 키/값 탭을 선택합니다.

   행 추가를 선택하고 키/값 쌍 섹션에서 키/값 탭을 선택합니다.

   - 키 - **`Password`**를 입력합니다.
   - 값 – 사용자의 암호를 입력합니다.
7. 프라이빗 키를 제공하려면 행 추가를 선택하고 키/값 쌍 섹션에서 키/값 탭을 선택합니다.

   - 키 - **`PrivateKey`**를 입력합니다.
   - 값 - 사용자의 프라이빗 키를 입력합니다. 이 값은 OpenSSH 형식으로 저장되어야 하며 원격 서버에 이 사용자에 대해 저장된 퍼블릭 키와 일치해야 합니다.
8. 다음을 선택합니다.
9. 암호 구성 페이지에 암호를 위한 명칭과 설명을 입력합니다. 이름에 **`aws/transfer/`**이라는 접두사를 사용하는 것이 좋습니다. 예컨대, 암호 **`aws/transfer/connector-1`**을 지정할 수 있습니다.
10. 다음을 선택한 후 교체 구성 페이지의 기본값을 그대로 사용합니다. 그 다음 다음을 선택합니다.
11. 검토 페이지에서 저장을 선택하여 암호를 만들고 저장합니다.

암호를 만든 후 커넥터를 만들 때 암호를 선택할 수 있습니다(AS2 커넥터 구성 참조). 기본 인증을 활성화하는 단계에서 사용 가능한 암호의 드롭다운 목록에서 암호를 선택합니다.

AS2 커넥터 세부 정보 보기

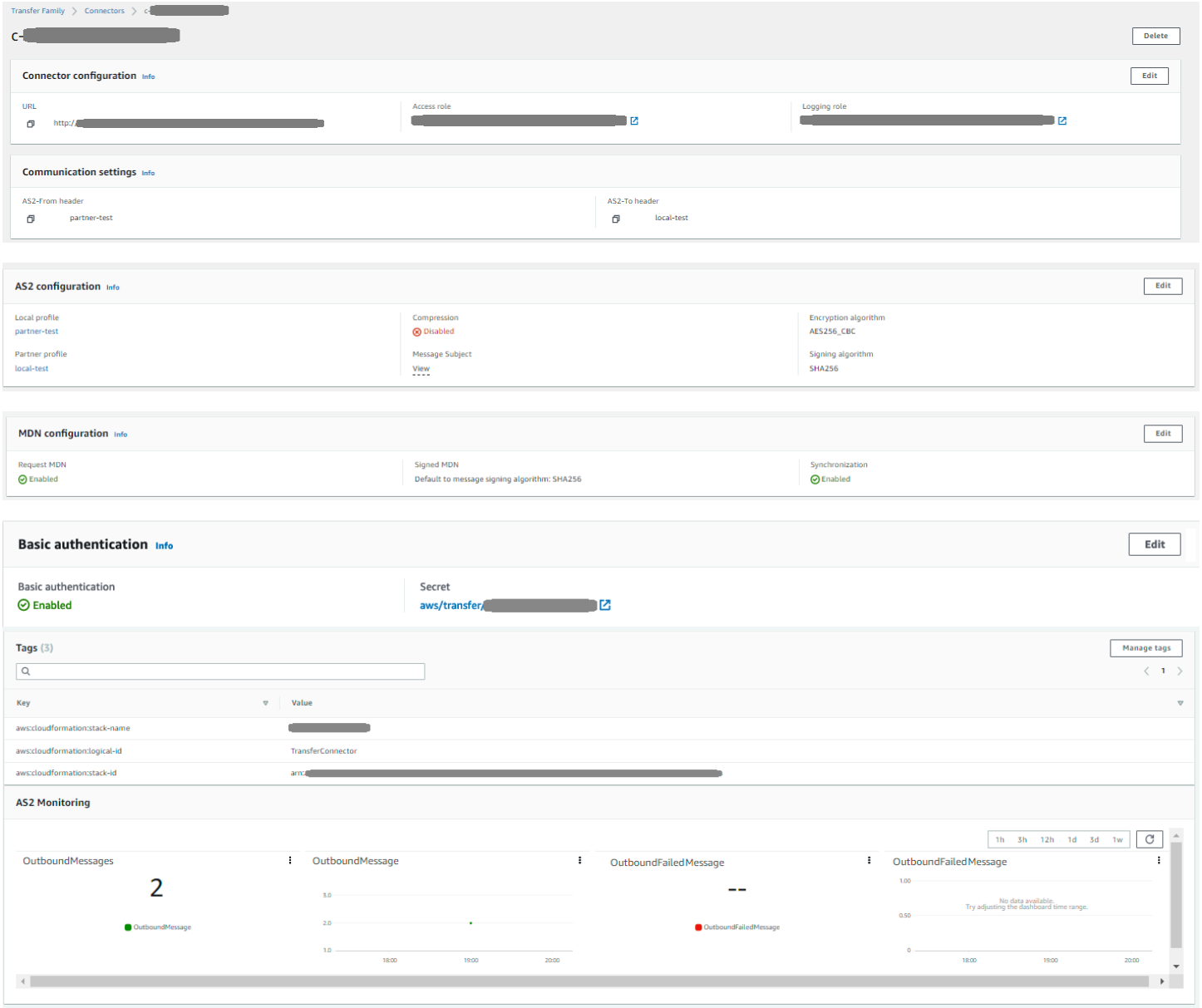
AWS Transfer Family 콘솔에서 AS2 AWS Transfer Family 커넥터의 세부 정보 및 속성 목록을 찾을 수 있습니다. AS2 커넥터의 속성에는 URL, 역할, 프로필, MDN, 태그 및 모니터링 지표가 포함됩니다.

이것은 커넥터 세부 정보 보기 절차입니다.

커넥터 세부 정보를 보려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 커넥터를 선택합니다.
3. 커넥터 ID 열에서 식별자를 선택하면 선택한 커넥터의 세부 정보 페이지가 표시됩니다.

편집을 선택하여 커넥터의 세부 정보 페이지에서 AS2 커넥터의 속성을 변경할 수 있습니다.



> **Note**
>
> 형식은 다르지만 다음 AWS Command Line Interface 를 실행하여 이 정보의 대부분을 얻을 수 있습니다 (AWS CLI 명령:
>
> ```
> aws transfer describe-connector --connector-id your-connector-id
> ```
>
> 자세한 내용은 API 참조의 DescribeConnector를 참조하세요.

AS2 파트너 관리

이 항목에서는 AS2 인증서, 프로필 및 계약을 관리하는 방법에 대해 설명합니다.

AS2 인증서 가져오기

Transfer Family AS2 프로세스는 전송된 정보의 암호화와 서명 모두에 인증서 키를 사용합니다. 파트너는 두 가지 용도로 동일한 키를 사용하거나 각 용도에 별도의 키를 사용할 수 있습니다. 재해 또는 보안 침해 발생 시 데이터를 복호화할 수 있도록 신뢰할 수 있는 제3자가 에스크로에 공통 암호화 키를 보관해 둔 경우 별도의 서명 키를 사용하는 것이 좋습니다. 별도의 서명 키(에스크로하지 않음)를 사용하면 디지털 서명의 부인 방지 기능이 손상되지 않습니다.

> **Note**
>
> AS2 인증서의 키 길이는 최소 2,048비트, 최대 4,096비트여야 합니다.

다음 사항은 프로세스 중에 AS2 인증서가 사용되는 방법을 자세히 설명합니다.

- 인바운드 AS2
  - 거래 파트너가 서명 인증서용 퍼블릭 키를 보내면 이 키를 파트너 프로필로 가져옵니다.
  - 현지 당사자가 암호화 및 서명 인증서를 위한 퍼블릭 키를 전송합니다. 그러면 파트너가 프라이빗 키를 가져옵니다. 로컬 당사자는 서명 및 암호화를 위해 별도의 인증서 키를 보내거나 두 가지 용도로 동일한 키를 사용하도록 선택할 수 있습니다.
- 아웃바운드 AS2
  - 파트너가 암호화 인증서용 퍼블릭 키를 보내고 이 키를 파트너 프로필로 가져옵니다.

- 로컬 당사자는 서명용 인증서의 퍼블릭 키를 보내고 서명을 위해 인증서의 프라이빗 키를 가져옵니다.
- HTTPS를 사용하는 경우 자체 서명된 전송 계층 보안 (TLS) 인증서를 가져올 수 있습니다.

인증서 생성 방법에 대한 자세한 설명은 the section called “1단계: AS2용 인증서 생성” 섹션을 참조하세요.

이 절차에서는 Transfer Family 콘솔을 사용하여 인증서를 가져오는 방법을 설명합니다. AWS CLI 대신 사용하려면 을 참조하십시오. the section called “3단계: Transfer Family 인증서 리소스로 인증서 가져오기”

AS2 지원 인증서를 지정하려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 왼쪽 탐색 창의 AS2 거래 파트너에서 인증서를 선택합니다.
3. 인증서 가져오기를 선택합니다.
4. 인증서 설명 섹션에서 쉽게 식별할 수 있는 인증서 명칭을 입력합니다. 설명으로 인증서의 용도를 식별할 수 있는지 확인하세요. 또한 인증서의 역할을 선택합니다.
5. 인증서 내용 섹션에서 거래 파트너의 공개 인증서를 제공하거나 로컬 인증서용 공개 및 프라이빗 키를 제공하세요.
6. 인증서 용도 섹션에서 이 인증서의 용도를 선택합니다. 암호화, 서명 또는 둘 다에 사용할 수 있습니다.

   > **Note**
   >
   > 사용량에 대해 암호화 및 서명을 선택하면 Transfer Family는 두 개의 동일한 인증서(각각 자체 ID가 있음)를 생성합니다. 하나는 사용 값이 `ENCRYPTION`이고 다른 하나는 사용 값이 `SIGNING`입니다.

7. 인증서 내용 섹션에 적절한 세부 정보를 입력합니다.
   - 자체 서명 인증서를 선택하는 경우 인증서 체인을 제공하지 않습니다.
   - 인증서 내용을 붙여 넣습니다.
   - 인증서가 자체 서명된 인증서가 아닌 경우 인증서 체인을 제공합니다.
   - 이 인증서가 로컬 인증서인 경우 프라이빗 키를 붙여 넣습니다.
8. 인증서 가져오기를 선택하여 프로세스를 완료하고 가져온 인증서의 세부 정보를 저장합니다.

> **Note**
>
> TLS 인증서는 파트너의 공개 인증서로만 가져올 수 있습니다. 파트너의 공개 인증서를 선택한 다음 사용 대상으로 전송 계층 보안 (TLS) 을 선택하면 경고 메시지가 표시됩니다. 또한 TLS 인증서는 자체 서명되어야 합니다. 즉, TLS 인증서를 가져오려면 자체 서명된 인증서를 선택해야 합니다.

AS2 인증서 교체

인증서는 6개월에서 1년 동안 유효한 경우가 많습니다. 더 오래 유지하려는 프로필을 설정했을 수 있습니다. 이를 용이하게 하기 위해 Transfer Family는 인증서 교체 서비스를 제공합니다. 프로필에 여러 인증서를 지정하여 프로필을 여러 해 동안 계속 사용할 수 있습니다. Transfer Family는 서명(옵션) 및 암호화(필수)에 인증서를 사용합니다. 원하는 경우 두 가지 용도로 사용할 단일 인증서를 지정할 수 있습니다.

인증서 교체는 만료되는 오래된 인증서를 새 인증서로 교체하는 프로세스입니다. 이 전환은 계약 파트너가 아직 아웃바운드 전송을 위한 새 인증서를 구성하지 않았거나 새 인증서를 사용할 수 있는 기간 중에 이전 인증서로 서명 또는 암호화된 페이로드를 보내는 경우 전송이 중단되지 않도록 점진적으로 진행됩니다. 기존 인증서와 새 인증서가 모두 유효한 중간 기간을 유예 기간이라고 합니다.

X.509 인증서에는 `Not Before` 및 `Not After` 날짜가 있습니다. 하지만 이러한 파라미터는 관리자에게 충분한 통제 기능을 제공하지 못할 수 있습니다. Transfer Family는 아웃바운드 페이로드에 사용되는 인증서와 인바운드 페이로드에 허용되는 인증서를 통제하는 `Active Date` 및 `Inactive Date` 설정을 제공합니다.

아웃바운드 인증서 선택은 전송 날짜 이전의 최대값을 `Inactive Date`로 사용합니다. 인바운드 프로세스는 범위 `Not Before` 및 `Not After` 내 및 범위 `Active Date` 및 `Inactive Date` 내에서 인증서를 수락합니다.

다음 표에서는 단일 프로필에 대해 두 개의 인증서를 구성할 수 있는 한 가지 방법을 설명합니다.

두 개의 인증서 교대

명칭	NOT BEFORE(인증 기관에서 통제)	ACTIVE DATE(Transfer Family에서 설정)	INACTIVE DATE(Transfer Family에서 설정)	NOT AFTER(인증 기관에서 설정)
Cert1(이전 인증서)	2019-11-01	2020-01-01	2020-12-31	2024-01-01
Cert2(최신 인증서)	2020-11-01	2020-06-01	2021-06-01	2025-01-01

유념할 사항:

- 인증서에 `Active Date` 및 `Inactive Date`를 지정하는 경우 범위는 `Not Before` 및 `Not After`의 범위 내에 있어야 합니다.
- 각 프로필에 대해 여러 인증서를 구성하여 모든 인증서의 활성 날짜 범위가 프로필을 사용하려는 기간을 포함하도록 하는 것이 좋습니다.
- 이전 인증서가 비활성화되는 시점과 새 인증서가 활성화되는 시점 사이에 약간의 유예 시간을 지정하는 것이 좋습니다. 위 예에서 첫 번째 인증서는 2020년 12월 31일까지는 비활성화되지 않지만, 두 번째 인증서는 2020년 6월 1일에 활성화되어 6개월의 유예 기간이 제공됩니다. 2020년 6월 1부터 2020년 12월 31일까지의 기간 동안 두 인증서 모두 활성 상태입니다.

AS2 프로필 생성

이 절차를 사용하여 로컬 프로필과 파트너 프로필을 모두 생성할 수 있습니다. 이 절차에서는 Transfer Family 콘솔을 사용하여 AS2 프로필을 생성하는 방법에 대해 설명합니다. 이 방법 대신 AWS CLI 를 사용하려면 the section called “4단계: 사용자와 사용자의 거래 파트너를 위한 프로필 생성” 섹션을 참조하세요.

AS2 프로필을 생성하려면

1. https://console.aws.amazon.com/transfer/ 에서 콘솔을 엽니다. AWS Transfer Family
2. 왼쪽 탐색 창의 AS2 거래 파트너에서 프로필을 선택한 다음 프로필 생성을 선택합니다.
3. 프로필 구성 섹션에서 프로필의 AS2 ID를 입력합니다. 이 값은 AS2 프로토콜별 HTTP 헤더 `as2-from` 및 `as2-to`에 사용되며 거래 파트너십을 식별하고 사용할 인증서를 결정하는 등의 용도로 사용됩니다.

4. 프로필 타입 섹션에서 로컬 프로필 또는 파트너 프로필을 선택합니다.
5. 인증서 섹션의 드롭다운 메뉴에서 하나 이상의 인증서를 선택합니다.

> **Note**
>
> 드롭다운 메뉴에 나열되지 않은 인증서를 가져오려면 새 인증서 가져오기를 선택합니다. 그러면 인증서 가져오기 화면에 새 브라우저 창이 열립니다. 인증서 가져오기 절차에 대한 자세한 설명은 AS2 인증서 가져오기 섹션을 참조하세요.

6. (옵션) 태그 섹션에서 이 프로필을 식별하는 데 도움이 되는 키-값 쌍을 하나 이상 지정합니다.
7. 프로필 생성을 선택하여 프로세스를 완료하고 새 프로필을 저장합니다.

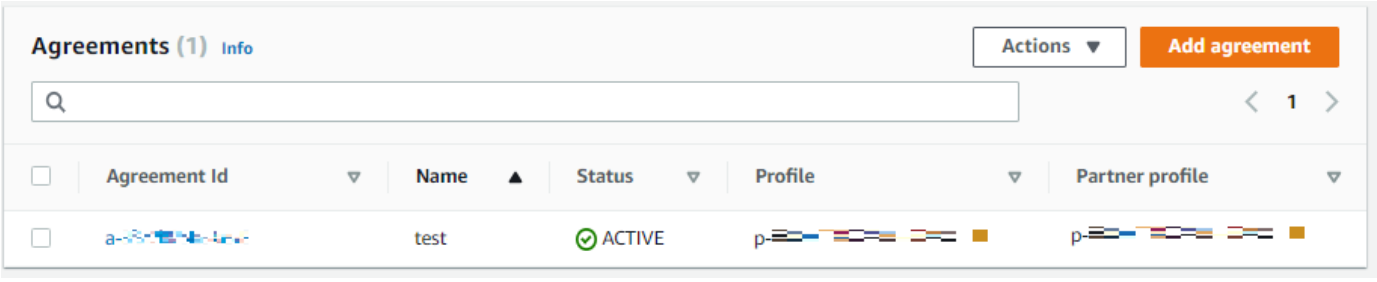
AS2 계약 생성

계약은 Transfer Family 서버와 연결되어 있습니다. AS2 프로토콜을 사용하는 거래 파트너가 Transfer Family를 사용하여 메시지 또는 파일을 교환하는 인바운드 전송, 즉 파트너 소유의 외부 소스에서 Transfer Family 서버로 AS2 파일을 보내는 방식으로 메시지나 파일을 교환하는 거래 파트너에 대한 세부 정보를 지정합니다.

이 절차에서는 Transfer Family 콘솔을 사용하여 AS2 계약서를 생성하는 방법에 대해 설명합니다. AWS CLI 대신 사용하려면 을 참조하십시오the section called “5단계: 사용자와 사용자의 파트너 간의 계약서 작성”.

Transfer Family 서버에 대한 계약을 만들려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 서버를 선택한 다음 AS2 프로토콜을 사용하는 서버를 선택합니다.
3. 서버 세부 정보 페이지에서 계약 섹션으로 스크롤합니다.



4. 계약 추가를 선택합니다.
5. 다음과 같이 계약 파라미터를 입력합니다:

a. 계약 구성 섹션에 설명적인 이름을 입력합니다. 이름으로 계약의 목적을 식별할 수 있는지 확인하세요. 또한 계약 상태를 활성(기본적으로 선택됨) 또는 비활성으로 설정합니다.

b. 통신 구성 섹션에서 로컬 프로필과 파트너 프로필을 선택합니다.

c. 수신함 폴더 구성 섹션에서 수신 파일을 저장할 Amazon S3 버킷과 이 버킷에 액세스할 수 있는 IAM 역할을 선택합니다. 선택적으로, 버킷에 파일을 저장하는 데 사용할 접두사(폴더)를 입력할 수 있습니다.

   예컨대, 버킷에 **DOC-EXAMPLE-BUCKET**, 접두사에 **incoming**를 입력하면 수신된 파일이 `/DOC-EXAMPLE-BUCKET/incoming` 폴더에 저장됩니다.

d. (옵션) 태그 섹션에서 태그를 추가합니다.

e. 계약에 대한 모든 정보를 입력한 후 계약 생성을 선택합니다.

새 계약은 서버 세부 정보 페이지의 계약 섹션에 표시됩니다.

AS2 메시지 전송 및 수신

이 섹션에서는 AS2 메시지를 보내고 받는 프로세스를 설명합니다. 또한 AS2 메시지와 관련된 파일 이름 및 위치에 대한 세부 정보도 제공합니다.

다음 표에는 AS2 메시지에 사용할 수 있는 암호화 알고리즘과 이를 사용할 수 있는 시기가 나와 있습니다.

암호화 알고리즘	HTTP	HTTPS	참고
AES128_CBC	예	예	
AES192_CBC	예	예	
AES256_CBC	예	예	
DES_EDE3_CBC	예	예	이 알고리즘은 암호화 알고리즘이 취약하므로 이 알고리즘이 필요한 레거시 클라이언트를 지원해야 하는 경우에만 사용하십시오.

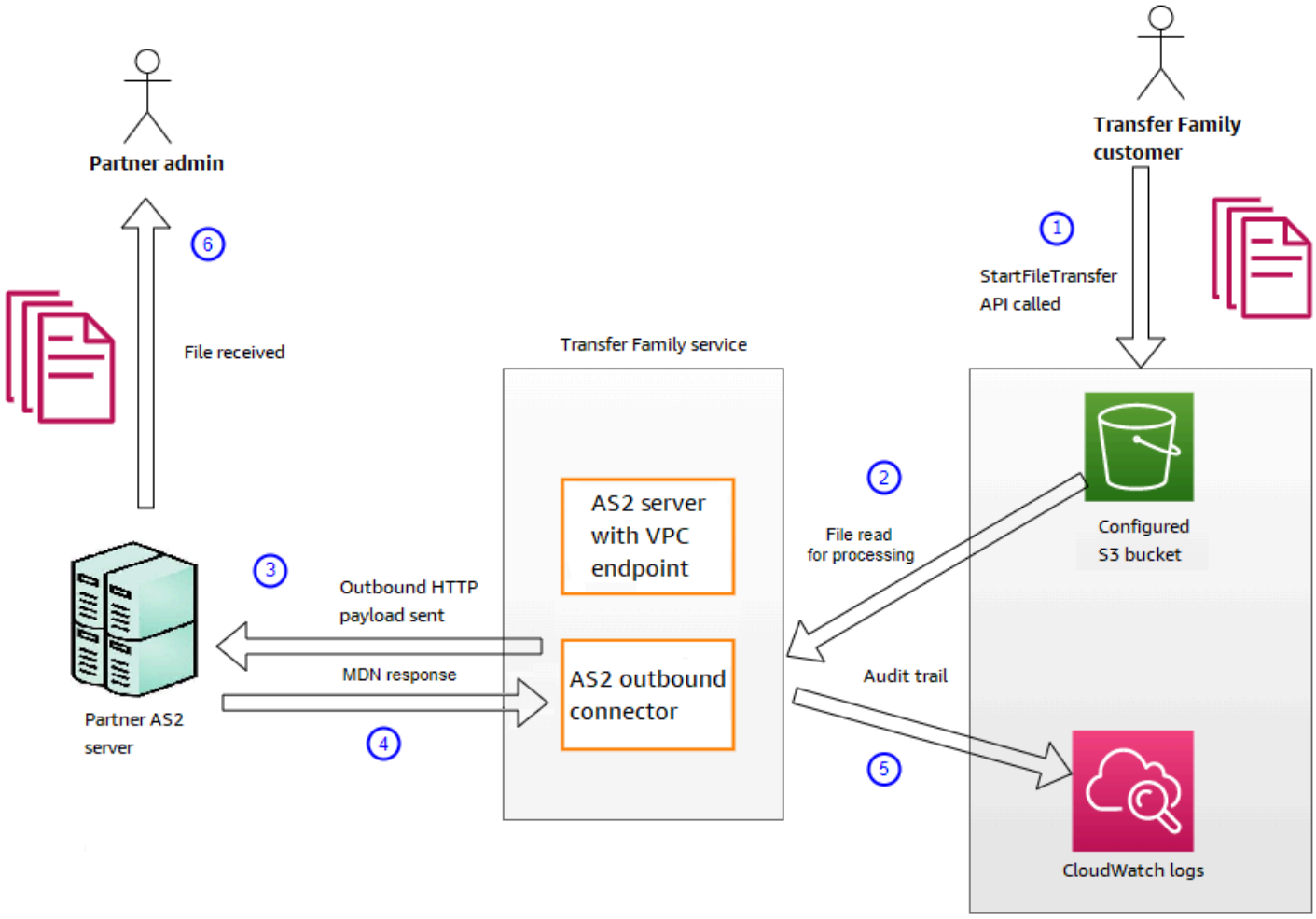
암호화 알고리즘	HTTP	HTTPS	참고
NONE	아니요	예	Transfer Family 서버로 메시지를 보내는 NONE 경우 애플리케이션 로드 밸런서 (ALB)를 사용하는 경우에만 선택할 수 있습니다.

주제

AS2 메시지 전송 프로세스

아웃바운드 프로세스는 외부 클라이언트 또는 서비스에서 AWS 전송되는 메시지 또는 파일로 정의됩니다. 아웃바운드 메시지의 순서는 다음과 같습니다.

1. 관리자가 `start-file-transfer` AWS Command Line Interface (AWS CLI) 명령 또는 `StartFileTransfer` API 작업을 호출합니다. 이 작업은 `connector` 구성을 참조합니다.
2. Transfer Family는 새 파일 요청을 감지하고 파일을 찾습니다. 파일은 압축, 서명 및 암호화됩니다.
3. 전송 HTTP 클라이언트는 HTTP POST 요청을 수행하여 페이로드를 파트너의 AS2 서버로 전송합니다.
4. 프로세스는 HTTP 응답(동기 MDN)과 함께 서명된 MDN 응답을 반환합니다.
5. 파일이 여러 전송 단계 사이를 이동할 때 프로세스는 MDN 응답 수신 및 처리 세부 정보를 고객에게 전달합니다.
6. 원격 AS2 서버는 복호화되고 검증된 파일을 파트너 관리자가 사용할 수 있도록 합니다.



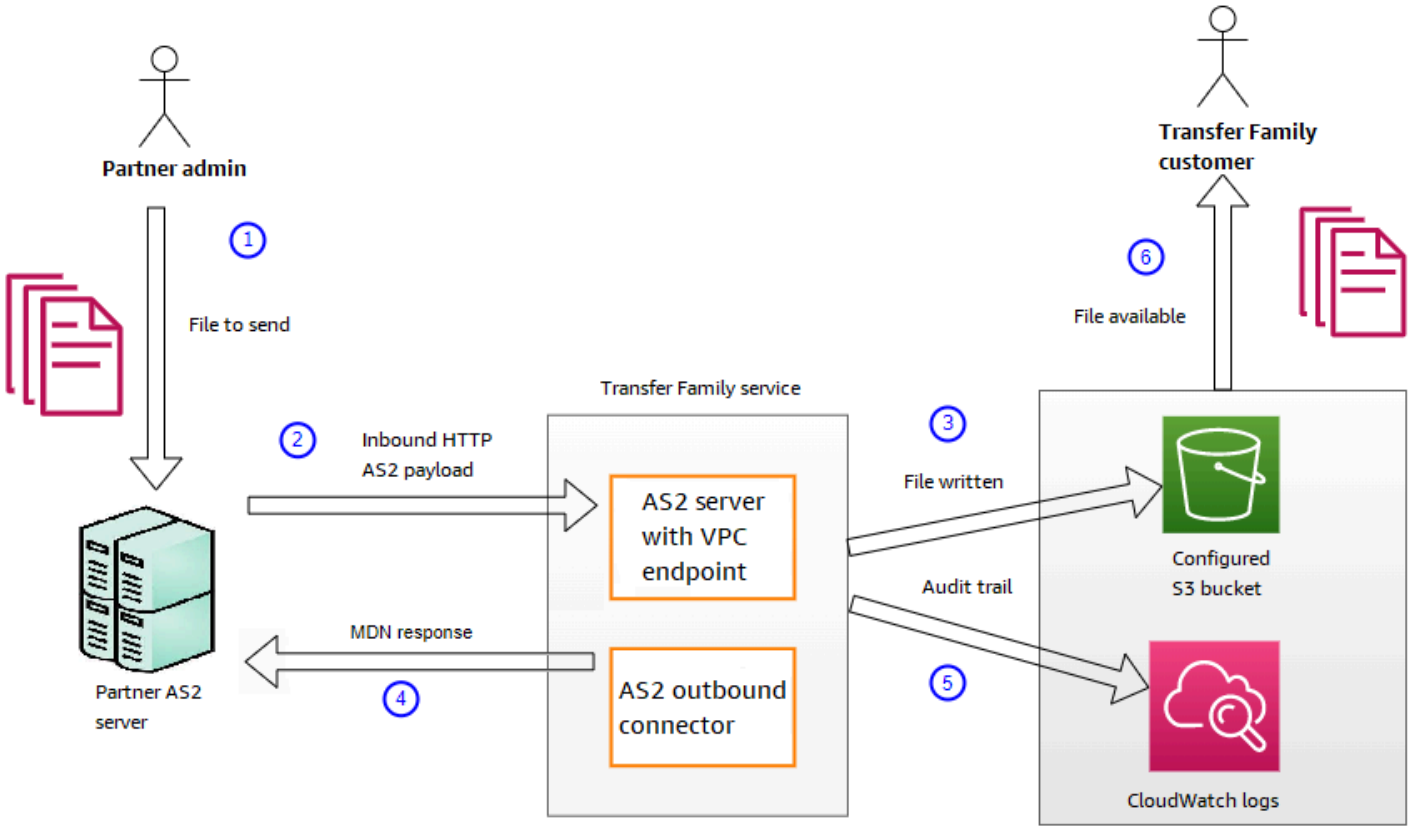
AS2 처리는 일반적인 사용 사례 및 기존 AS2 지원 서버 구현과의 통합에 중점을 두고 많은 RFC 4130 프로토콜을 지원합니다. 지원되는 구성에 대한 자세한 설명은 AS2 지원 구성 섹션을 참조하세요.

AS2 메시지 수신 프로세스

인바운드 프로세스는 서버로 전송되는 메시지 또는 파일로 정의됩니다. AWS Transfer Family 인바운드 메시지의 순서는 다음과 같습니다.

1. 관리자 또는 자동 프로세스는 파트너의 원격 AS2 서버에서 AS2 파일 전송을 시작합니다.
2. 파트너의 원격 AS2 서버가 파일 내용에 서명하고 암호화한 다음 Transfer Family에서 호스팅되는 AS2 인바운드 엔드포인트에 HTTP POST 요청을 보냅니다.
3. Transfer Family는 서버, 파트너, 인증서 및 계약에 대해 구성된 값을 사용하여 AS2 페이로드를 복호화하고 확인합니다. 파일 콘텐츠는 구성된 Amazon S3 파일 스토어에 저장됩니다.
4. 서명된 MDN 응답은 HTTP 응답과 함께 인라인으로 반환되거나 별도의 HTTP POST 요청을 통해 비동기적으로 소스 서버로 반환됩니다.

5. 거래소에 대한 세부 정보가 CloudWatch 포함된 감사 내역이 Amazon에 전송됩니다.
6. 암호가 복호화된 파일은 `inbox/processed`라는 폴더에서 사용할 수 있습니다.



HTTPS를 통한 AS2 메시지 전송 및 수신

이 섹션에서는 AS2 프로토콜을 사용하여 HTTPS를 통해 메시지를 보내고 받는 Transfer Family 서버를 구성하는 방법에 대해 설명합니다.

주제

- HTTPS를 통해 AS2 메시지를 보냅니다.
- HTTPS를 통해 AS2 메시지를 받습니다.

HTTPS를 통해 AS2 메시지를 보냅니다.

HTTPS를 사용하여 AS2 메시지를 보내려면 다음 정보가 포함된 커넥터를 만드세요.

- URL의 경우 HTTPS URL을 지정하세요.

- 암호화 알고리즘의 경우 사용 가능한 알고리즘 중 하나를 선택하십시오.

> **Note**
>
> 암호화를 사용하지 않는 동안 (즉, 암호화 알고리즘을 선택한 NONE 경우) Transfer Family 서버로 메시지를 보내려면 AIB (Application Load Balancer) 를 사용해야 합니다.

- AS2 커넥터 구성에 설명된 대로 커넥터의 나머지 값을 입력합니다.

HTTPS를 통해 AS2 메시지를 받습니다.

AWS Transfer Family AS2 서버는 현재 포트 5080을 통한 HTTP 전송만 제공합니다. 하지만 선택한 포트와 인증서를 사용하여 Transfer Family 서버 VPC 엔드포인트 앞에 있는 네트워크 또는 애플리케이션 로드 밸런서에서 TLS를 종료할 수 있습니다. 이 방법을 사용하면 들어오는 AS2 메시지가 HTTPS를 사용하도록 할 수 있습니다.

사전 조건

- VPC는 Transfer Family AWS 리전 서버와 같아야 합니다.
- VPC의 서브넷은 서버를 사용하려는 가용 영역 내에 있어야 합니다.

> **Note**
>
> 각 Transfer Family 서버는 최대 3개의 가용 영역을 지원할 수 있습니다.

- 서버와 동일한 지역에 엘라스틱 IP 주소를 최대 3개까지 할당하세요. 또는 고유한 IP 주소 범위 (BYOIP) 를 가져오도록 선택할 수 있습니다.

> **Note**
>
> 엘라스틱 IP 주소 수는 서버 엔드포인트에서 사용하는 가용 영역 수와 일치해야 합니다.

NLB (네트워크 로드 밸런서) 또는 애플리케이션 로드 밸런서 (ALB) 를 구성할 수 있습니다. 다음 표에는 각 접근 방식의 장단점이 나와 있습니다.

아래 표에는 NLB와 ALB를 사용하여 TLS를 종료할 때의 기능 차이가 나와 있습니다.

기능	NLB (네트워크 로드 밸런서)	애플리케이션 로드 밸런서 (ALB)
지연 시간	네트워크 계층에서 작동하므로 지연 시간이 짧습니다.	애플리케이션 계층에서 작동하므로 지연 시간이 길어집니다.
고정 IP 지원	고정적일 수 있는 엘라스틱 IP 주소를 연결할 수 있습니다.	엘라스틱 IP 주소를 연결할 수 없음: 기본 IP 주소를 변경할 수 있는 도메인을 제공합니다.
고급 라우팅	고급 라우팅은 지원하지 않습니다.	고급 라우팅을 지원합니다. AS2에 필요한 `X-Forwarded-Proto` 헤더를 암호화 없이 삽입할 수 있습니다. 이 헤더는 developer.mozilla.org 웹 사이트의 X-Forwarded-Proto에 설명되어 있습니다.
TLS/SSL 종료	TLS/SSL 터미네이션 지원	TLS/SSL 터미네이션 지원
상호 TLS (MTL)	Transfer Family는 현재 MTL용 NLB 사용을 지원하지 않습니다.	MTL에 대한 지원

Configure NLB

이 절차는 VPC에 인터넷 연결 네트워크 로드 밸런서 (NLB) 를 설정하는 방법을 설명합니다.

Network Load Balancer를 생성하고 서버의 VPC 엔드포인트를 로드 밸런서의 대상으로 정의하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon Elastic Compute Cloud 콘솔을 엽니다.
2. 탐색 창에서, 로드 밸런서를 선택한 다음, 로드 밸런서 만들기를 선택합니다.
3. Network Load Balancer에서 [생성(Create)]을 선택합니다.
4. 기본 구성 섹션에서 다음 정보를 입력합니다.
   - 이름에 로드 밸런서를 설명하는 이름을 입력합니다.

- 체계에서 인터넷 연결 을 선택합니다.
- IP 주소 타입에서 Ipv4를 선택합니다.

5. 네트워크 맵핑 섹션에서 다음 정보를 입력합니다.

   - VPC의 경우 생성한 Virtual Private Cloud(VPC)를 선택합니다.
   - 매핑아래에서 서버 엔드포인트에서 사용하는 것과 동일한 VPC에서 사용할 수 있는 퍼블릭 서브넷과 연결된 가용 영역을 선택합니다.
   - 각 서브넷의 IPv4 주소에 대해 할당한 탄력적 IP 주소 중 하나를 선택합니다.

6. 리스너 및 라우팅 섹션에서 다음 정보를 입력합니다.

   - 프로토콜에서 TCP를 선택합니다.
   - 포트에서 **5080**를 입력합니다.
   - 기본 작업, 대상 그룹 생성을 선택합니다. 새 대상 그룹을 만드는 세부 사항에 대한 자세한 내용은 대상 그룹 생성을 참조하세요.

   대상 그룹을 생성한 후 기본 작업 필드에 해당 이름을 입력합니다.

7. 보안 리스너 설정 섹션에서 기본 SSL/TLS 인증서 영역에서 인증서를 선택합니다.
8. 로드밸런서 만들기를 선택해서 사용자의 NLB를 생성합니다.
9. (옵션이지만 권장됨)Network Load Balancer의 액세스 로그에 설명된 대로 Network Load Balancer의 액세스 로그를 켜서 전체 감사 추적을 유지하세요.

   NLB에서 TLS 연결이 종료되므로 이 단계를 사용하는 것이 좋습니다. 따라서 Transfer Family AS2 CloudWatch 로그 그룹에 반영되는 소스 IP 주소는 거래 파트너의 외부 IP 주소가 아닌 NLB의 사설 IP 주소입니다.

Configure ALB

이 절차는 VPC에서 애플리케이션 로드 밸런서 (NLB) 를 설정하는 방법을 설명합니다.

Application Load Balancer를 생성하고 서버의 VPC 엔드포인트를 로드 밸런서의 대상으로 정의하려면

1. https://console.aws.amazon.com/ec2/에서 Amazon Elastic Compute Cloud 콘솔을 엽니다.
2. 탐색 창에서, 로드 밸런서를 선택한 다음, 로드 밸런서 만들기를 선택합니다.
3. Application Load Balancer 아래에서 생성(Create)을 선택합니다.

4. ALB 콘솔에서 포트 443 (HTTPS) 에 새 HTTP 리스너를 생성합니다.
5. (선택 사항). 상호 인증 (MTL) 을 설정하려면 보안 설정과 신뢰 저장소를 구성하십시오.
   a. SSL/TLS 인증서를 리스너에 연결합니다.
   b. 클라이언트 인증서 처리에서 상호 인증 (mTLS) 을 선택합니다.
   c. 신뢰 저장소를 통한 확인을 선택합니다.
   d. 고급 mTLS 설정에서 CA 인증서를 업로드하여 신뢰 저장소를 선택하거나 생성합니다.
6. 새 대상 그룹을 생성하고 Transfer Family AS2 서버 엔드포인트의 사설 IP 주소를 포트 5080의 대상으로 추가합니다. 새 대상 그룹을 만드는 세부 사항에 대한 자세한 내용은 대상 그룹 생성을 참조하세요.
7. 포트 5080에서 TCP 프로토콜을 사용하도록 대상 그룹의 상태 점검을 구성하십시오.
8. 리스너의 HTTPS 트래픽을 대상 그룹으로 전달하는 새 규칙을 생성합니다.
9. SSL/TLS 인증서를 사용하도록 리스너를 구성합니다.

로드 밸런서를 설정하고 나면 클라이언트는 사용자 지정 포트 리스너를 통해 로드 밸런서와 통신합니다. 그러면 로드 밸런서가 포트 5080을 통해 서버와 통신합니다.

대상 그룹 생성

1. 이전 절차에서 대상 그룹 만들기를 선택하면 새 대상 그룹에 대한 그룹 세부 정보 지정 페이지로 이동합니다.
2. 기본 구성 섹션에서, 다음 정보를 입력합니다.
   - 대상 타입 선택에서 IP 주소를 선택합니다.
   - 대상 그룹 이름에 대상 그룹의 이름을 입력합니다.
   - 프로토콜에서 TCP를 선택합니다.
   - 포트에서 **5080**를 입력합니다.
   - IP 주소 타입에서 Ipv4를 선택합니다.
   - VPC 에서 Transfer Family AS2 서버용으로 생성한 VPC를 선택합니다.
3. 상태 확인 섹션에서 상태 확인 프로토콜에 대한 TCP를 선택합니다.
4. 다음을 선택합니다.
5. 대상 등록 페이지에서 다음 정보를 입력합니다.
   - 네트워크의 경우, Transfer Family AS2 서버용으로 만든 VPC가 지정되었는지 확인합니다.

- IPv4 주소에 Transfer Family AS2 서버 엔드포인트의 프라이빗 IPv4 주소를 입력합니다.

  서버에 엔드포인트가 두 개 이상 있는 경우, IPv4 주소 추가를 선택하여 다른 IPv4 주소를 입력하는 행을 하나 더 추가합니다. 모든 서버 엔드포인트의 사설 IP 주소를 입력할 때까지 이 프로세스를 반복합니다.
- Ports가 **5080**로 설정되어 있는지 확인하세요.
- 아래 보류 중인 항목으로 포함을 선택하여 항목을 대상 검토 섹션에 추가하세요.

6. 대상 검토 섹션에서 IP 대상을 검토하세요.
7. 대상 그룹 생성을 선택한 다음 이전 NLB 생성 절차로 돌아가 표시된 위치에 새 대상 그룹을 입력합니다.

탄력적 IP 주소에서 서버 액세스 테스트

Network Load Balancer의 엘라스틱 IP 주소 또는 DNS 이름을 사용하여 사용자 지정 포트를 통해 서버에 연결합니다.

> **Important**
>
> 로드 밸런서에 구성된 서브넷의 네트워크 액세스 제어 목록(네트워크 ACL)를 사용하여 클라이언트 IP 주소를 통한 서버 액세스를 관리합니다. 네트워크 ACL 권한은 서브넷 수준에서 설정되므로 해당 서브넷을 사용하는 모든 리소스에 규칙이 적용됩니다. 로드 밸런서의 대상 타입이 인스턴스가 아닌IP 주소 로 설정되어 있기 때문에 보안 그룹을 사용하여 클라이언트 IP 주소에 서의액세스를 제어할 수 없습니다. 따라서 로드 밸런서는 원본 IP 주소를 보존하지 않습니다. Network Load Balancer의 상태 확인이 실패하면 로드 밸런서가 서버 엔드포인트에 연결할 수 없다는 의미입니다. 이 문제를 해결하려면 다음을 확인 하세요.
>
> - 서버 엔드포인트의 관련 보안 그룹이 로드 밸런서에 구성된 서브넷으로부터의 인바운드 연결을 허용하는지 확인하세요. 로드 밸런서는 포트 5080을 통해 서버 엔드포인트에 연결할 수 있어야 합니다.
> - 서버 상태가 온라인인지 확인합니다.

AS2 커넥터를 사용하여 파일 전송

AS2 커넥터는 Transfer Family 서버에서 파트너 소유의 외부 대상으로 AS2 메시지를 전송하기 위해 거래 파트너들 사이의 관계를 설정합니다.

Transfer Family를 사용하면 다음 `start-file-transfer` AWS Command Line Interface ()AWS CLI 명령에 표시된 대로 커넥터 ID 및 파일 경로를 참조하여 AS2 메시지를 보낼 수 있습니다.

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

커넥터의 세부 정보를 가져오려면 다음 명령을 실행합니다.

```
aws transfer list-connectors
```

이 `list-connectors` 명령은 커넥터의 커넥터 ID, URL 및 Amazon 리소스 이름(ARN)을 반환합니다.

특정 커넥터의 속성을 반환하려면 사용하려는 ID로 다음 명령을 실행합니다.

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

이 `describe-connector` 명령은 커넥터의 URL, 역할, 프로필, mDNS (메시지 처리 알림), 태그 및 모니터링 메트릭을 포함하여 커넥터의 모든 속성을 반환합니다.

JSON 및 MDN 파일을 보면 파트너가 파일을 성공적으로 수신했는지 확인할 수 있습니다. 이러한 파일 이름은 파일 이름 및 위치에 설명된 규칙에 따라 지정됩니다. 커넥터를 생성할 때 로깅 역할을 구성한 경우 CloudWatch 로그에서 AS2 메시지 상태를 확인할 수도 있습니다.

AS2 커넥터 세부 정보를 보려면 AS2 커넥터 세부 정보 보기 섹션을 참조하세요. AS2 생성에 대한 자세한 설명은 AS2 커넥터 구성 섹션을 참조하세요.

파일 이름 및 위치

이 섹션에서는 AS2 전송에 대한 파일 이름 지정 규칙에 대해 설명합니다.

인바운드 파일 전송에 대해 다음 사항을 참조하세요.

- 계약서에 기본 디렉터리를 지정합니다. 기본 디렉터리는 접두사(있는 경우)와 결합된 Amazon S3 버킷 이름입니다. 예를 들어 `/DOC-EXAMPLE-BUCKET/AS2-folder`입니다.
- 수신 파일이 성공적으로 처리되면 파일(및 해당 JSON 파일)이 `/processed` 폴더에 저장됩니다. 예를 들어 `/DOC-EXAMPLE-BUCKET/AS2-folder/processed`입니다.

  JSON 파일에는 다음과 같은 필드가 포함되어 있습니다.

  - 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
- 들어오는 파일을 제대로 처리할 수 없는 경우 파일(및 해당 JSON 파일)이 /failed 폴더에 저장됩니다. 예를 들어 /DOC-EXAMPLE-BUCKET/AS2-folder/failed입니다.
- 전송된 파일은 processed 폴더에 *original_filename*.*messageId*.*original_extension*로 저장됩니다. 즉, 전송 메시지 ID가 파일 이름에 원래 확장명 앞에 추가됩니다.
- JSON 파일이 생성되고 *original_filename*.*messageId*.*original_extension*.json로 저장됩니다. 메시지 ID가 추가될 뿐 아니라 전송된 파일 이름에도 문자열 .json이 추가됩니다.
- 메시지 처리 알림(MDN) 파일이 생성되어 *original_filename*.*messageId*.*original_extension*.mdn로 저장됩니다. 메시지 ID가 추가될 뿐 아니라 전송된 파일 이름에도 문자열 .mdn이 추가됩니다.
- ExampleFileInS3Payload.dat라는 인바운드 파일이 있는 경우 다음 파일이 생성됩니다.
  - File – ExampleFileInS3Payload.c4d6b6c7-23ea-4b8c-9ada-0cb811dc8b35@44313c54b0a46a36.

- JSON – ExampleFileInS3Payload.c4d6b6c7-23ea-4b8c-9ada-0cb811dc8b35@44313c54b0a46a36.
- MDN – ExampleFileInS3Payload.c4d6b6c7-23ea-4b8c-9ada-0cb811dc8b35@44313c54b0a46a36.

아웃바운드 전송의 경우 받는 메시지 파일이 없다는 점을 제외하면 이름이 비슷하며 전송된 메시지의 전송 ID가 파일 이름에 추가된다는 차이점이 있습니다. 전송 ID는 StartFileTransfer API 작업(또는 다른 프로세스나 스크립트가 이 작업을 호출할 때)에 의해 반환됩니다.

- transfer-id는 파일 전송과 관련된 식별자입니다. StartFileTransfer 호출에 포함된 모든 요청은 transfer-id를 공유합니다.
- 기본 디렉터리는 소스 파일에 사용하는 경로와 동일합니다. 즉, 기본 디렉터리는 StartFileTransfer API 작업 또는 start-file-transfer AWS CLI 명령에서 지정하는 경로입니다. 예:

  ```
  aws transfer start-file-transfer --send-file-paths /DOC-EXAMPLE-BUCKET/AS2-folder/
  file-to-send.txt
  ```

  이 명령을 실행하면 MDN 및 JSON 파일이 /DOC-EXAMPLE-BUCKET/AS2-folder/processed(전송 성공 시) 또는 /DOC-EXAMPLE-BUCKET/AS2-folder/failed(전송 실패 시)에 저장됩니다.
- JSON 파일이 생성되고 *original_filename*.*transferId*.*messageId*.*original_extension*.json로 저장됩니다.
- MDN 파일이 생성되고 *original_filename*.*transferId*.*messageId*.*original_extension*.mdn로 저장됩니다.
- ExampleFileOutTestOutboundSyncMdn.dat라는 아웃바운드 파일이 있는 경우 다음 파일이 생성됩니다:
  - 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

또한 CloudWatch 로그를 확인하여 실패한 전송을 포함한 전송의 세부 정보를 볼 수 있습니다.

상태 코드

다음 표에는 귀하 또는 파트너가 AS2 메시지를 보낼 때 CloudWatch 로그에 기록할 수 있는 모든 상태 코드가 나와 있습니다. 메시지 유형마다 다른 메시지 처리 단계가 적용되며 모니터링용으로만 사용됩니다. 완료 및 실패 상태는 처리의 마지막 단계를 나타내며 JSON 파일에서 확인할 수 있습니다.

코드	설명	처리가 완료되었나요?
가공	메시지가 최종 형식으로 변환되는 중입니다. 예를 들어, 압축 해제 단계와 암호 해독 단계 모두 이 상태입니다.	아니요
MDN_TRANSBIT	메시지 프로세싱이 MDN 응답을 보내는 중입니다.	아니요
MDN_RECEIVE	메시지 프로세싱에서 MDN 응답을 받고 있습니다.	아니요
완료됨	메시지 처리가 성공적으로 완료되었습니다. 이 상태에는 인바운드 메시지 또는 아웃바운드 메시지의 MDN 확인을 위해 MDN을 보내는 경우가 포함됩니다.	예
FAILED	메시지 처리에 실패했습니다. 오류 코드 목록은 을 참조하십시오AS2 오류 코드.	예

샘플 JSON 파일

이 섹션에는 성공적인 전송과 실패한 전송을 위한 샘플 파일을 포함하여 인바운드 및 아웃바운드 전송에 대한 샘플 JSON 파일이 나열되어 있습니다.

성공적으로 전송된 샘플 아웃바운드 파일:

```
{
```

```
  "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"
}
```

전송에 실패한 샘플 아웃바운드 파일:

```
{
  "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"
}
```

성공적으로 전송된 샘플 인바운드 파일:

```
{
```

```
  "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"
}
```


전송에 실패한 샘플 인바운드 파일:


```
{
  "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"
}
```

AS2 사용량 모니터링

Amazon CloudWatch 및 AWS CloudTrail 을 사용하여 AS2 활동을 모니터링할 수 있습니다. 다른 Transfer Family 서버 지표(metrics)를 보려면 아마존 CloudWatch 로깅 대상 AWS Transfer Family을 참조하십시오.

AS2 지표

지표	설명
`InboundMessage`	거래 파트너로부터 성공적으로 수신한 AS2 메시지의 총 개수. 단위: 개 기간: 5분
`InboundFailedMessage`	거래 파트너로부터 성공적으로 수신되지 못한 AS2 메시지의 총 개수. 즉, 거래 파트너가 메시지를 보냈지만 Transfer Family 서버가 메시지를 성공적으로 처리하지 못했습니다. 단위: 개 기간: 5분
`OutboundMessage`	Transfer Family 서버에서 거래 파트너로 성공적으로 보낸 AS2 메시지의 총 개수입니다. 단위: 개 기간 = 5분
`OutboundFailedMessage`	거래 파트너에게 성공적으로 전송되지 못한 AS2 메시지의 총 개수. 즉, Transfer Family 서버에서 전송되었지만 거래 파트너가 성공적으로 수신하지 못했습니다. 단위: 개 기간: 5분

AS2 상태 코드

다음 표에는 귀하 또는 파트너가 AS2 메시지를 보낼 때 CloudWatch 로그에 기록할 수 있는 모든 상태 코드가 나와 있습니다. 메시지 유형마다 다른 메시지 처리 단계가 적용되며 모니터링용으로만 사용됩니다. 완료 및 실패 상태는 처리의 마지막 단계를 나타내며 JSON 파일에서 확인할 수 있습니다.

코드	설명	처리가 완료되었나요?
가공	메시지가 최종 형식으로 변환되는 중입니다. 예를 들어, 압축 해제 단계와 암호 해독 단계 모두 이 상태입니다.	아니요
MDN_TRANSBIT	메시지 프로세싱이 MDN 응답을 보내는 중입니다.	아니요
MDN_RECEIVE	메시지 프로세싱에서 MDN 응답을 받고 있습니다.	아니요
완료됨	메시지 처리가 성공적으로 완료되었습니다. 이 상태에는 인바운드 메시지 또는 아웃바운드 메시지의 MDN 확인을 위해 MDN을 보내는 경우가 포함됩니다.	예
FAILED	메시지 처리에 실패했습니다. 오류 코드 목록은 을 참조하십시오AS2 오류 코드.	예

AS2 오류 코드

다음 표는 AS2 파일 전송에서 발생할 수 있는 오류 코드를 나열하고 설명합니다.

AS2 오류 코드

코드	오류	설명 및 해상도
ACCESS_DENIED	• 액세스가 거부되었습니다. 액세스 역할에 필요한 권한이 있는지 확인하세요. • 잘못된 파일 경로 *send-file-path* • ErrorCode다음과 같은 *##-###* 자격 증명을 가져오지 못했습니다.	유효하지 않거나 형식이 StartFileTransfer 잘못된 요청을 처리할 때 발생합니다SendFilePaths . 즉, 경로에 Amazon S3 버킷 이름이 없거나 경로에 유효하지 않은 문자가 포함되어 있습니다. Transfer Family가 액세스 역할 또는 로깅 역할을 맡지 못하는 경우에도 발생합니다. 경로에 유효한 Amazon S3 버킷과 키 이름이 포함되어 있는지 확인하십시오.
AGREEMENT_NOT_FOUND	계약을 찾을 수 없습니다.	계약을 찾을 수 없거나 계약이 비활성 프로파일과 연결되어 있습니다. Transfer Family 서버 내에서 계약을 업데이트하여 활성 프로파일을 포함하십시오.
CONNECTOR_NOT_FOUND	커넥터 또는 관련 구성을 찾을 수 없습니다.	계약을 찾을 수 없거나 계약이 비활성 프로파일과 연결되어 있습니다. 활성 프로파일을 포함하도록 커넥터를 업데이트하십시오.
CREDENTIALS_RETRIEVAL_FAILED	1. Secrets Manager에서 보안 암호를 찾을 수 없습니다. 2. Secrets Manager에 액세스할 수 없습니다	AS2 Basic 인증의 경우 보안 암호를 올바르게 형식화해야 합니다. 다음 해결 방법은 이전 열에 나열된 오류에 해당합니다.

코드	오류	설명 및 해상도
	3. Secrets Manager에서 보안 암호 해독 실패 4. 제한으로 인해 보안 암호 값을 가져올 수 없습니다.	1. 보안 암호 ID가 정확한지 확인하십시오. 2. 액세스 역할에 암호를 읽을 수 있는 적절한 권한이 있는지 확인하십시오. 접근 역할은 `StartFileTransfer` 요청에서 사용된 파일 위치의 상위 디렉터리에 대한 읽기 및 쓰기 액세스를 제공해야 합니다. 또한 그 역할이 `StartFileTransfer` 와 함께 전송하려는 파일의 상위 디렉터리에 대한 읽기 및 쓰기 액세스 권한을 제공하는 지 확인하십시오. 3. 고객 관리 키를 보안 암호로 사용하는 경우 액세스 역할에 AWS Key Management Service (AWS KMS) 키에 대한 권한이 있는지 확인하십시오. 4. 해당 할당량은 비밀 처리를 위한 할당량을 참조하십시오.
`DECOMPRESSION_FAILED`	메시지 압축을 풀지 못했습니다.	전송된 파일이 손상되었거나 압축 알고리즘이 유효하지 않습니다. 메시지를 다시 전송하고 ZLIB 압축이 사용되는지 확인하거나 압축을 사용하지 않고 메시지를 다시 보내십시오.

코드	오류	설명 및 해상도
DECRYPT_FAILED	*### ID* 메시지를 해독하지 못했습니다. 파트너가 올바른 공개 암호화 키를 가지고 있는지 확인하십시오.	복호화에 실패했습니다. 파트너가 유효한 인증서를 사용하여 페이로드를 전송했고 암호화는 유효한 암호화 알고리즘을 사용하여 수행되었는지 확인합니다.
DECRYPT_FAILED_INVALID_SMIME_FORMAT	엔벨로핑된 MimePart를 파싱할 수 없습니다.	MIME 페이로드가 손상되었거나 지원되지 않는 SMIME 형식입니다. 발신자는 사용 중인 형식이 지원되는지 확인한 다음 페이로드를 다시 전송해야 합니다.
DECRYPT_FAILED_NO_DECRYPTION_KEY_FOUND	일치하는 암호 해독 키를 찾을 수 없습니다.	파트너 프로필에 메시지와 일치하는 인증서가 할당되지 않았거나 메시지와 일치하는 인증서가 현재 만료되었거나 더 이상 유효하지 않습니다. 파트너 프로필을 업데이트하고 유효한 인증서가 포함되어 있는지 확인해야 합니다.
DECRYPT_FAILED_UNSUPPORTED_ENCRYPTION_ALG	지원되지 않는 알고리즘(ID: *encryption-ID*)을 사용하여 SMIME 페이로드 암호 해독을 요청했습니다.	원격 발신자가 지원되지 않는 암호화 알고리즘이 포함된 AS2 페이로드를 전송했습니다. 발신자는 AWS Transfer Family에서 지원하는 암호화 알고리즘을 선택해야 합니다.

코드	오류	설명 및 해상도
DUPLICATE_MESSAGE	중복 또는 이중 처리된 단계.	페이로드에 중복 처리 단계가 있습니다. 예를 들어 2개의 암호화 단계가 있습니다. 서명, 압축, 암호화를 한 번에 완료하여 메시지를 재전송합니다.
ENCRYPT_FAILED_NO_ENCRYPTION_KEY_FOUND	프로필에서 유효한 공개 암호화 인증서를 찾을 수 없음: *local-profile-ID*	Transfer Family가 아웃바운드 메시지를 암호화하려고 시도하지만 로컬 프로필에 대한 암호화 인증서를 찾을 수 없습니다. 해결 옵션: • 로컬 프로필에 암호화를 위한 인증서와 개인 키가 첨부되어 있는지 확인하십시오. • 암호화 인증서가 현재 활성 상태인지 확인하십시오.
ENCRYPTION_FAILED	*## ##* 파일을 암호화하지 못했습니다.	전송할 파일은 암호화에 사용할 수 없습니다. 파일이 예상 AS2 위치에 있고 AWS Transfer Family가 파일을 읽을 권한이 있는지 확인하십시오.
FILE_SIZE_TOO_LARGE	파일 크기가 너무 큽니다.	이는 파일 크기 제한을 초과하는 파일을 보내거나 받을 때 발생합니다.

코드	오류	설명 및 해상도
HTTP_ERROR_RESPONSE_FROM_PARTNER	*### URL* ID=*message-ID* 인 메시지에 대해 상태 400을 반환했습니다.	파트너의 AS2 서버와 통신할 때 예상치 못한 HTTP 응답 코드가 반환되었습니다. 파트너는 AS2 서버 로그에서 추가 진단을 제공할 수 있습니다.
INSUFFICENT_MESSAGE_SECURITY_UNENCRYPTED	암호화가 필요합니다.	파트너가 Transfer Family에 암호화되지 않은 메시지를 보냈지만 이 메시지는 지원되지 않습니다. 발신자는 암호화된 페이로드를 사용해야 합니다.
INVALID_ENDPOINT_PROTOCOL	HTTP와 HTTPS만 지원됩니다.	AS2 커넥터 구성에서 HTTP 또는 HTTPS를 프로토콜로 지정해야 합니다.

코드	오류	설명 및 해상도
INVALID_REQUEST	1. 메시지 헤더에 문제가 있습니다. 2. 보안 JSON을 구문 분석할 수 없습니다. 시크릿 JSON이 예상 형식과 일치하지 않았습니다. 3. 보안 암호는 JSON 문자열이어야 합니다. 4. 사용자 이름은 콜론을 포함할 수 없습니다. 사용자 이름은 제어 문자를 포함할 수 없습니다. 사용자 이름은 ASCII 문자만 포함해야 합니다. 비밀번호는 제어 문자를 포함할 수 없습니다. 비밀번호는 ASCII 문자만 포함해야 합니다.	이 오류에는 여러 가지 원인이 있습니다. 다음 해결 방법은 이전 열에 나열된 오류에 해당합니다. 1. as2-from 및 as2-to 필드를 확인합니다. 원본 메시지 ID가 MDN 형식에 맞는지 확인하십시오. 또한 메시지 ID 형식에 AS2 헤더가 누락되지 않았는지 확인하십시오. 2. AS2 커넥터에 대한 기본 인증을 활성화합니다.에 설명된 대로 암호 값이 문서화된 형식과 일치하는지 확인하십시오. 3. 암호는 바이너리가 아닌 문자열로 제공되어야 합니다. 4. 사용자 이름 또는 비밀번호를 필요에 따라 수정하십시오.
INVALID_URL_FORMAT	잘못된 URL 형식: *URL*	잘못된 URL로 구성된 커넥터를 사용하여 아웃바운드 메시지를 보낼 때 이 문제가 발생합니다. 커넥터가 유효한 HTTP 또는 HTTPS URL로 구성되어 있는지 확인하십시오.

코드	오류	설명 및 해상도
MDN_RESPONSE_INDICATES_AUTHENTICATION_FAILED	해당 사항 없음	수신자는 발신자를 인증할 수 없습니다. 거래 파트너가 처리 수정자 오류: authentication-failed와 함께 Transfer Family에 MDN을 반환합니다.
MDN_RESPONSE_INDICATES_DECOMPRESSION_FAILED	해당 사항 없음	이는 수신자가 메시지 내용을 압축 해제할 수 없을 때 발생합니다. 거래 파트너가 처리 수정자 오류: decompression-failed와 함께 Transfer Family에 MDN을 반환합니다.
MDN_RESPONSE_INDICATES_DECRYPTION_FAILED	해당 사항 없음	수신자는 메시지 내용을 해독할 수 없습니다. 거래 파트너가 처리 수정자 오류: authentication-failed와 함께 Transfer Family에 MDN을 반환합니다.
MDN_RESPONSE_INDICATES_INSUFFICIENT_MESSAGE_SECURITY	해당 사항 없음	수신자는 메시지가 서명되거나 암호화되기를 기대하지만, 그렇지 않습니다. 거래 파트너가 처분 수정자 Error: 와 함께 Transfer Family에 MDN을 반환합니다. insufficient-message-security 커넥터에서 서명 및/또는 암호화를 활성화하여 거래 파트너의 기대에 부합하도록 하십시오.

코드	오류	설명 및 해상도
MDN_RESPONSE_INDICATES_INTEGRITY_CHECK_FAILED	해당 사항 없음	수신자는 콘텐츠 무결성을 확인할 수 없습니다. 거래 파트너가 처분 수정자 Error: 와 함께 Transfer Family에 MDN을 반환합니다. integrity-check-failed
PATH_NOT_FOUND	디렉터리 *file-path* 를 생성할 수 없습니다. 상위 경로를 찾을 수 없습니다.	Transfer Family가 고객의 Amazon S3 버킷에 디렉터리를 생성하려고 시도했지만 버킷을 찾을 수 없습니다. StartFileTransfer 명령에 언급된 각 경로에 기존 버킷의 이름이 포함되어 있는지 확인하십시오.
SEND_FILE_NOT_FOUND	파일 경로 *## ##*를 찾을 수 없습니다.	Transfer Family는 파일 보내기 작업에서 파일을 찾을 수 없습니다. 구성된 홈 디렉토리 및 경로가 유효하고 Transfer Family에 파일에 대한 읽기 권한이 있는지 확인합니다.
SERVER_NOT_FOUND	이벤트와 연결된 서버를 찾을 수 없습니다.	Transfer Family가 메시지를 받았을 때 서버를 찾지 못했습니다. 수신 메시지를 처리하는 중에 서버가 삭제되면 이런 일이 발생할 수 있습니다.

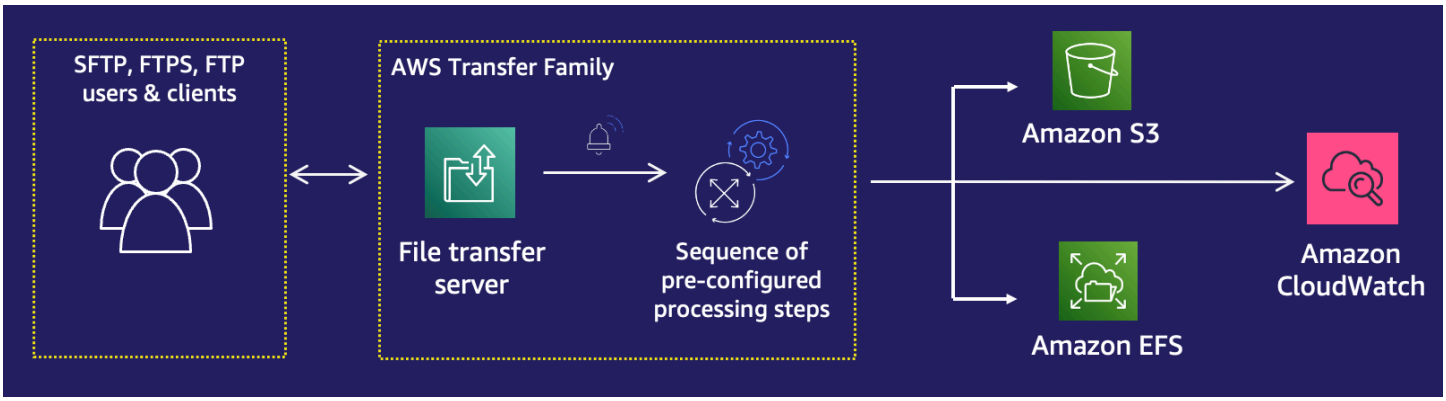
코드	오류	설명 및 해상도
SERVER_NOT_ONLINE	서버 *server-ID* 가 온라인 상태가 아닙니다.	Transfer Family 서버가 오프라인 상태입니다. 메시지를 수신하고 처리할 수 있도록 서버를 시작합니다.
SIGNING_FAILED	파일에 서명하지 못했습니다.	전송할 파일을 서명할 수 없거나 서명을 수행할 수 없습니다. 파일이 예상 AS2 위치에 있고 AWS Transfer Family가 파일을 읽을 권한이 있는지 확인하십시오.
SIGNING_FAILED_NO_SIGNING_KEY_FOUND	프로필: *##* 프로필 ID에 대한 인증서를 찾을 수 없습니다.	아웃바운드 메시지에 서명하려고 시도했지만 로컬 프로필에 대한 서명 인증서를 찾을 수 없습니다. 해결 옵션: • 로컬 프로필에 서명을 위한 인증서와 개인 키가 첨부되어 있는지 확인하십시오. • 서명 인증서가 현재 활성 상태인지 확인하십시오.
UNABLE_RESOLVE_HOST_TO_IP_ADDRESS	호스트 이름을 IP 주소로 확인할 수 없습니다.	Transfer Family는 AS2 커넥터에 구성된 공용 DNS 서버에서 DNS 대 IP 주소 확인을 수행할 수 없습니다. 커넥터가 유효한 파트너 URL을 가리키도록 업데이트하십시오.

코드	오류	설명 및 해상도
UNABLE_TO_CONNECT_TO_REMOTE_HOST_OR_IP	엔드포인트에 대한 연결이 시간 초과되었습니다.	Transfer Family는 구성된 파트너의 AS2 서버에 소켓 연결을 설정할 수 없습니다. 구성된 IP 주소에서 파트너의 AS2 서버를 사용할 수 있는지 확인하십시오.
UNABLE_TO_RESOLVE_HOSTNAME	*### ##*의 호스트 이름을 확인할 수 없습니다.	Transfer Family 서버가 공용 DNS 서버를 사용하여 파트너의 호스트 이름을 확인할 수 없습니다. 구성된 호스트가 등록되어 있고 DNS 레코드를 게시할 시간이 있었는지 확인하십시오.
VERIFICATION_FAILED	*### ID* AS2 메시지에 대한 서명 확인에 실패했거나 MIC 코드가 일치하지 않았습니다.	발신자의 서명 인증서가 원격 프로파일의 서명 인증서와 일치하는지 확인하십시오. 또한 MIC 알고리즘이 AWS Transfer Family과 호환되는지도 확인하십시오.

코드	오류	설명 및 해상도
VERIFICATION_FAILED_NO_MATCHING_KEY_FOUND	• 프로필: *### ### ID*메시지 서명과 일치하는 공개 인증서를 찾을 수 없습니다. • 존재하지 않는 프로필 (*partner-profile-ID*)에 대한 인증서를 얻을 수 없습니다. • 프로필(*partner-profile-ID*)에서 유효한 인증서를 찾을 수 없습니다.	AWS Transfer Family이 받은 메시지의 서명을 확인하려고 하는데 파트너 프로필과 일치하는 서명 인증서를 찾을 수 없습니다. 해결 옵션: • 파트너 프로필에 서명을 위한 인증서가 첨부되어 있는지 확인하십시오. • 인증서가 현재 활성 상태인지 확인하십시오. • 인증서가 파트너를 위한 정확한 서명 인증서인지 확인하십시오.

AWS Transfer Family 관리형 워크플로

AWS Transfer Family 파일 처리를 위한 관리형 워크플로를 지원합니다. 관리형 워크플로를 사용하면 SFTP, FTPS 또는 FTP를 통해 파일이 전송된 후 워크플로를 시작할 수 있습니다. 이 기능을 사용하면 파일 처리에 필요한 모든 단계를 조정하여 business-to-business (B2B) 파일 교환에 대한 규정 준수 요구 사항을 안전하고 비용 효율적으로 충족할 수 있습니다. 또한 end-to-end 감사 및 가시성의 이점을 누릴 수 있습니다.



관리형 워크플로는 파일 처리 작업을 오케스트레이션하여 다운스트림 애플리케이션에서 데이터를 사용하기 전에 데이터를 사전 처리하는 데 도움이 됩니다. 이러한 파일 처리 작업에는 다음이 포함될 수 있습니다.

- 파일을 사용자별 폴더로 이동.
- 워크플로의 일부로 파일 복호화.
- 파일 태그 지정.
- AWS Lambda 함수를 만들고 워크플로에 연결하여 사용자 지정 처리를 수행합니다.
- 파일이 성공적으로 전송되면 알림을 보냅니다. (이 사용 사례를 자세히 설명하는 블로그 게시물의 경우 AWS Transfer Family 관리형 워크플로를 사용한 파일 전송 알림 사용자 지정을 참조하십시오.)

조직 내 여러 사업부에 걸쳐 있는 일반적인 업로드 후 파일 처리 작업을 빠르게 복제하고 표준화하려면 코드형 인프라(IaC)를 사용하여 워크플로를 배포할 수 있습니다. 전체 업로드된 파일에 대해 관리형 워크플로를 시작하도록 지정할 수 있습니다. 또한 세션이 조기에 끊어져 일부만 업로드된 파일에 대해 다른 관리형 워크플로를 시작하도록 지정할 수 있습니다. 기본 제공되는 예외 처리 기능을 통해 파일 처리 결과에 신속하게 대응하는 동시에 오류 처리 방법을 제어할 수 있습니다. 또한 각 워크플로 단계는 세부 로그를 생성하며, 이를 감사하여 데이터 계보를 추적할 수 있습니다.

시작하려면 다음 단계를 수행합니다.

1. 요구 사항에 따라 복사, 태그 지정 및 기타 단계와 같은 사전 처리 작업을 포함하도록 워크플로를 설정합니다. 세부 정보는 워크플로 만들기를 참조하세요.
2. Transfer Family가 워크플로를 실행하는 데 사용하는 실행 역할을 구성합니다. 세부 정보는 워크플로에 대한 IAM 정책를 참조하세요.
3. 워크플로를 서버에 매핑하여 파일 도착 시 이 워크플로에 지정된 작업이 실시간으로 평가 및 시작되도록 합니다. 세부 정보는 워크플로 구성 및 실행를 참조하세요.

관련 정보

- 워크플로 실행을 모니터링하려면 Transfer Family에 대한 CloudWatch 측정항목 사용를 참조하세요.
- 자세한 실행 로그 및 문제 해결 정보는 Amazon을 사용하여 워크플로 관련 오류 문제 해결 CloudWatch를 참조하세요.
- Transfer Family는 파일 전송 솔루션 구축 과정을 안내하는 블로그 게시물과 워크샵을 제공합니다. 이 솔루션은 관리형 SFTP/FTPS 엔드포인트와 Amazon Cognito 및 DynamoDB를 활용하여 AWS Transfer Family 사용자 관리를 수행합니다.

  블로그 게시물은 Amazon Cognito를 Amazon AWS Transfer Family S3와 함께 자격 증명 공급자로 사용하기 페이지에서 확인할 수 있습니다. 워크숍에 대한 세부 정보는 여기에서 확인할 수 있습니다.
- Transfer Family 워크플로에 대한 간략한 소개는 AWS Transfer Family 관리형 워크플로를 보세요.

주제

- 워크플로 만들기
- 사전 정의된 단계 사용
- 사용자 지정 파일 처리 단계 사용
- 워크플로에 대한 IAM 정책
- 워크플로의 예외 처리
- 워크플로 실행 모니터링
- 템플릿에서 워크플로 생성
- Transfer Family 서버에서 워크플로 제거
- 관리형 워크플로 제한 및 제약 조건

관리형 워크플로를 시작하는 데 도움이 더 필요하면 다음 리소스를 참조하세요.

- AWS Transfer Family 관리형 워크플로우 데모 비디오

- AWS Transfer Family 워크플로를 사용하여 클라우드 네이티브 파일 전송 플랫폼 구축 블로그 게시물

워크플로 만들기

이 항목에 설명된 대로 를 사용하여 관리형 워크플로를 만들 수 있습니다. AWS Management Console 워크플로를 가능한 한 쉽게 만들기 위해 콘솔의 대부분의 섹션에서 상황별 도움말 패널을 사용할 수 있습니다.

워크플로에는 다음과 같은 두 가지 단계가 있습니다.

- 공칭 단계 - 공칭 단계는 들어오는 파일에 적용할 파일 처리 단계입니다. 공칭 단계를 두 개 이상 선택한 경우 각 단계는 선형 시퀀스로 처리됩니다.
- 예외 처리 단계 - 예외 처리기는 정상적인 단계가 실패하거나 검증 오류가 발생하는 경우에 AWS Transfer Family 실행되는 파일 처리 단계입니다.

워크플로 만들기

1. AWS Transfer Family https://console.aws.amazon.com/transfer/ 에서 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 워크플로를 선택합니다.
3. 워크플로 페이지에서 워크플로 생성을 선택합니다.
4. 워크플로 생성 페이지에서 설명을 입력합니다. 이 설명은 워크플로 페이지에 표시됩니다.
5. 공칭 단계 섹션에서 단계 추가를 선택합니다. 단계를 하나 이상 추가합니다.
    a. 사용 가능한 옵션에서 단계 타입을 선택합니다. 다양한 단계 타입에 대한 자세한 내용은 the section called “사전 정의된 단계 사용”를 참조하세요.
    b. 다음을 선택한 다음 해당 단계에 대한 파라미터를 구성합니다.
    c. 다음을 선택한 다음 해당 단계에 대한 세부 정보를 검토합니다.
    d. 단계 생성을 선택하여 단계를 추가하고 계속합니다.
    e. 필요에 따라 단계를 계속 추가합니다. 워크플로의 최대 단계 수는 8개입니다.
    f. 필요한 공칭 단계를 모두 추가한 후 예외 처리기 — 옵션 섹션으로 아래로 스크롤하여 단계 추가를 선택합니다.

> **Note**
>
> 실패 알림을 실시간으로 받을 수 있도록 워크플로가 실패할 때 실행할 예외 처리기와 단계를 설정하는 것이 좋습니다.

6. 예외 처리기를 구성하려면 앞에서 설명한 것과 같은 방식으로 단계를 추가하세요. 파일로 인해 특정 단계에서 예외가 발생하는 경우 예외 처리기가 하나씩 간접적으로 호출됩니다.
7. (옵션) 태그 섹션으로 스크롤하여 워크플로에 사용할 태그를 추가합니다.
8. 플릿 구성을 살펴본 후 워크플로 생성을 선택합니다.

> **Important**
>
> 워크플로를 만든 후에는 편집할 수 없으므로 구성을 주의 깊게 검토해야 합니다.

워크플로 구성 및 실행

워크플로를 실행하려면 먼저 Transfer Family 서버에 연결해야 합니다.

업로드된 파일에서 워크플로를 실행하도록 Transfer Family를 구성하려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 서버를 선택합니다.
   - 기존 서버에 워크플로를 추가하려면 워크플로에 사용할 서버를 선택합니다.
   - 또는 새 서버를 만들고 이 서버에 워크플로를 추가할 수도 있습니다. 자세한 정보는 SFTP, FTPS 또는 FTP 서버 엔드포인트 구성을 참조하세요.
3. 서버의 세부 정보 페이지에서 추가 세부 정보 섹션까지 아래로 스크롤한 다음 편집을 선택합니다.

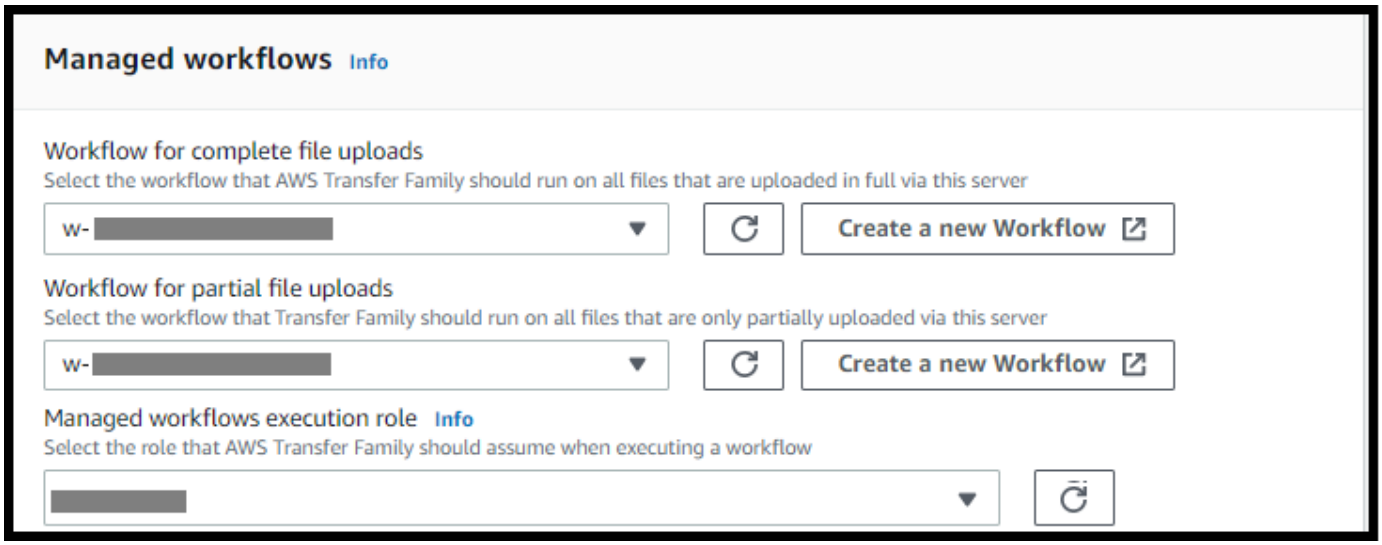
> **Note**
>
> 기본적으로 서버에는 연결된 워크플로가 없습니다. 추가 세부 정보 섹션을 사용하여 워크플로를 선택한 서버에 연결할 수 있습니다.

4. 추가 세부 정보 편집 페이지의 관리형 워크플로 섹션에서 모든 업로드에서 실행할 워크플로를 선택합니다.

> **Note**
>
> 워크플로가 아직 없는 경우 새 워크플로 생성을 선택하여 워크플로를 생성합니다.

a. 사용할 워크플로 ID를 선택합니다.

b. 실행 역할을 선택합니다. 이 역할은 워크플로의 단계를 실행할 때 Transfer Family가 맡는 역할입니다. 자세한 내용은 워크플로에 대한 IAM 정책를 참조하세요. 저장(Save)을 선택합니다.



> **Note**
>
> 워크플로를 더 이상 서버와 연결하지 않으려면 연결을 제거하면 됩니다. 자세한 내용은 Transfer Family 서버에서 워크플로 제거를 참조하세요.

워크플로를 실행하려면

워크플로를 실행하려면 관련 워크플로로 구성한 Transfer Family 서버에 파일을 업로드합니다.

> **Note**
>
> 서버에서 워크플로를 제거하고 새 워크플로로 바꾸거나 워크플로의 실행 역할에 영향을 미치는 서버 구성을 업데이트할 때마다 새 워크플로를 실행하기 전에 약 10분을 기다려야 합니다. Transfer Family 서버는 워크플로 세부 정보를 캐시하며 서버가 캐시를 새로 고치는 데 10분이 걸립니다.
> 또한 활성 SFTP 세션에서 로그아웃한 다음 10분 대기 시간이 지난 후 다시 로그인해야 변경 사항을 확인할 수 있습니다.

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
>
```

파일이 업로드되고 나면 파일에 정의된 작업이 수행됩니다. 예를 들어 워크플로에 복사 단계가 포함된 경우 해당 단계에서 정의한 위치에 파일이 복사됩니다. Amazon CloudWatch Logs를 사용하여 실행된 단계와 실행 상태를 추적할 수 있습니다.

워크플로 세부 정보 보기

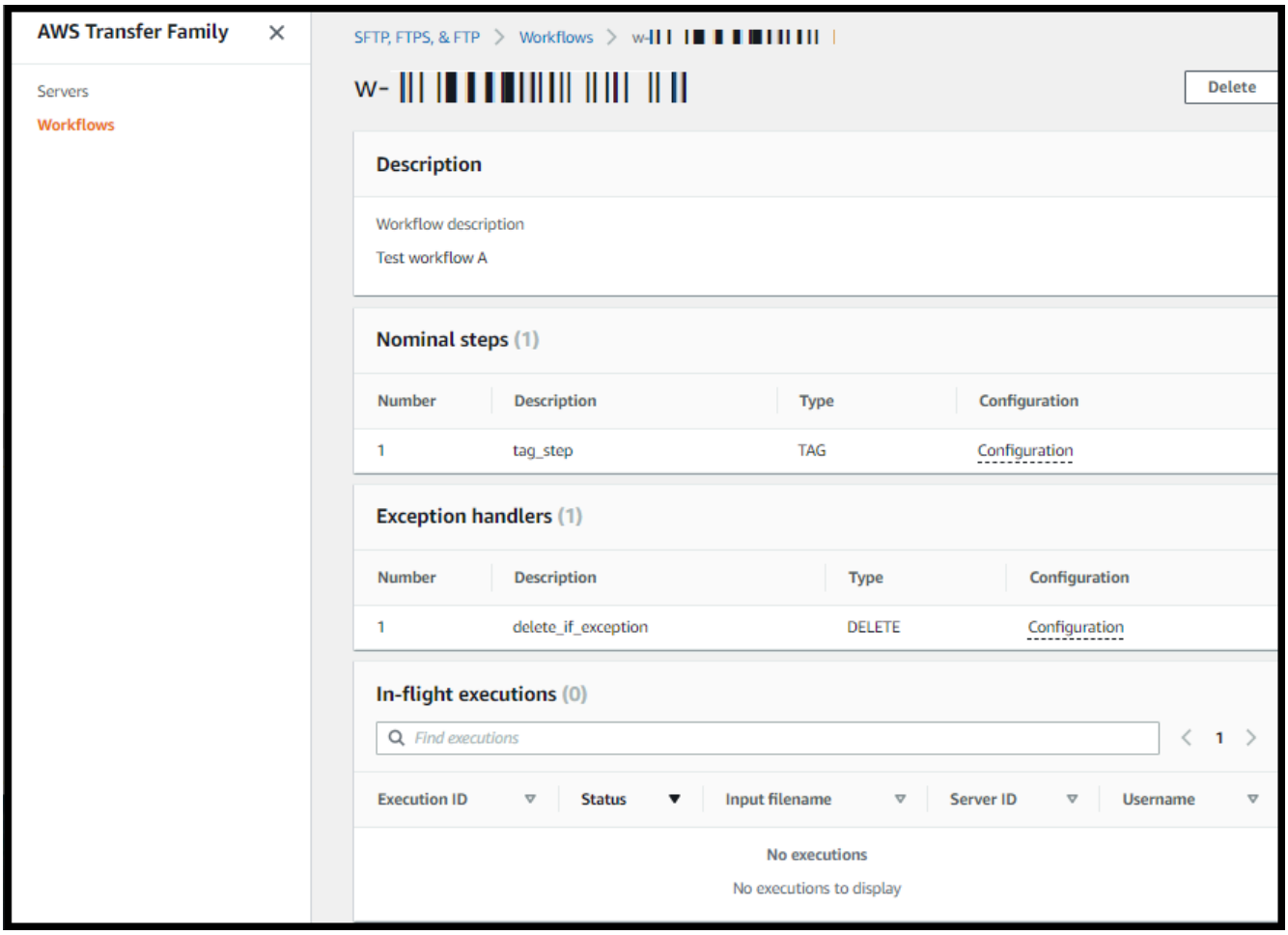
이전에 만든 워크플로 또는 워크플로 실행에 대한 세부 정보를 볼 수 있습니다. 콘솔이나 AWS Command Line Interface (AWS CLI) 를 사용하여 이러한 세부 정보를 볼 수 있습니다.

Console

워크플로 세부 정보 보기

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 워크플로를 선택합니다.
3. 워크플로 페이지에서 워크플로를 선택합니다.

워크플로 세부 정보 페이지가 열립니다.



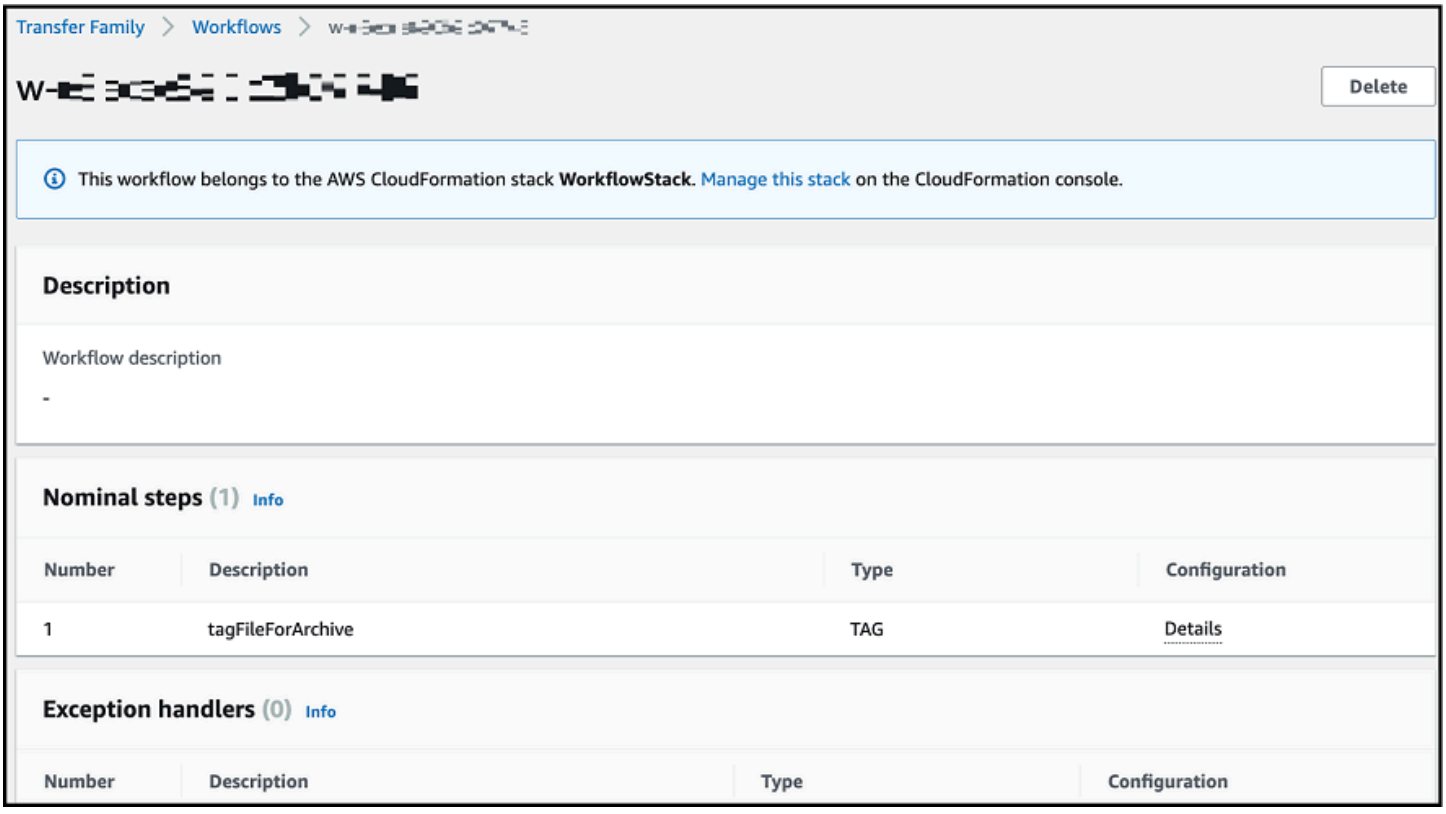
CLI

워크플로 세부 정보를 보려면 다음 예에서와 같이 `describe-workflow` CLI 명령을 사용합니다. 워크플로 ID *w-1234567890abcdef0*를 사용자의 고유한 값으로 바꿉니다. 자세한 내용은 AWS CLI 명령 참조에서 describe-workflow를 참조하세요.

```
# 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": {}
    }
}
```

워크플로가 AWS CloudFormation 스택의 일부로 생성된 경우 AWS CloudFormation 콘솔 (https://console.aws.amazon.com/cloudformation) 을 사용하여 워크플로를 관리할 수 있습니다.



사전 정의된 단계 사용

워크플로를 만들 때 이 항목에 설명된 다음과 같은 사전 정의된 단계 중 하나를 추가하도록 선택할 수 있습니다. 또한 사용자 지정 파일 처리 단계 추가를 선택할 수도 있습니다. 자세한 내용은 the section called “사용자 지정 파일 처리 단계 사용”를 참조하세요.

주제

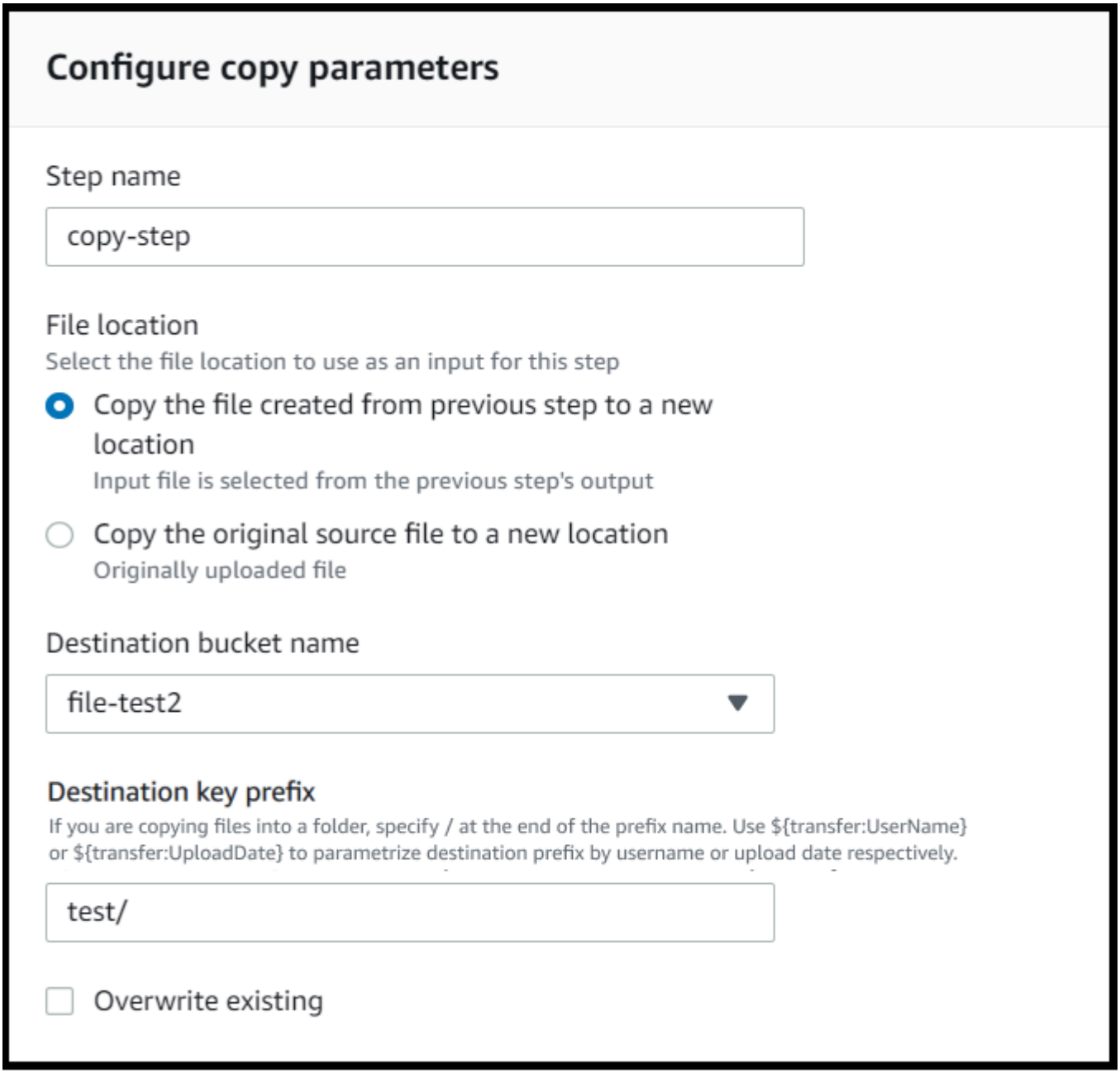
- 파일 복사
- 파일 복호화
- 파일 태그 지정
- 파일 삭제
- 워크플로에 명명된 변수
- 태그 및 삭제 워크플로 예시

파일 복사

파일 복사 단계는 업로드된 파일의 사본을 새 Amazon S3 위치에 생성합니다. 현재는 Amazon S3에서만 파일 복사 단계를 사용할 수 있습니다.

다음 파일 복사 단계는 `file-test` 대상 버킷의 `test` 폴더에 파일을 복사합니다.

파일 복사 단계가 워크플로의 첫 단계가 아닌 경우 파일 위치를 지정할 수 있습니다. 파일 위치를 지정하면 이전 단계에서 사용한 파일 또는 업로드된 원본 파일을 복사할 수 있습니다. 이 기능을 사용하면 파일 보관 및 기록 보존을 위해 원본 파일을 그대로 유지하면서 원본 파일의 사본을 여러 개 만들 수 있습니다. 예는 태그 및 삭제 워크플로 예시를 참조하세요.



버킷 및 키 세부 정보 제공

파일 복사 단계의 대상 버킷 이름과 키를 제공해야 합니다. 키는 경로 이름 또는 파일 이름일 수 있습니다. 키를 경로 이름으로 취급할지 파일 이름으로 처리할지는 키 끝에 슬래시(/) 문자를 사용하는지 여부에 따라 결정됩니다.

마지막 문자가 /인 경우 파일은 폴더에 복사되며 이름은 변경되지 않습니다. 최종 문자가 영숫자인 경우 업로드한 파일의 이름이 키 값으로 바뀝니다. 이 경우 해당 이름의 파일이 이미 있는 경우 기존 항목 덮어쓰기 필드의 설정에 따라 동작이 달라집니다.

- 기존 파일 덮어쓰기를 선택하면 기존 파일이 처리 중인 파일로 대체됩니다.
- 기존 항목 덮어쓰기를 선택하지 않은 경우 아무 일도 일어나지 않고 워크플로 처리가 중지됩니다.

> **Tip**
>
> 동일한 파일 경로에서 동시 쓰기를 실행하면 파일을 덮어쓸 때 예상치 못한 동작이 발생할 수 있습니다.

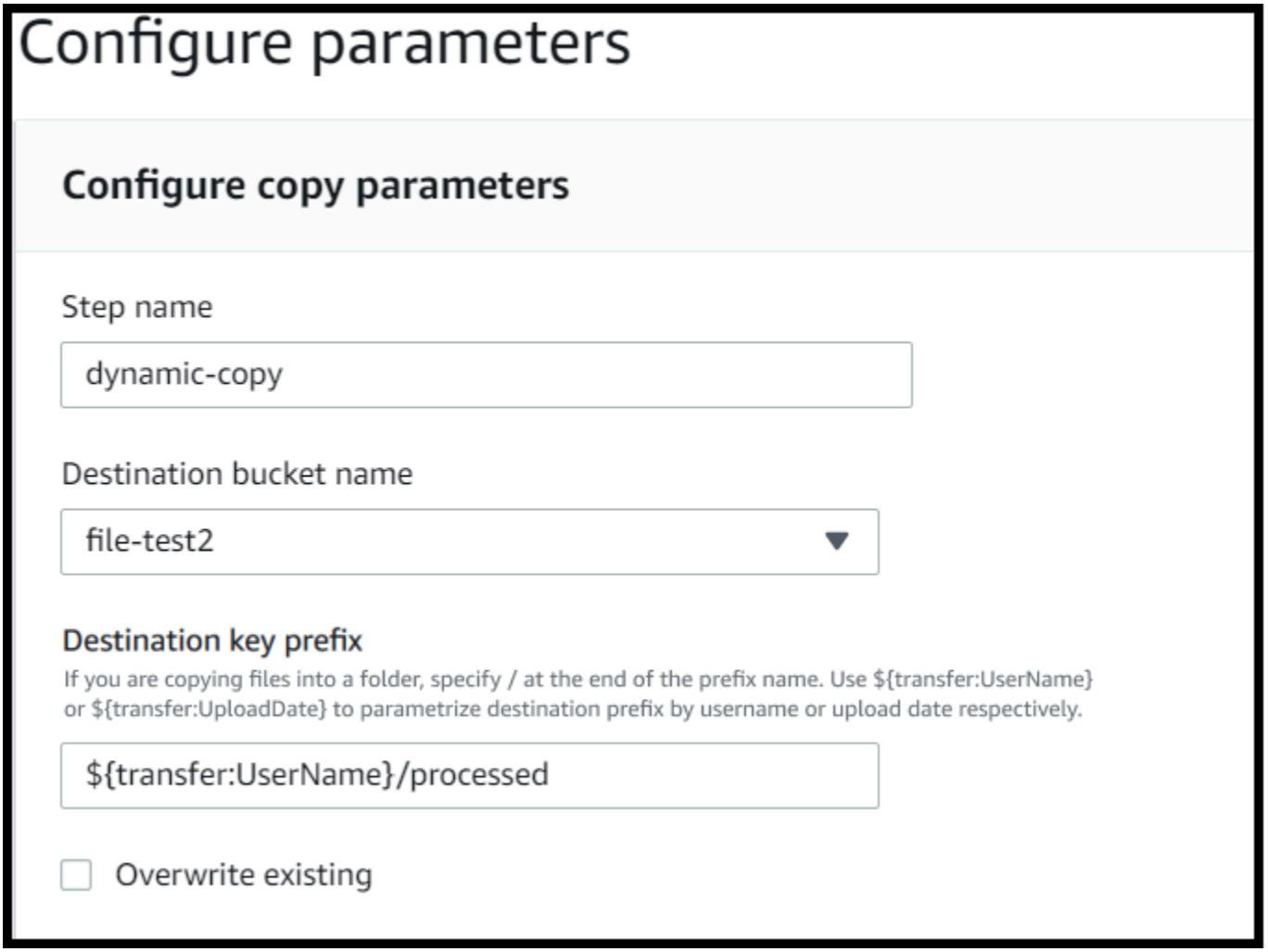
예를 들어 키 값이 `test/`인 경우 업로드한 파일이 `test` 폴더에 복사됩니다. 키 값이 `test/today`인 경우(및 기존 파일 덮어쓰기가 선택된 경우) 업로드하는 모든 파일이 `test` 폴더에 `today`(으)로 이름이 지정된 파일에 복사되고 이후의 각 파일은 이전 파일을 덮어씁니다.

> **Note**
>
> Amazon S3는 버킷과 객체를 지원하며 계층 구조가 없습니다. 하지만 객체 키 이름에 접두사와 구분 기호를 사용하여 계층 구조를 나타내고 폴더와 비슷한 방식으로 데이터를 구성할 수 있습니다.

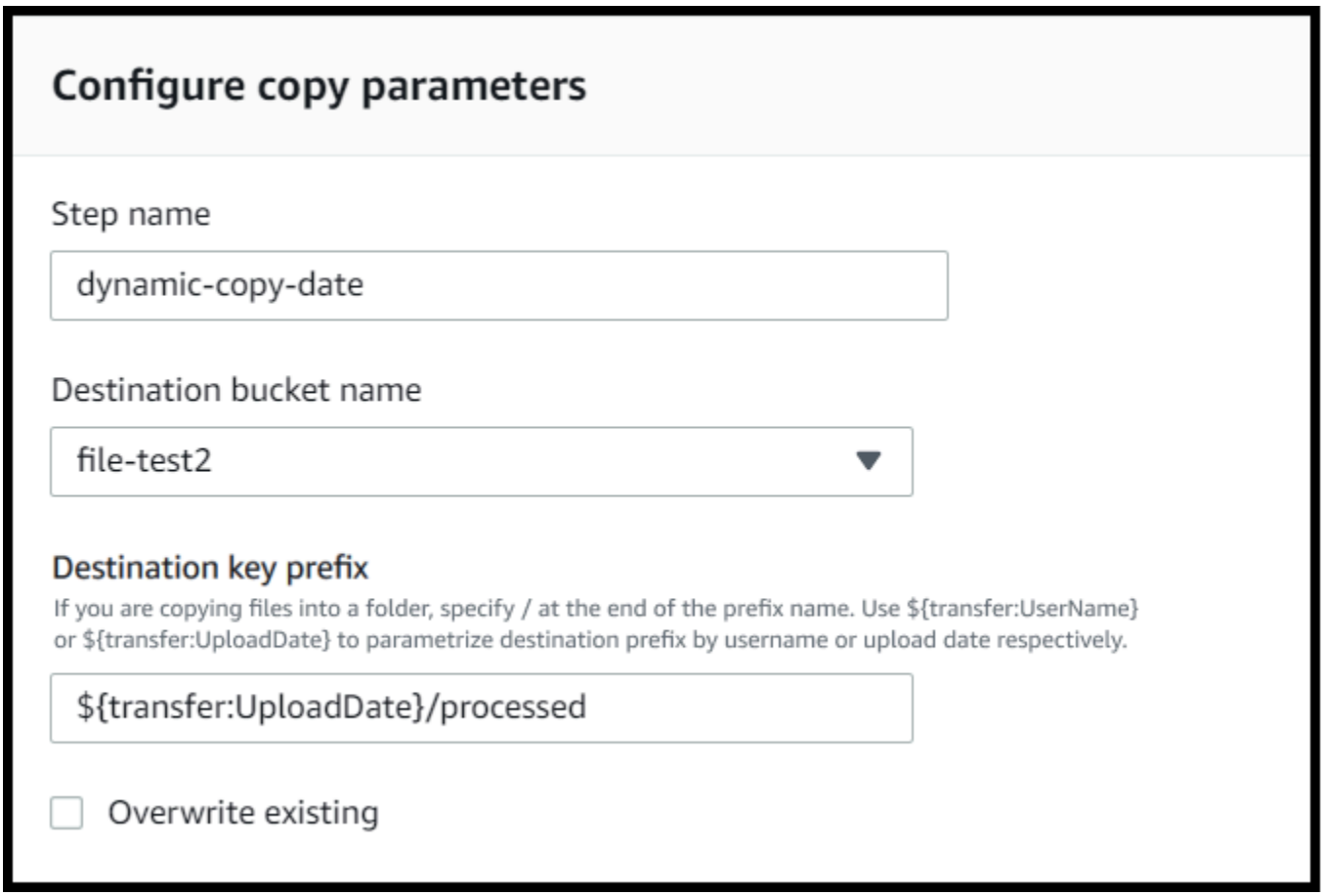
파일 복사 단계에서 이름이 지정된 변수를 사용하세요.

파일 복사 단계에서 변수를 사용하여 파일을 사용자별 폴더에 동적으로 복사할 수 있습니다. 현재는 `${transfer:UserName}` 또는 `${transfer:UploadDate}`를 변수로 사용하여 파일을 업로드하는 특정 사용자의 대상 위치 또는 현재 날짜를 기준으로 파일을 복사할 수 있습니다.

다음 예에서는 사용자가 파일을 `richard-roe` 업로드하면 파일이 `file-test2/richard-roe/processed/` 폴더에 복사됩니다. 사용자가`mary-major` 파일을 업로드하면 파일이 `file-test2/mary-major/processed/` 폴더에 복사됩니다.



마찬가지로 `${transfer:UploadDate}`를 변수로 사용하여 현재 날짜의 이름이 지정된 대상 위치에 파일을 복사할 수 있습니다. 다음 예에서 2022년 2월 1일에 대상을 `${transfer:UploadDate}/processed`으로 설정하면 업로드된 파일이 `file-test2/2022-02-01/processed/` 폴더에 복사됩니다.



이 두 변수를 함께 사용하여 기능을 조합하여 사용할 수도 있습니다. 예:

- 대상 키 접두사를 **folder/${transfer:UserName}/${transfer:UploadDate}/**(으)로 설정하면 중첩된 폴더가 생성됩니다. 예: `folder/marymajor/2023-01-05/`.
- 대상 키 접두사를 **folder/${transfer:UserName}-${transfer:UploadDate}/**(으)로 설정하여 두 변수를 연결할 수 있습니다. 예: `folder/marymajor-2023-01-05/`.

복사 단계 IAM 권한

복사 단계를 성공적으로 수행하려면 워크플로의 실행 역할에 다음 권한이 포함되어 있어야 합니다.

```
{
            "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**
>
> 이 `s3:ListBucket` 권한은 기존 항목 덮어쓰기를 선택하지 않은 경우에만 필요합니다. 이 권한은 버킷을 검사하여 같은 이름의 파일이 이미 존재하는지 확인합니다. 기존 파일 덮어쓰기를 선택한 경우 워크플로에서 파일을 확인할 필요 없이 그냥 쓰기만 하면 됩니다.
> Amazon S3 파일에 태그가 있는 경우 IAM 정책에 하나 또는 두 개의 권한을 추가해야 합니다.
>
> - 버전이 지정되지 않은 Amazon S3 파일에 `s3:GetObjectTagging`을 추가합니다.
> - 버전이 지정된 Amazon S3 파일에 `s3:GetObjectVersionTagging`을 추가합니다.

파일 복호화

AWS 스토리지 블로그에는 Transfer Family Managed 워크플로를 사용하여 코드를 작성하지 않고 파일을 간단히 해독하는 방법, PGP를 사용한 파일 암호화 및 암호 해독 방법을 설명하는 게시물이 있습니다. AWS Transfer Family

워크플로에서 PGP 복호화 사용

Transfer Family는 Pretty Good Privacy(PGP) 복호화를 기본적으로 지원합니다. SFTP, FTPS 또는 FTP를 통해 Amazon Simple Storage Service(S3) 또는 Amazon Elastic File System(Amazon EFS)에 업로드된 파일에서 PGP 복호화를 사용할 수 있습니다.

PGP 복호화를 사용하려면 파일 복호화에 사용할 PGP 프라이빗 키를 생성하고 저장해야 합니다. 그러면 사용자가 Transfer Family 서버에 파일을 업로드하기 전에 해당 PGP 암호화 키를 사용하여 파일을

암호화할 수 있습니다. 암호화된 파일을 받은 후 워크플로에서 해당 파일을 복호화할 수 있습니다. 자세한 자습서는 파일 암호 해독을 위한 관리형 워크플로우 설정를 참조하세요.

워크플로에서 PGP 복호화 사용

1. 워크플로를 호스팅할 Transfer Family 서버를 식별하거나 새 서버를 생성합니다. PGP 키를 올바른 비밀 이름으로 저장하려면 먼저 서버 ID가 있어야 합니다. AWS Secrets Manager
2. PGP 키를 필수 비밀 이름 AWS Secrets Manager 아래에 저장합니다. 자세한 내용은 PGP 키 관리를 참조하세요. 워크플로는 Secrets Manager의 암호 이름을 기반으로 복호화에 사용할 올바른 PGP 키를 자동으로 찾을 수 있습니다.

   > **Note**
   >
   > Secrets Manager에 비밀을 저장하면 AWS 계정 요금이 발생합니다. 요금에 대한 자세한 내용은 AWS Secrets Manager 요금을 참조하세요.

3. PGP 키 쌍을 사용하여 파일을 암호화합니다. (지원되는 클라이언트 목록은 지원되는 PGP 클라이언트를 참조하세요.) 명령줄을 사용하는 경우 다음 명령을 실행하세요. 이 명령을 사용하려면 *username@example.com*을 PGP 키 쌍을 만들 때 사용한 이메일 주소로 바꾸세요. *testfile.txt*를 암호화할 파일의 이름으로 바꿉니다.

   ```
   gpg -e -r username@example.com testfile.txt
   ```

4. 암호화된 파일을 Transfer Family 서버에 업로드합니다.
5. 워크플로에서 복호화 단계를 구성합니다. 자세한 내용은 복호화 단계 추가를 참조하세요.

복호화 단계 추가

복호화 단계는 워크플로의 일부로 Amazon S3 또는 Amazon EFS에 업로드된 암호화된 파일을 복호화합니다. 복호화 구성에 대한 자세한 내용은 워크플로에서 PGP 복호화 사용을 참조하세요.

워크플로의 복호화 단계를 생성할 때는 복호화한 파일의 대상을 지정해야 합니다. 대상 위치에 파일이 이미 있는 경우 기존 파일을 덮어쓸지 여부도 선택해야 합니다. Amazon CloudWatch Logs를 사용하여 암호 해독 워크플로 결과를 모니터링하고 각 파일에 대한 감사 로그를 실시간으로 가져올 수 있습니다.

해당 단계에서 파일 복호화 타입을 선택하면 파라미터 구성 페이지가 나타납니다. PGP 복호화 파라미터 구성 섹션의 값을 입력합니다.

사용 가능한 옵션은 다음과 같습니다.

- 단계 이름 - 단계를 설명하는 이름을 입력합니다.
- 파일 위치 - 파일 위치를 지정하여 이전 단계에서 사용한 파일 또는 업로드된 원본 파일을 복호화할 수 있습니다.

> **Note**
>
> 이 단계가 워크플로의 첫 번째 단계인 경우에는 이 파라미터를 사용할 수 없습니다.

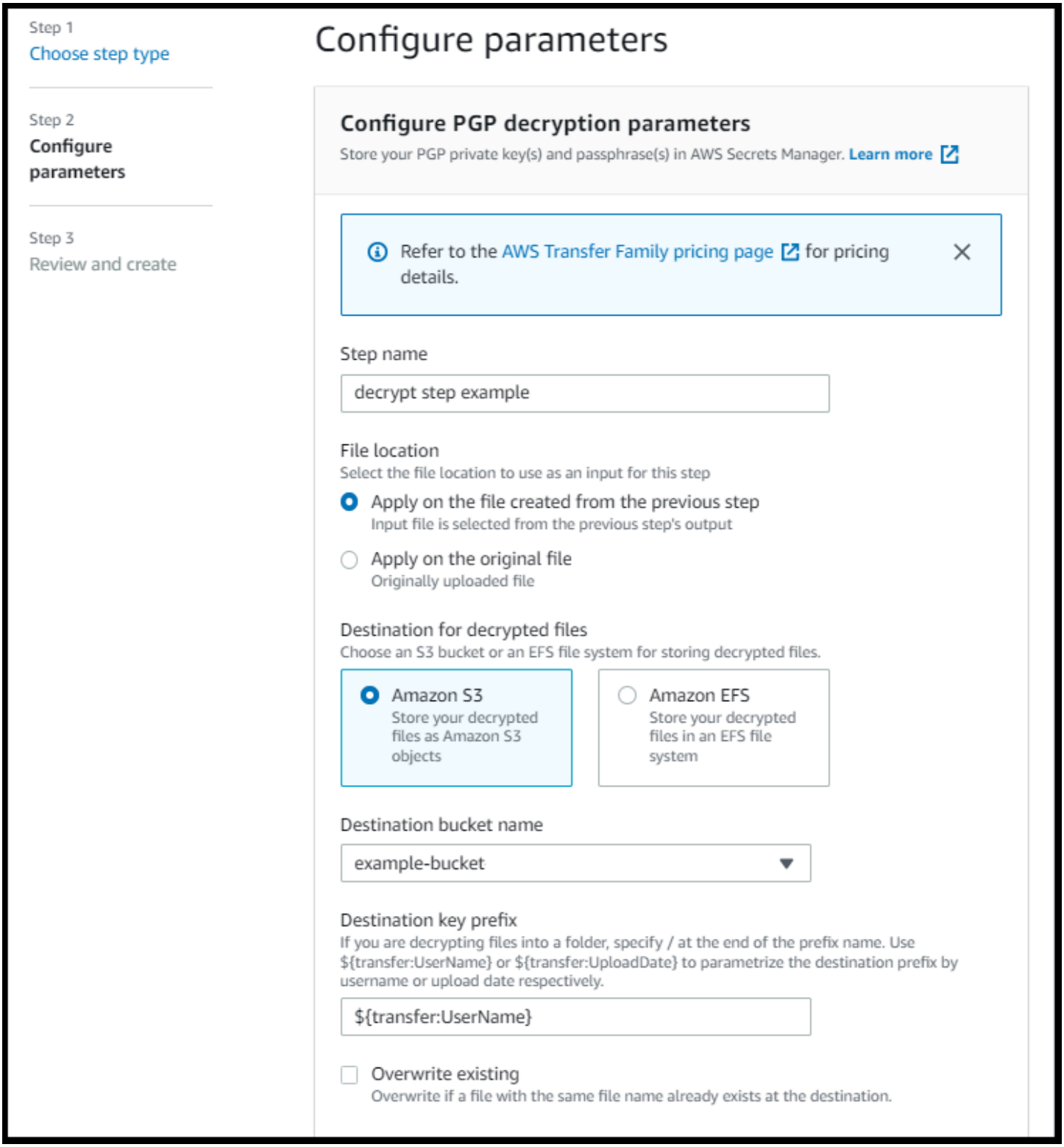
- 복호화된 파일의 대상 — Amazon S3 버킷 또는 Amazon EFS 파일 시스템을 복호화된 파일의 대상으로 선택합니다.
  - Amazon S3를 선택하는 경우 대상 버킷 이름과 대상 키 접두사를 제공해야 합니다. 사용자 이름을 기준으로 대상 키 접두사를 파라미터화하려면 대상 키 접두사에 **`${transfer:UserName}`**을 입력합니다. 마찬가지로 업로드 날짜별로 대상 키 접두사를 파라미터화하려면 대상 키 접두사에 **`${Transfer:UploadDate}`**를 입력합니다.
  - Amazon EFS를 선택하는 경우 대상 파일 시스템과 경로를 제공해야 합니다.

> **Note**
>
> 여기서 선택하는 스토리지 옵션은 이 워크플로가 연결된 Transfer Family 서버에서 사용하는 스토리지 시스템과 일치해야 합니다. 그렇지 않으면 이 워크플로를 실행하려고 시도하는 동안 오류가 발생합니다.

- 기존 파일 덮어쓰기 - 파일을 업로드했는데 같은 파일 이름을 가진 파일이 대상에 이미 있는 경우 동작은 이 파라미터의 설정에 따라 달라집니다.
  - 기존 파일 덮어쓰기를 선택하면 기존 파일이 처리 중인 파일로 대체됩니다.
  - 기존 항목 덮어쓰기를 선택하지 않은 경우 아무 일도 일어나지 않고 워크플로 처리가 중지됩니다.

> **Tip**
>
> 동일한 파일 경로에서 동시 쓰기를 실행하면 파일을 덮어쓸 때 예상치 못한 동작이 발생할 수 있습니다.

다음 스크린샷은 파일 복호화 단계에서 선택할 수 있는 옵션의 예를 보여줍니다.



복호화 단계를 위한 IAM 권한

복호화 단계를 성공적으로 수행하려면 워크플로의 실행 역할에 다음 권한이 포함되어 있어야 합니다.

```
{
            "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

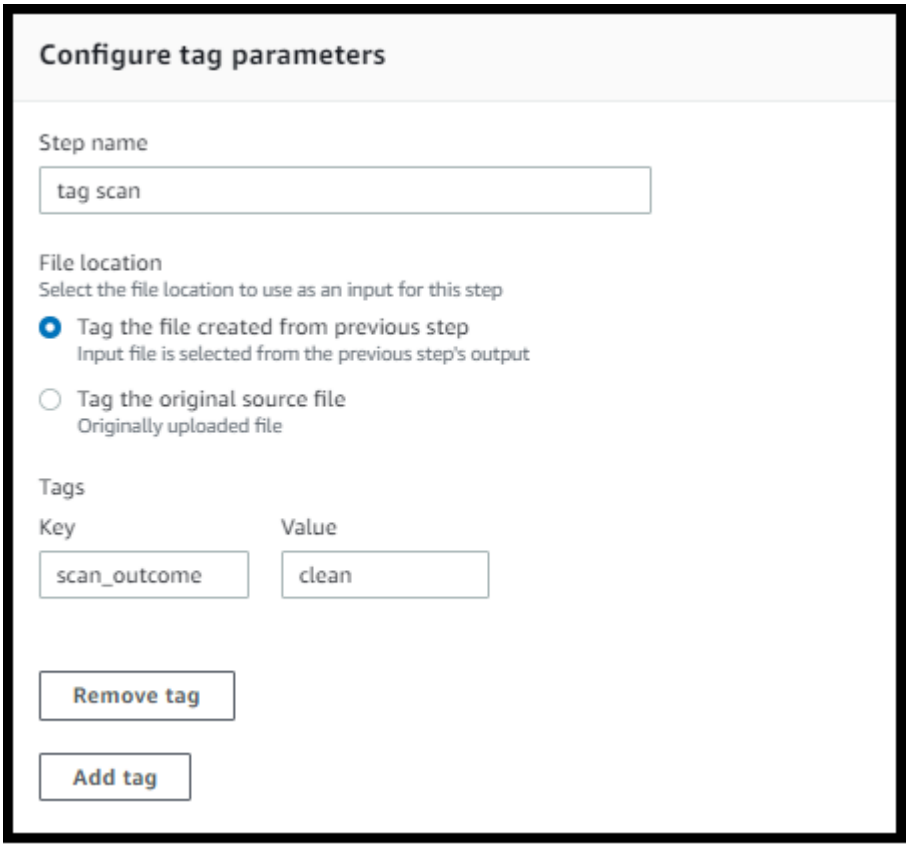
이 `s3:ListBucket` 권한은 기존 항목 덮어쓰기를 선택하지 않은 경우에만 필요합니다. 이 권한은 버킷을 검사하여 같은 이름의 파일이 이미 존재하는지 확인합니다. 기존 파일 덮어쓰기를 선택한 경우 워크플로에서 파일을 확인할 필요 없이 그냥 쓰기만 하면 됩니다.
Amazon S3 파일에 태그가 있는 경우 IAM 정책에 하나 또는 두 개의 권한을 추가해야 합니다.

- 버전이 지정되지 않은 Amazon S3 파일에 `s3:GetObjectTagging`을 추가합니다.
- 버전이 지정된 Amazon S3 파일에 `s3:GetObjectVersionTagging`을 추가합니다.

파일 태그 지정

추가 다운스트림 처리를 위해 인입 파일에 태그를 지정하려면 태그 단계를 사용하세요. 인입 파일에 할당하려는 태그의 값을 입력합니다. 현재 태그 작업은 Transfer Family 서버 스토리지로 Amazon S3를 사용하는 경우에만 지원됩니다.

다음 태그 단계 예에서는 각각 `scan_outcome` 및 `clean`를 태그 키와 값으로 할당합니다.



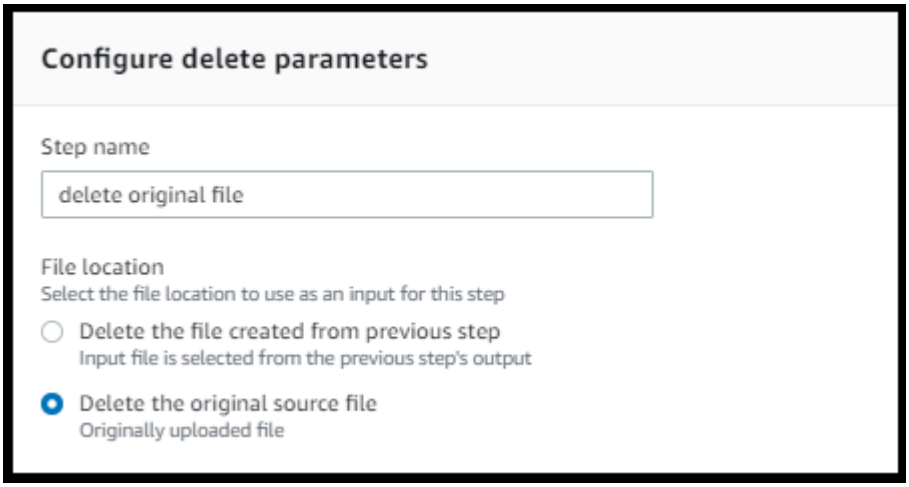
태그 단계를 성공적으로 수행하려면 워크플로의 실행 역할에 다음 권한이 포함되어 있어야 합니다.

```
{
            "Sid": "Tag",
            "Effect": "Allow",
            "Action": [
                "s3:PutObjectTagging",
                "s3:PutObjectVersionTagging"
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ]
}
```

> **Note**
>
> 워크플로에 복사 또는 복호화 단계 전에 실행되는 태그 단계가 포함되어 있는 경우 IAM 정책에 하나 또는 두 개의 권한을 추가해야 합니다.
>
> - 버전이 지정되지 않은 Amazon S3 파일에 `s3:GetObjectTagging`을 추가합니다.
> - 버전이 지정된 Amazon S3 파일에 `s3:GetObjectVersionTagging`을 추가합니다.

파일 삭제

이전 워크플로 단계에서 처리된 파일을 삭제하거나 원래 업로드된 파일을 삭제하려면 파일 삭제 단계를 사용하세요.



삭제 단계를 성공적으로 수행하려면 워크플로의 실행 역할에 다음 권한이 포함되어 있어야 합니다.

```
{
            "Sid": "Delete",
            "Effect": "Allow",
            "Action": [
                "s3:DeleteObjectVersion",
                "s3:DeleteObject"
            ],
            "Resource": "arn:aws:secretsmanager:region:account-ID:secret:aws/transfer/
*"
        }
```

워크플로에 명명된 변수

복사 및 복호화 단계의 경우 변수를 사용하여 작업을 동적으로 수행할 수 있습니다. 현재, 다음과 같은 명명된 변수를 AWS Transfer Family 지원합니다.

- 업로드하는 사용자를 기준으로 `${transfer:UserName}`을 사용하여 대상으로 파일을 복사하거나 암호를 해독합니다.
- 현재 날짜를 기준으로 `${transfer:UploadDate}`를 사용하여 대상으로 파일을 복사하거나 암호를 해독합니다.

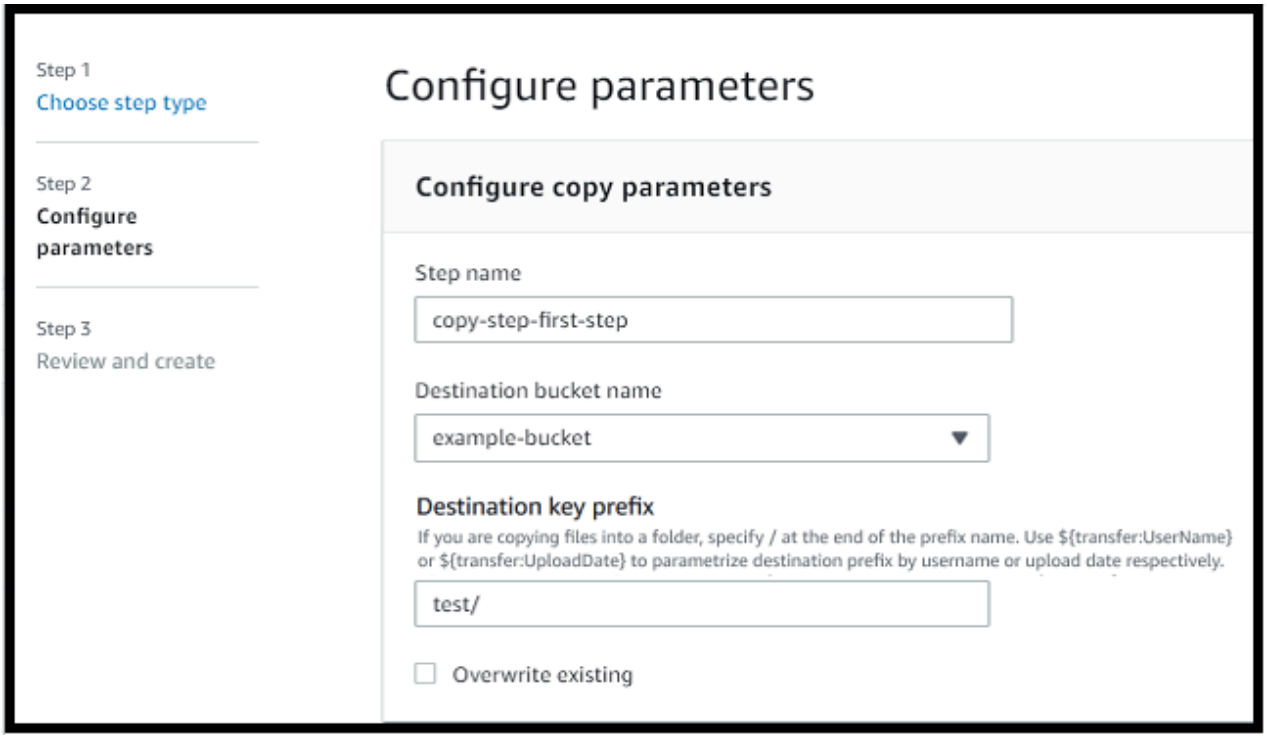
태그 및 삭제 워크플로 예시

다음 예는 데이터 분석 플랫폼과 같은 다운스트림 애플리케이션에서 처리해야 하는 수신 파일에 태그를 지정하는 워크플로를 보여줍니다. 인입 파일에 태그를 지정한 후 워크플로는 원래 업로드된 파일을 삭제하여 스토리지 비용을 절약합니다.

Console

태그 및 이동 워크플로 예

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 워크플로를 선택합니다.
3. 워크플로 페이지에서 워크플로 생성을 선택합니다.
4. 워크플로 생성 페이지에서 설명을 입력합니다. 이 설명은 워크플로 페이지에 표시됩니다.
5. 첫 번째 단계(복사)를 추가합니다.
   a. 공칭 단계 섹션에서 단계 추가를 선택합니다.
   b. 파일 복사를 선택한 후 다음을 선택합니다.
   c. 단계 이름을 입력한 다음 대상 버킷과 키 접두사를 선택합니다.



d. 다음을 선택한 다음 해당 단계에 대한 세부 정보를 검토합니다.

e. 단계 생성을 선택하여 단계를 추가하고 계속합니다.

6. 두 번째 단계(태그)를 추가합니다.

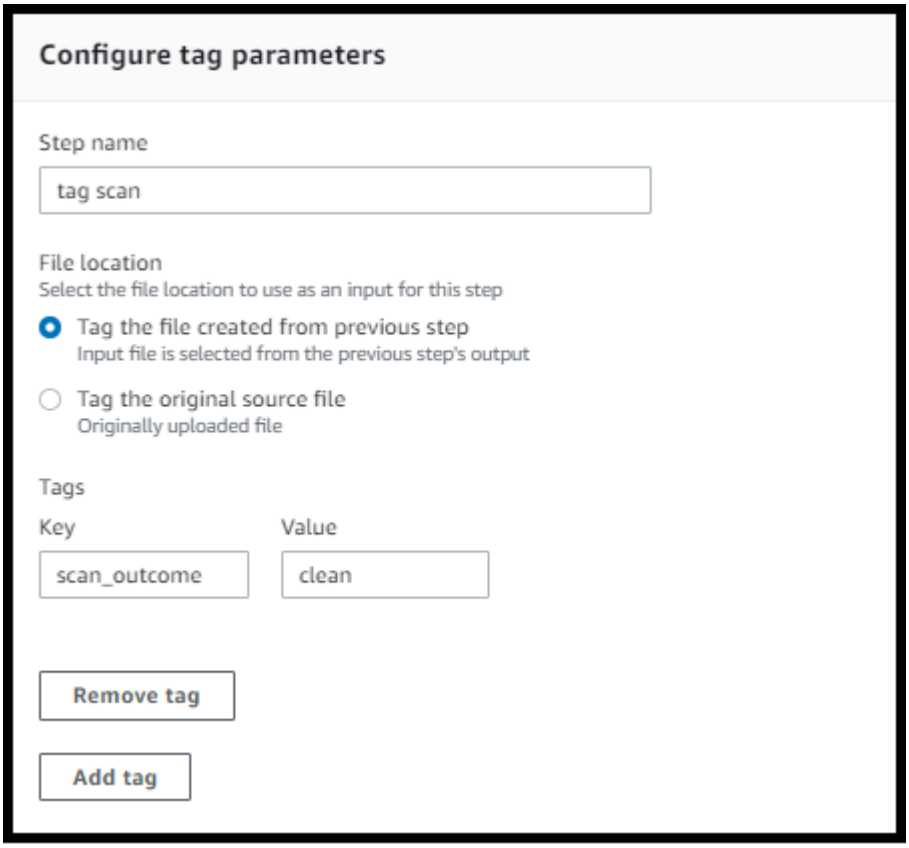
a. 공칭 단계 섹션에서 단계 추가를 선택합니다.

b. 파일 태그 지정을 선택한 후 다음을 선택합니다.

c. 단계 이름을 입력합니다.

d. 파일 위치의 경우 이전 단계에서 만든 파일에 태그 지정을 선택합니다.

e. 키와 값을 입력합니다.



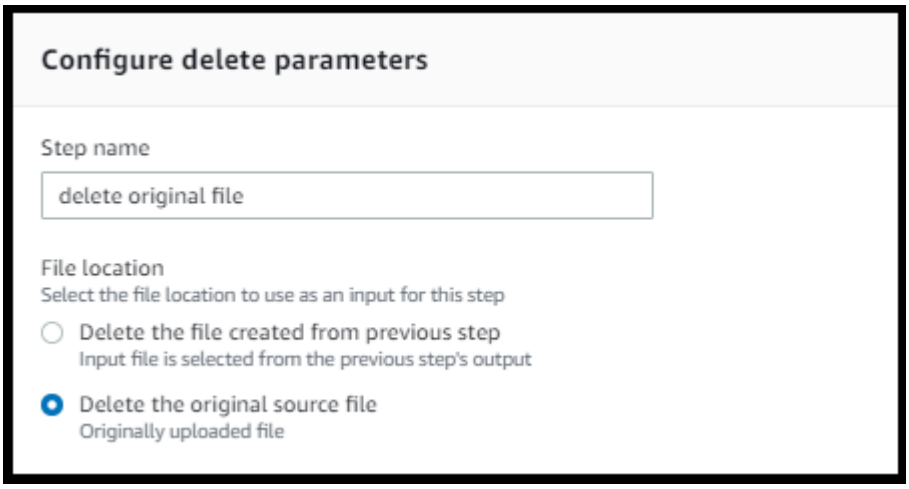
f. 다음을 선택한 다음 해당 단계에 대한 세부 정보를 검토합니다.

g. 단계 생성을 선택하여 단계를 추가하고 계속합니다.

7. 세 번째 단계 추가(삭제).

a. 공칭 단계 섹션에서 단계 추가를 선택합니다.

b. 파일 삭제를 선택하고 다음을 선택합니다.



c. 단계 이름을 입력합니다.

d. 파일 위치에서 원본 소스 파일 삭제를 선택합니다.

e. 다음을 선택한 다음 해당 단계에 대한 세부 정보를 검토합니다.

f. 단계 생성을 선택하여 단계를 추가하고 계속합니다.

8. 워크플로 구성을 검토한 다음 워크플로 생성을 선택합니다.

CLI

태그 및 이동 워크플로 예

1. 다음 코드를 파일에 저장합니다. 예: `tagAndMoveWorkflow.json`. 각 *`user input placeholder`*를 사용자의 정보로 바꿉니다.

```
[
   {
       "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}"
      }
  }
]
```

첫 번째 단계에서는 업로드된 파일을 새 Amazon S3 위치에 복사합니다. 두 번째 단계에서는 새 위치에 복사된 파일(`previous.file`)에 태그(키-값 쌍)를 추가합니다. 마지막으로 세 번째 단계에서는 원본 파일(`original.file`)을 삭제합니다.

2. 저장된 파일에서 워크플로를 생성합니다. 각 *`user input placeholder`*를 사용자의 정보로 바꿉니다.

```
aws transfer create-workflow --description "short-description" --steps
 file://path-to-file --region region-ID
```

예:

```
aws transfer create-workflow --description "copy-tag-delete workflow" --steps
 file://tagAndMoveWorkflow.json --region us-east-1
```

> **Note**
>
> 파일을 사용하여 파라미터를 로드하는 방법에 대한 자세한 내용은 파일에서 파라미터를 로드하는 방법을 참조하세요.

3. 기존 서버를 업데이트합니다.

> **Note**
>
> 이 단계에서는 이미 Transfer Family 서버가 있고 이 서버에 워크플로를 연결하려는 것으로 가정합니다. 그렇지 않은 경우 SFTP, FTPS 또는 FTP 서버 엔드포인트 구성를 참조하세요. 각 *`user input placeholder`*를 사용자의 정보로 바꿉니다.

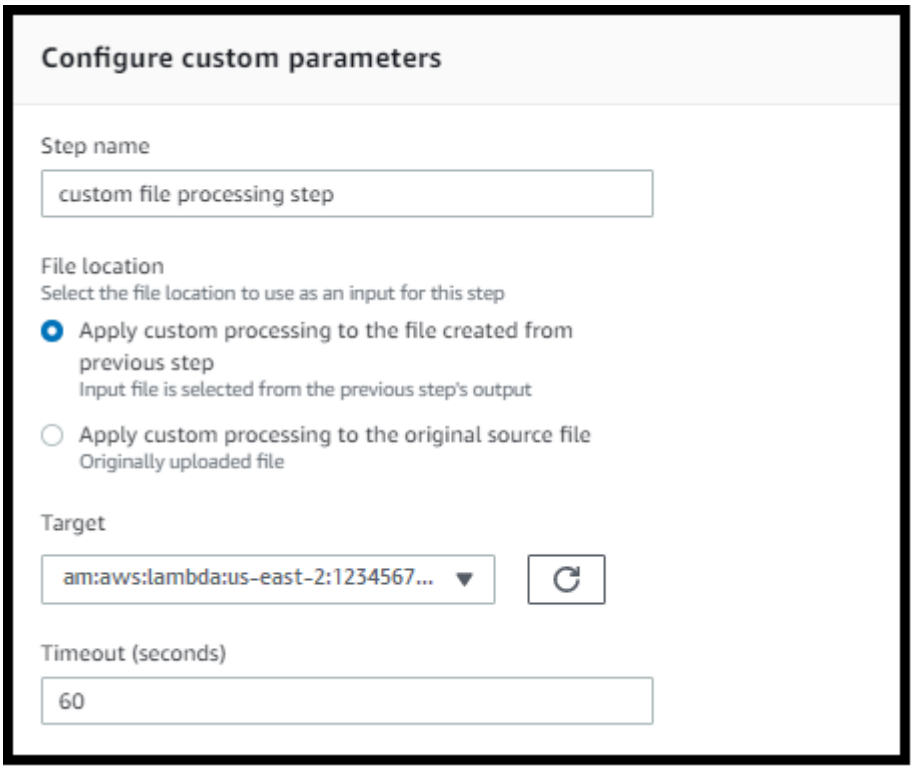
```
aws transfer update-server --server-id server-ID --region region-ID
  --workflow-details '{"OnUpload":[{ "WorkflowId": "workflow-
ID","ExecutionRole": "execution-role-ARN"}]}'
```

예:

```
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"}]}'
```

사용자 지정 파일 처리 단계 사용

사용자 지정 파일 처리 단계를 사용하면 AWS Lambda을 사용하여 Bring Your Own 파일 처리 로직을 사용할 수 있습니다. 파일이 도착하면 Transfer Family 서버는 파일 암호화, 멀웨어 검사 또는 잘못된 파일 타입 검사와 같은 사용자 지정 파일 처리 로직이 포함된 Lambda 함수를 간접적으로 호출합니다. 다음 예시에서는 대상 AWS Lambda 함수를 사용하여 이전 단계의 출력 파일을 처리합니다.



> **Note**
>
> Lambda 함수의 예는 사용자 지정 워크플로 단계를 위한 Lambda 함수 예를 참조하세요. 예를 들어 이벤트(Lambda로 전달된 파일의 위치 포함)는 파일 업로드 시 AWS Lambda 전송된 예제 이벤트를 참조하세요.

사용자 지정 워크플로 단계를 사용하여 API 작업을 SendWorkflowStepState호출하도록 Lambda 함수를 구성해야 합니다. SendWorkflowStepState단계가 성공 또는 실패 상태로 완료되었음을 워크플로 실행에 알립니다. SendWorkflowStepState API 작업의 상태는 Lambda 함수의 결과에 따라 선형 시퀀스의 예외 처리 단계 또는 공칭 단계를 간접적으로 호출합니다.

Lambda 함수가 실패하거나 제한 시간이 초과되면 단계가 실패하고 로그에서 StepErrored 확인할 수 있습니다. CloudWatch Lambda 함수가 공칭 단계의 일부이고 함수가 Status="FAILURE"로 SendWorkflowStepState에 응답하거나 제한 시간이 초과되면 흐름은 예외 처리 단계로 계속됩니다. 이 경우 워크플로는 나머지(있는 경우) 공칭 단계를 계속 실행하지 않습니다. 자세한 내용은 워크플로의 예외 처리를 참조하세요.

SendWorkflowStepState API 작업을 직접적으로 호출할 때는 다음 파라미터를 전송해야 합니다.

```
{
    "ExecutionId": "string",
    "Status": "string",
    "Token": "string",
    "WorkflowId": "string"
}
```

Lambda 함수가 실행될 때 전달되는 입력 이벤트에서 ExecutionId, Token, 및 WorkflowId를 추출할 수 있습니다(예는 다음 섹션에 표시됨). 이때 Status 값은 SUCCESS 또는 FAILURE가 될 수 있습니다.

Lambda 함수에서 SendWorkflowStepState API 작업을 호출하려면 관리형 워크플로가 도입된 이후에 게시된 SDK 버전을 사용해야 합니다. AWS

여러 Lambda 함수를 연속적으로 사용

여러 사용자 지정 단계를 차례로 사용하는 경우 파일 위치 옵션은 단일 사용자 지정 단계만 사용하는 경우와 다르게 작동합니다. Transfer Family는 Lambda 처리 파일을 다음 단계의 입력으로 사용하기 위해 다시 전달하는 것을 지원하지 않습니다. 따라서 previous.file 옵션을 사용하도록 구성된 사용자 지정 단계가 여러 개 있는 경우 모두 동일한 파일 위치(첫 번째 사용자 지정 단계의 입력 파일 위치)를 사용합니다.

> **Note**
>
> 사용자 지정 단계 이후에 사전 정의된 단계(태그 지정, 복사, 복호화 또는 삭제)가 있는 경우에도 previous.file 설정이 다르게 작동합니다. 사전 정의된 단계가 previous.file 설정을

사용하도록 구성된 경우 사전 정의된 단계는 사용자 정의 단계에서 사용한 것과 동일한 입력 파일을 사용합니다. 사용자 정의 단계에서 처리된 파일은 사전 정의된 단계로 전달되지 않습니다.

사용자 정의 처리 후 파일에 액세스

Amazon S3를 스토리지로 사용하고 있고 워크플로에 원래 업로드된 파일에서 작업을 수행하는 사용자 지정 단계가 포함되어 있는 경우, 후속 단계에서 처리된 파일에 액세스할 수 없습니다. 즉, 사용자 지정 단계 이후의 모든 단계는 사용자 지정 단계 출력의 업데이트된 파일을 참조할 수 없습니다.

예를 들어 워크플로에 다음과 같은 세 단계가 있다고 가정하겠습니다.

- 1단계 — `example-file.txt`라는 이름의 파일을 업로드합니다.
- 2단계 — 어떤 식으로든 `example-file.txt`를 변경하는 Lambda 함수를 간접적으로 호출합니다.
- 3단계 — `example-file.txt`의 업데이트된 버전에서 추가 처리를 시도합니다.

3단계에 대한 `sourceFileLocation`을 `${original.file}`로 구성한 경우 3단계에서는 1단계에서 서버가 스토리지에 파일을 업로드한 시점의 원래 파일 위치를 사용합니다. 3단계에서 `${previous.file}`을 사용하는 경우 3단계에서는 2단계에서 입력으로 사용한 파일 위치를 다시 사용합니다.

따라서 3단계에서 오류가 발생합니다. 예를 들어 3단계에서 업데이트된 `example-file.txt`를 복사하려고 하면 다음과 같은 오류 메시지가 나타납니다.

```
{
    "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"
    },
```

이 오류는 사용자 지정 단계에서 원본 파일과 일치하지 않도록 `example-file.txt`의 개체 태그(ETag)를 수정하기 때문에 발생합니다.

> **Note**
>
> Amazon EFS는 개체 태그를 사용하여 파일을 식별하지 않기 때문에 Amazon EFS를 사용하는 경우에는 이 동작이 발생하지 않습니다.

파일 업로드 시 AWS Lambda 전송된 예제 이벤트

다음 예는 파일 업로드가 AWS Lambda 완료될 때 전송되는 이벤트를 보여줍니다. 한 예로, 도메인이 Amazon S3으로 구성된 Transfer Family 서버를 사용합니다. 다른 예시에서는 도메인이 Amazon EFS를 사용하는 Transfer Family 서버를 사용합니다.

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"
    }
}
```

사용자 지정 워크플로 단계를 위한 Lambda 함수 예

다음 Lambda 함수는 실행 상태와 관련된 정보를 추출한 다음 API 작업을 SendWorkflowStepState호출하여 해당 단계의 워크플로에 상태 (또는) 를 반환합니다. SUCCESS FAILURE 함수가 SendWorkflowStepState API 작업을 직접적으로 호출하기 전에 Lambda가 워크플로 로직을 기반으로 조치를 취하도록 구성할 수 있습니다.

```
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)
    }
```

사용자 지정 단계 IAM 권한

Lambda를 직접적으로 호출하는 단계가 성공하도록 허용하려면 워크플로의 실행 역할에 다음 권한이 포함되어 있어야 합니다.

```
{
            "Sid": "Custom",
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Resource": [
                "arn:aws:lambda:region:account-id:function:function-name"
            ]
        }
```

워크플로에 대한 IAM 정책

서버에 워크플로를 추가할 때는 실행 역할을 선택해야 합니다. 서버는 워크플로를 실행할 때 이 역할을 사용합니다. 역할에 적절한 권한이 없는 경우 워크플로를 실행할 AWS Transfer Family 수 없습니다.

이 섹션에서는 워크플로를 실행하는 데 사용할 수 있는 한 가지 AWS Identity and Access Management (IAM) 권한 집합에 대해 설명합니다. 다른 예는 이 주제 후반부에서 설명합니다.

> **Note**
>
> Amazon S3 파일에 태그가 있는 경우 IAM 정책에 하나 또는 두 개의 권한을 추가해야 합니다.
>
> - 버전이 지정되지 않은 Amazon S3 파일에 `s3:GetObjectTagging`을 추가합니다.
> - 버전이 지정된 Amazon S3 파일에 `s3:GetObjectVersionTagging`을 추가합니다.

워크플로에 대한 실행 역할을 만들려면

1. 새 IAM 역할을 생성하고 이 역할에 AWS 관리형 정책을 `AWSTransferFullAccess` 추가합니다. 새 IAM 역할을 생성하는 방법에 대한 자세한 내용은 the section called “IAM 역할 및 정책 생성”을 참조하세요.
2. 다음 권한이 있는 다른 정책을 생성하여 이를 역할에 연결합니다. 각 *`user input placeholder`*를 사용자의 정보로 바꿉니다.

```
{
    "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. 이 역할을 저장하고 서버에 워크플로를 추가할 때 실행 역할로 지정하세요.

> **Note**
>
> IAM 역할을 구성할 때는 워크플로우에서 리소스에 대한 액세스를 최대한 제한할 AWS 것을 권장합니다.

워크플로 신뢰 관계

워크플로 실행 역할에는 `transfer.amazonaws.com`과의 신뢰 관계도 필요합니다. AWS Transfer Family에 대한 신뢰 관계를 설정하려면 신뢰 관계를 구축하기 위해을 참조하세요.

신뢰 관계를 구축하는 동시에 혼란스러운 대리인 문제를 피하도록 조치를 취할 수도 있습니다. 이 문제에 대한 설명과 이를 방지하는 방법의 예는 the section called “교차 서비스 혼동된 대리인 방지”을 참조하세요.

실행 역할 예시: 복호화, 복사, 태그 지정

태그 지정, 복사 및 복호화 단계가 포함된 워크플로가 있는 경우 다음 IAM 정책을 사용할 수 있습니다. 각 *user input placeholder*를 사용자의 정보로 바꿉니다.

```
{
    "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/
*"
        }
    ]
}
```

실행 역할 예시: 함수 실행 및 삭제

이 예제에는 함수를 호출하는 워크플로가 있습니다. AWS Lambda 워크플로에서 업로드된 파일을 삭제하고 이전 단계에서 실패한 워크플로 실행에 대해 조치를 취하는 예외 처리 단계가 있는 경우 다음 IAM 정책을 사용합니다. 각 *user input placeholder*를 사용자의 정보로 바꿉니다.

```
{
    "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"
            ]
        }
    ]
}
```

워크플로의 예외 처리

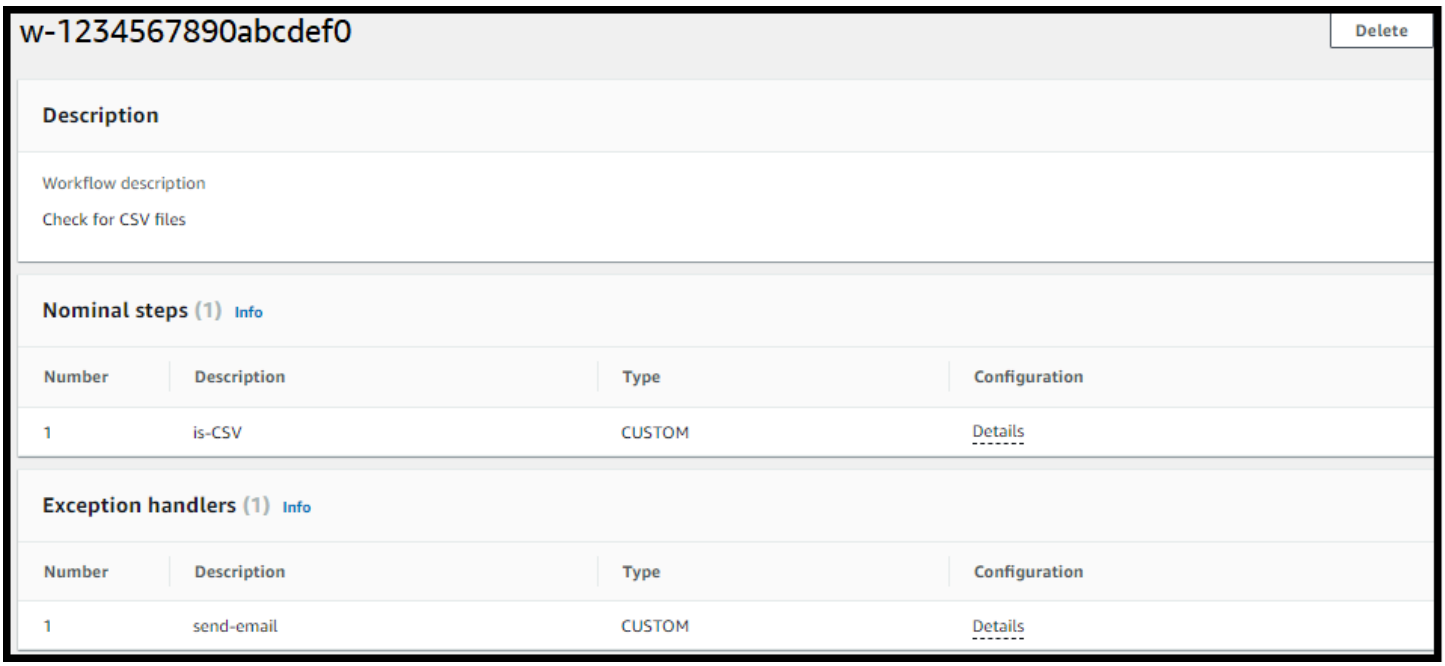
워크플로 실행 중에 오류가 발생하는 경우 지정한 예외 처리 단계가 실행됩니다. 워크플로의 공칭 단계를 지정하는 것과 같은 방식으로 워크플로의 오류 처리 단계를 지정합니다. 예를 들어 들어오는 파일의 유효성을 검사하기 위해 공칭의 단계로 사용자 지정 처리를 구성했다고 가정해 보겠습니다. 파일 검증이 실패하는 경우 예외 처리 단계를 통해 관리자에게 이메일을 보낼 수 있습니다.

다음 예 워크플로에는 두 단계가 포함되어 있습니다.

- 업로드된 파일이 CSV 형식인지 여부를 확인하는 공칭 단계 1개

- 업로드된 파일이 CSV 형식이 아니고, 공칭 단계가 실패하는 경우 이메일을 보내는 예외 처리 단계 1개

예외 처리 단계를 시작하려면 공칭 단계의 AWS Lambda 함수가 다음으로 응답해야 합니다. `Status="FAILURE"` 워크플로에서 오류를 처리하는 방법에 대한 자세한 내용은 the section called "사용자 지정 파일 처리 단계 사용"를 참조하세요.



워크플로 실행 모니터링

Amazon은 사용자가 실행하는 AWS 리소스와 애플리케이션을 AWS 클라우드 실시간으로 CloudWatch 모니터링합니다. CloudWatch Amazon을 사용하여 워크플로우에서 측정할 수 있는 변수인 지표를 수집하고 추적할 수 있습니다. CloudWatchAmazon을 사용하여 워크플로 지표와 통합 로그를 볼 수 있습니다.

CloudWatch 워크플로에 대한 로깅

CloudWatch 워크플로우 진행 상황 및 결과에 대한 통합 감사 및 로깅을 제공합니다.

워크플로에 대한 Amazon CloudWatch 로그 보기

1. https://console.aws.amazon.com/cloudwatch/ 에서 아마존 CloudWatch 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 로그를 선택한 다음, 로그 그룹을 선택합니다.
3. 로그 그룹 페이지의 탐색 표시줄에서 AWS Transfer Family 서버에 적합한 지역을 선택합니다.

4. 서버에 해당하는 로그 그룹을 선택합니다.

   예를 들어 서버 ID가 `s-1234567890abcdef0`인 경우 로그 그룹은 `/aws/transfer/s-1234567890abcdef0`입니다.

5. 서버의 로그 그룹 세부 정보 페이지에는 가장 최근의 로그 스트림이 표시됩니다. 탐색 중인 사용자에 대한 로그 스트림은 두 가지입니다.

   - 각 Secure Shell(SSH) File Transfer 프로토콜(SFTP) 세션별 로그 스트림
   - 서버에서 실행 중인 워크플로의 로그 스트림 워크플로의 로그 스트림 형식은 *username*.*workflowID*.*uniqueStreamSuffix*입니다.

   예를 들어 사용자가 `mary-major`이면 다음과 같은 로그 스트림이 있습니다.

   ```
   mary-major-east.1234567890abcdef0
   mary.w-abcdef01234567890.021345abcdef6789
   ```

   > **Note**
   >
   > 이 예에 나열된 16자리 영숫자 식별자는 가상의 식별자입니다. Amazon에서 보는 CloudWatch 값은 다릅니다.

`mary-major-usa-east.1234567890abcdef0`의 로그 이벤트 페이지에는 각 사용자 세션의 세부 정보가 표시되고 `mary.w-abcdef01234567890.021345abcdef6789` 로그 스트림에는 워크플로에 대한 세부 정보가 포함됩니다.

다음은 복사 단계가 포함된 워크플로(`w-abcdef01234567890`)를 기반으로 하는 `mary.w-abcdef01234567890.021345abcdef6789`의 샘플 로그 스트림입니다.

```
{
    "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 워크플로에 대한 지표

AWS Transfer Family 워크플로에 대한 여러 지표를 제공합니다. 이전 1분 간 시작, 성공적으로 완료, 및 실패한 워크플로 실행 수에 대한 지표를 볼 수 있습니다. Transfer Family에 대한 모든 CloudWatch 지표는 에 설명되어 Transfer Family에 대한 CloudWatch 측정항목 사용 있습니다.

템플릿에서 워크플로 생성

템플릿에서 워크플로와 서버를 생성하는 AWS CloudFormation 스택을 배포할 수 있습니다. 이 절차에는 워크플로를 빠르게 배포하는 데 사용할 수 있는 예가 포함되어 있습니다.

AWS Transfer Family 워크플로와 서버를 생성하는 AWS CloudFormation 스택을 만들려면

1. https://console.aws.amazon.com/cloudformation 에서 AWS CloudFormation 콘솔을 엽니다.
2. 다음 코드를 파일에 저장합니다.

   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. 다음 항목을 실제 값으로 대체합니다.

   - *workflow-execution-role-arn*을 실제 워크플로 실행 역할의 ARN으로 바꿉니다. 예제: `arn:aws:transfer:us-east-2:111122223333:workflow/w-1234567890abcdef0`
   - `arn:aws:lambda:`*region*`:`*account-id*`:function:`*function-name*을 Lambda 함수 ARN으로 바꿉니다. 예를 들어 `arn:aws:lambda:us-east-2:123456789012:function:example-lambda-idp`입니다.

4. 사용 AWS CloudFormation 설명서의 AWS CloudFormation 스택 템플릿 선택에 나와 있는 기존 템플릿에서 스택을 배포하는 방법에 대한 지침을 따르십시오.

스택을 배포한 후에는 CloudFormation 콘솔의 출력 탭에서 스택에 대한 세부 정보를 볼 수 있습니다. 템플릿은 서비스 관리 사용자를 사용하는 새 AWS Transfer Family SFTP 서버와 새 워크플로를 만들고 워크플로를 새 서버에 연결합니다.

Transfer Family 서버에서 워크플로 제거

워크플로를 Transfer Family 서버와 연결했는데 이제 해당 연결을 제거하려는 경우 콘솔을 사용하거나 프로그래밍 방식으로 제거할 수 있습니다.

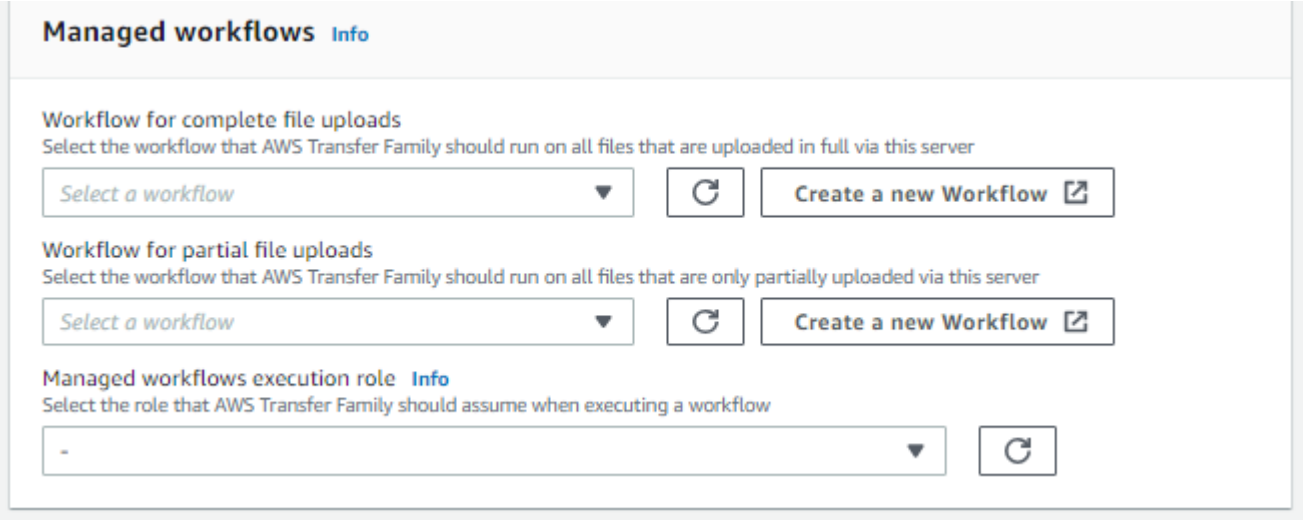
Console

Transfer Family 서버에서 워크플로 제거

1. https://console.aws.amazon.com/transfer/ 에서 콘솔을 엽니다. AWS Transfer Family
2. 왼쪽 탐색 창에서 서버를 선택합니다.
3. 서버 ID 열에서 서버 식별자를 선택합니다.
4. 서버의 세부 정보 페이지에서 추가 세부 정보 섹션까지 아래로 스크롤한 다음 편집을 선택합니다.
5. 추가 세부 정보 편집 페이지의 관리형 워크플로 섹션에서 모든 설정에 대한 정보를 지웁니다.
   - 전체 파일 업로드를 위한 워크플로의 워크플로 목록에서 대시(-)를 선택합니다.
   - 아직 지우지 않은 경우 부분 파일 업로드 워크플로의 워크플로 목록에서 대시(-)를 선택합니다.
   - 관리형 워크플로 실행 역할의 역할 목록에서 대시(-)를 선택합니다.

   대시가 보이지 않는 경우 각 메뉴의 첫 번째 값이므로 대시가 보일 때까지 위로 스크롤하세요.

   결과는 다음과 같아야 합니다.



6. 변경 내용을 저장하려면 아래로 스크롤하여 저장을 선택합니다.

CLI

`update-server`(또는 API용 `UpdateServer`) 직접 호출을 사용하고 `OnUpload` 및 `OnPartialUpload` 파라미터에 빈 인수를 제공합니다.

에서 AWS CLI다음 명령을 실행합니다.

```
aws transfer update-server --server-id your-server-id --workflow-details
 '{"OnPartialUpload":[],"OnUpload":[]}'
```

*your-server-id*를 서버의 ID로 바꿉니다. 예를 들어, 서버 ID가 `s-01234567890abcdef`인 경우 명령은 다음과 같습니다.

```
aws transfer update-server --server-id s-01234567890abcdef --workflow-details
 '{"OnPartialUpload":[],"OnUpload":[]}'
```

관리형 워크플로 제한 및 제약 조건

제한 사항

현재 다음과 같은 제한 사항이 AWS Transfer Family의 업로드 후 처리 워크플로에 적용됩니다.

- 교차 계정 및 지역 간 AWS Lambda 함수는 지원되지 않습니다. 하지만 AWS Identity and Access Management (IAM) 정책이 올바르게 구성되어 있으면 계정 간에 복사할 수 있습니다.

- 모든 워크플로 단계에서 워크플로로 액세스하는 모든 Amazon S3 버킷은 워크플로와 동일한 리전에 있어야 합니다.
- 복호화 단계의 경우 복호화 대상이 리전 및 지원 스토어의 원본과 일치해야 합니다(예를 들어, 해독할 파일이 Amazon S3에 저장되어 있는 경우 지정된 대상도 Amazon S3에 있어야 함).
- 비동기 사용자 지정 단계만 지원됩니다.
- 사용자 지정 단계 제한 시간은 근사값입니다. 즉, 지정된 제한 시간보다 약간 더 오래 걸릴 수 있습니다. 또한 워크플로는 Lambda 함수에 따라 달라집니다. 따라서 실행 중에 함수가 지연되는 경우 워크플로는 지연을 인식하지 못합니다.
- 제한 한도를 초과하는 경우 Transfer Family는 대기열에 워크플로 작업을 추가하지 않습니다.
- 크기가 0인 파일에 대해서는 워크플로가 시작되지 않습니다. 크기가 0보다 큰 파일은 관련 워크플로를 시작합니다.

제한 사항

또한 Transfer Family의 워크플로에는 다음과 같은 기능 제한이 적용됩니다.

- 리전별, 계정별 워크플로 수는 10개로 제한됩니다.
- 사용자 지정 단계의 최대 제한 시간은 30분입니다.
- 워크플로의 최대 단계 수는 8개입니다.
- 워크플로당 최대 태그 수는 50개입니다.
- 복호화 단계가 포함된 최대 동시 실행 수는 워크플로당 250개입니다.
- 사용자당 Transfer Family 서버당 최대 3개의 PGP 프라이빗 키를 저장할 수 있습니다.
- 복호화된 파일의 최대 크기는 10GB입니다.
- 버스트 용량이 100이고 리필 비율이 1인 토큰 버킷 시스템을 사용하여 새 실행률을 제한합니다.
- 서버에서 워크플로를 제거하고 새 워크플로로 바꾸거나 워크플로의 실행 역할에 영향을 미치는 서버 구성을 업데이트할 때마다 새 워크플로를 실행하기 전에 약 10분을 기다려야 합니다. Transfer Family 서버는 워크플로 세부 정보를 캐시하며 서버가 캐시를 새로 고치는 데 10분이 걸립니다.

  또한 활성 SFTP 세션에서 로그아웃한 다음 10분 대기 시간이 지난 후 다시 로그인해야 변경 사항을 확인할 수 있습니다.

서버 관리

이 섹션에서는 서버 목록을 보는 방법, 서버 세부 정보를 보는 방법, 서버 세부 정보를 편집하는 방법, SFTP 지원 서버의 호스트 키를 변경하는 방법에 대한 정보를 찾을 수 있습니다.

주제

서버 목록 보기

AWS Transfer Family 콘솔에서 선택한 AWS 지역에 위치한 모든 서버의 목록을 찾을 수 있습니다.

AWS 지역에 있는 서버 목록을 찾으려면

- https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.

  현재 AWS 지역에 하나 이상의 서버가 있는 경우 콘솔이 열리고 서버 목록이 표시됩니다. 서버 목록이 보이지 않으면, 올바른 AWS 리전에 있는지 확인하세요. 탐색 창에서 서버를 선택할 수도 있습니다.

  로그의 세부 정보 보기에 대한 자세한 내용은 SFTP, FTPS 및 FTP 서버 세부 정보 보기를 참조하세요.

서버 삭제

이 절차에서는 AWS Transfer Family 콘솔 또는 를 사용하여 Transfer Family 서버를 삭제하는 방법에 대해 설명합니다 AWS CLI.

> **Important**
>
> 엔드포인트에 액세스할 수 있도록 설정된 각 프로토콜에 대해 서버를 삭제할 때까지 요금이 청구됩니다.

> **Warning**
>
> 서버를 삭제하면 해당 사용자가 모두 삭제됩니다. 서버를 사용하여 액세스한 버킷의 데이터는 삭제되지 않으며 해당 Amazon S3 버킷에 대한 권한을 가진 AWS 사용자가 계속 액세스할 수 있습니다.

Console

콘솔을 사용하여 서버를 삭제하려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 서버를 선택합니다.
3. 삭제할 서버의 확인란을 선택합니다.
4. 작업에 대해 삭제를 선택합니다.
5. 표시되는 확인 대화 상자에서 단어 **delete**를 입력한 다음 삭제를 선택하여 서버 삭제를 확인합니다.

서버 페이지에서 서버가 삭제되며 더 이상 요금이 청구되지 않습니다.

AWS CLI

CLI를 사용하여 서버를 삭제하려면

1. (선택 사항) 다음 명령을 실행하여 영구적으로 삭제하려는 서버의 세부 정보를 확인합니다.

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

   이 `describe-server` 명령은 서버의 모든 세부 정보를 반환합니다.
2. 다음 명령을 실행하여 서버를 삭제합니다.

```
aws transfer delete-server --server-id your-server-id
```

성공하면 이 명령은 서버를 삭제하고 정보를 반환하지 않습니다.

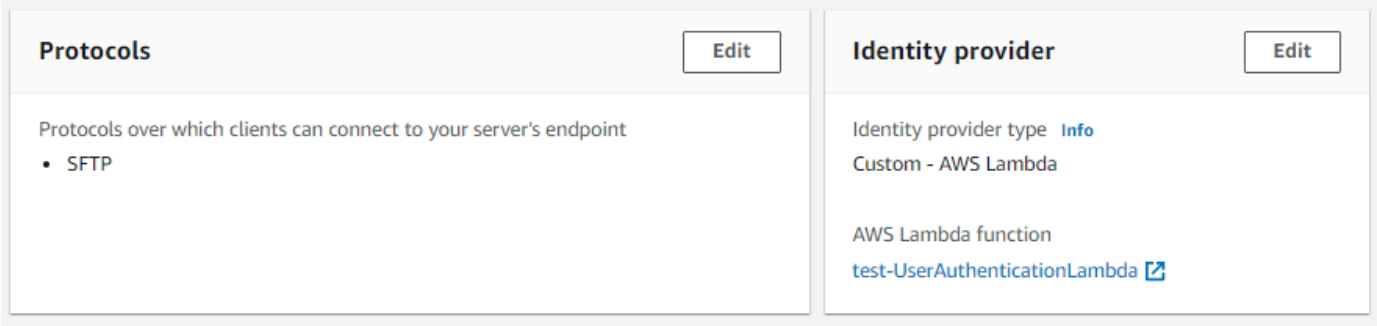
SFTP, FTPS 및 FTP 서버 세부 정보 보기

개별 서버의 세부 정보 및 속성 목록을 찾을 수 있습니다. AWS Transfer Family 서버 속성에는 프로토콜, 자격 증명 제공자, 상태, 엔드포인트 타입, 사용자 지정 호스트 이름, 엔드포인트, 사용자, 로깅 역할, 서버 호스트 키 및 태그가 포함됩니다.

서버 세부 정보를 보려면

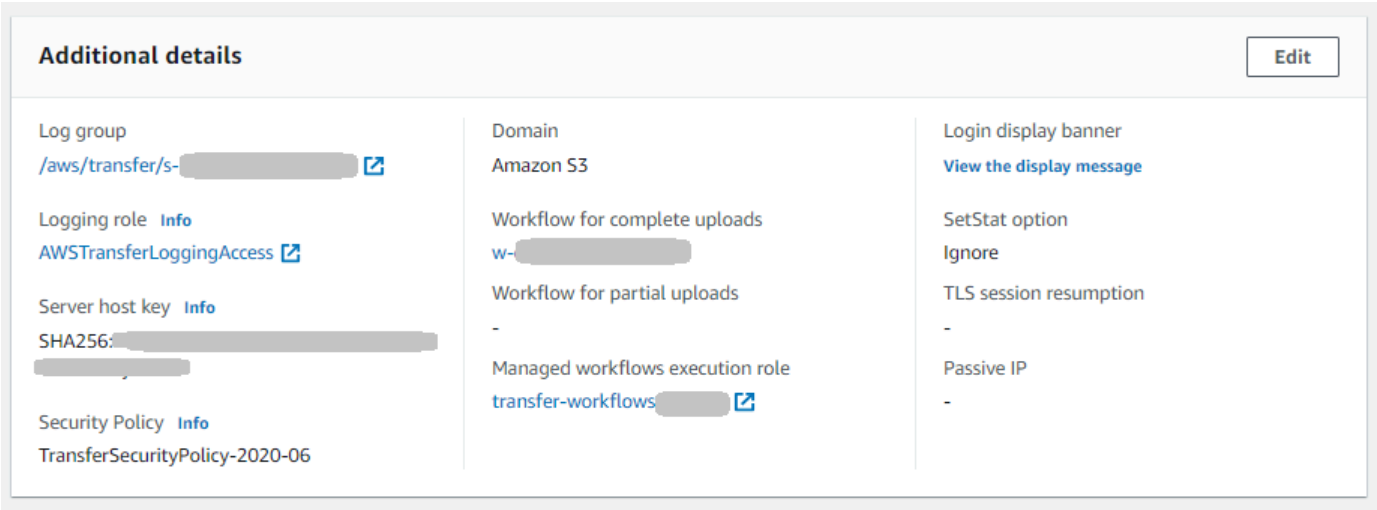
1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 탐색 창에서 서버를 선택합니다.
3. 서버 ID 열에서 식별자를 선택하여 다음과 같은 서버 세부 정보 페이지를 표시합니다.

   편집을 선택하여 이 페이지에서 서버의 속성을 변경할 수 있습니다. 서버 편집에 대한 자세한 내용은 서버 세부 정보 편집를 참조하세요. AS2 서버의 세부 정보 페이지는 약간 다릅니다. AS2 서버의 경우 을 참조하십시오. AS2 서버 세부 정보 보기



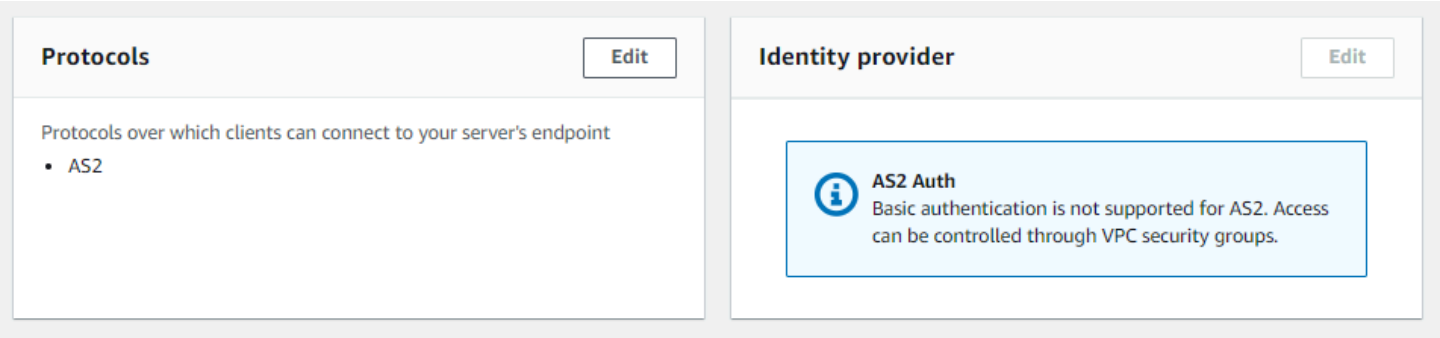
> **Note**
>
> 서버 호스트 키 설명 및 가져온 날짜 값은 2022년 9월 기준으로 새로 추가되었습니다. 이러한 값은 다중 호스트 키 기능을 지원하기 위해 도입되었습니다. 이 기능을 사용하려면 여러 호스트 키가 도입되기 전에 사용 중이던 단일 호스트 키를 모두 마이그레이션해야 했습니다.

마이그레이션된 서버 호스트 키의 가져온 날짜 값은 서버의 마지막 수정 날짜로 설정됩니다. 즉, 마이그레이션된 호스트 키로 표시되는 날짜는 어떤 식으로든 서버 호스트 키 마이그레이션 전에 서버를 마지막으로 수정한 날짜와 일치합니다.
마이그레이션된 유일한 키는 가장 오래된 또는 유일한 서버 호스트 키입니다. 모든 추가 키에는 가져온 시점의 실제 날짜가 표시됩니다. 또한 마이그레이션된 키에는 마이그레이션된 키임을 쉽게 식별할 수 있는 설명이 있습니다.
마이그레이션은 9월 2일에서 9월 13일 사이에 이루어졌습니다. 이 범위 내의 실제 마이그레이션 날짜는 서버 리전에 따라 다릅니다.

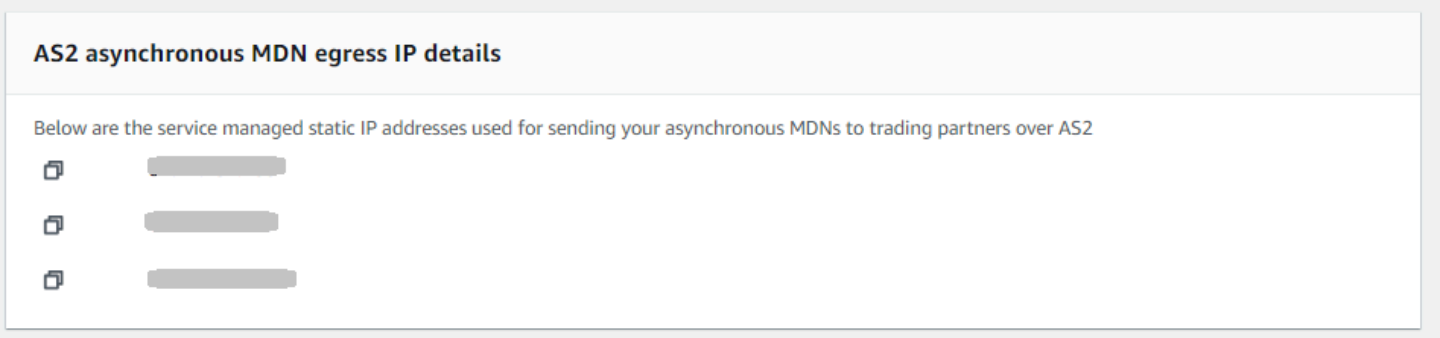


AS2 서버 세부 정보 보기

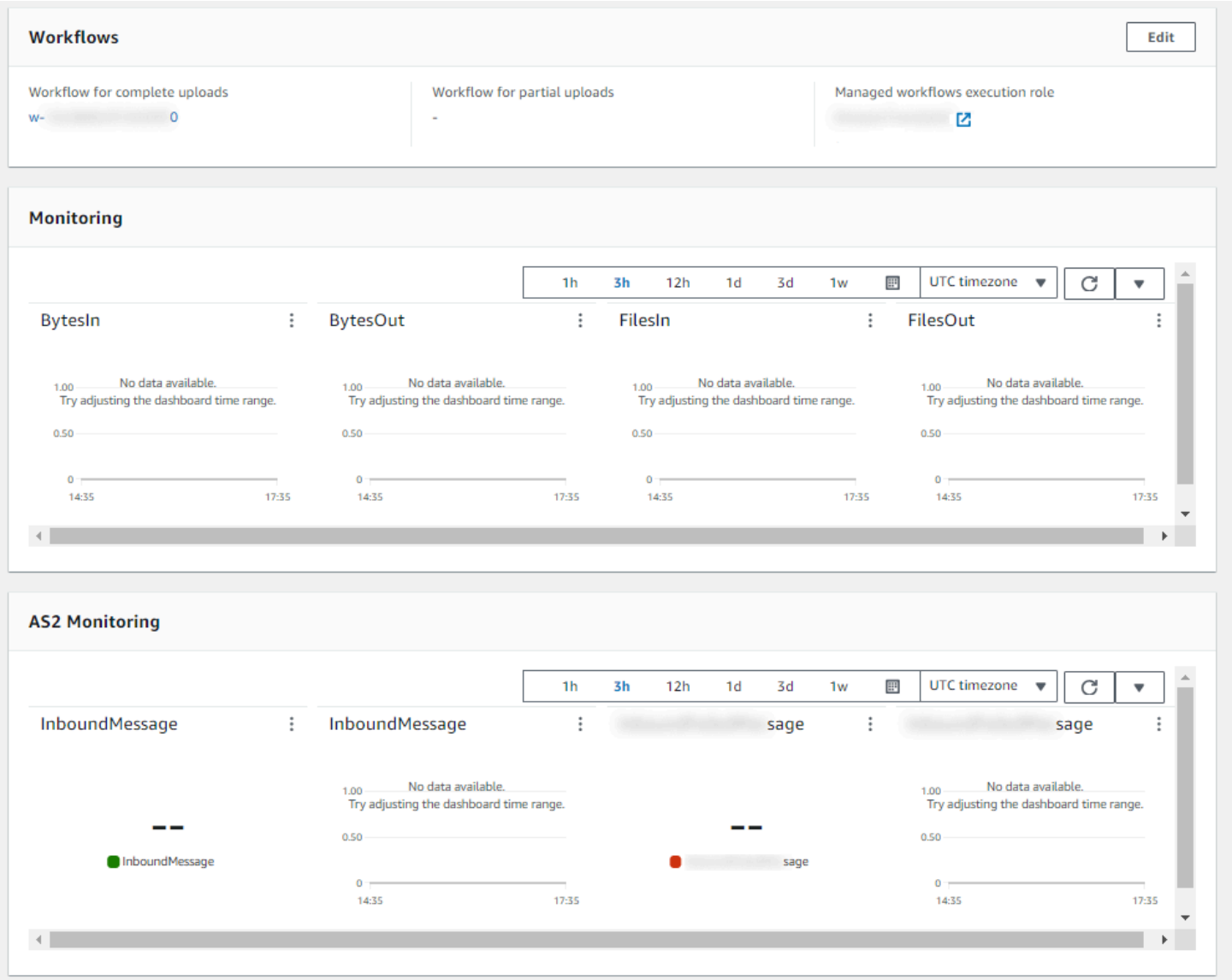
개별 AWS Transfer Family 서버의 세부 정보 및 속성 목록을 찾을 수 있습니다. 서버 속성에는 프로토콜, 상태 등이 포함됩니다. AS2 서버의 경우 AS2 비동기 MDN 송신 IP 주소도 볼 수 있습니다.



각 AS2 서버에는 세 개의 고정 IP 주소가 할당됩니다. 이 IP 주소를 사용하여 AS2를 통해 거래 파트너에게 비동기 mDN을 보낼 수 있습니다.



AS2 서버 세부 정보 페이지 하단에는 첨부된 모든 워크플로와 모니터링 및 태깅 정보에 대한 세부 정보가 있습니다.



서버 세부 정보 편집

서버를 생성한 후 AWS Transfer Family 서버 구성을 편집할 수 있습니다.

주제

서버의 구성을 편집하는 방법

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 서버를 선택합니다.
3. 서버 ID 열에서 식별자를 선택하여 다음과 같은 서버 세부 정보 페이지를 표시합니다.

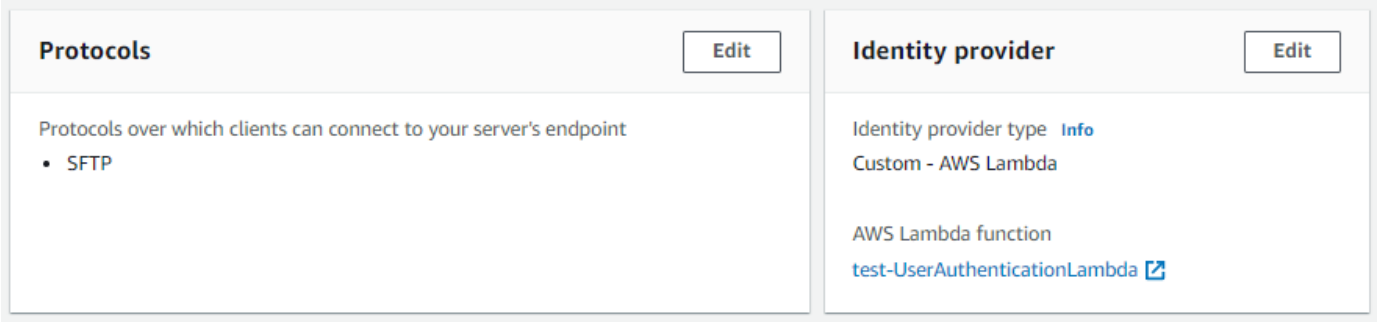
   이 페이지에서 편집을 선택하여 서버의 속성을 변경할 수 있습니다.

   - 프로토콜을 변경하려면 File Transfer 프로토콜 편집를 참조하세요.
   - 자격 증명 공급자의 경우 서버 생성 후에는 서버의 자격 증명 공급자 타입을 변경할 수 없습니다. 자격 증명 공급자를 변경하려면, 서버를 삭제하고 원하는 자격 증명 공급자를 이용해 새 서버를 만들어야 합니다.

     > **Note**
     >
     > 서버에서 사용자 지정 ID 공급자를 사용하는 경우 일부 속성을 편집할 수 있습니다. 자세한 내용은 사용자 지정 자격 증명 제공자 파라미터 편집를 참조하세요.

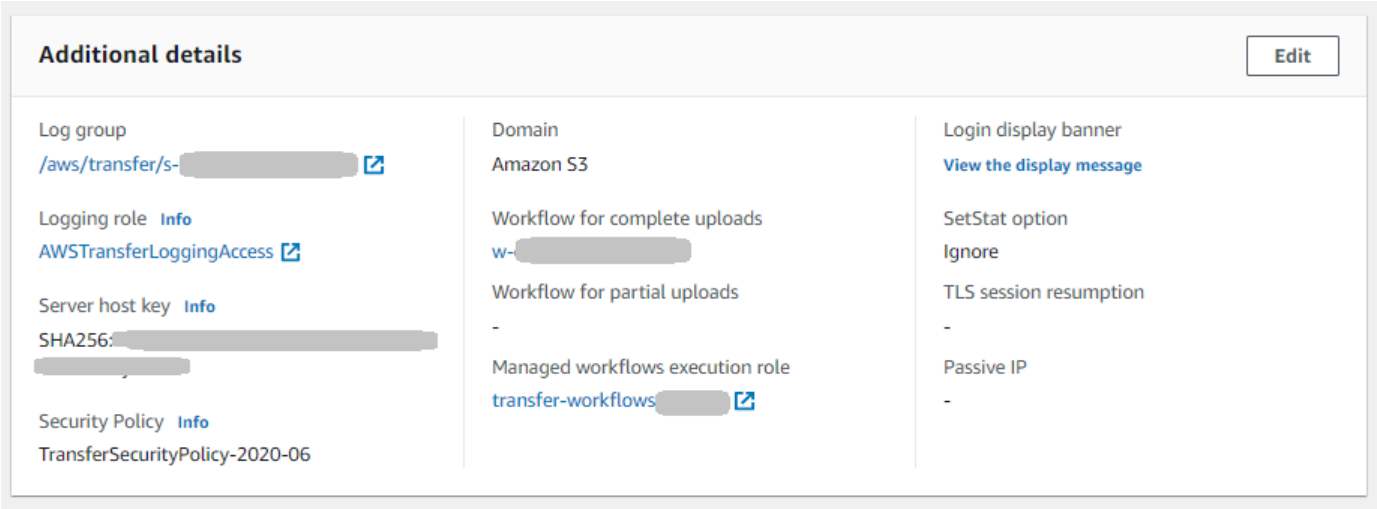
   - 엔드포인트 타입 또는 사용자 지정 호스트 이름을 변경하려면 서버 엔드포인트 편집를 참조하세요.
   - 계약을 추가하려면 먼저 AS2를 프로토콜로 서버에 추가해야 합니다. 자세한 내용은 File Transfer 프로토콜 편집를 참조하세요.

- 서버의 호스트 키를 관리하려면 SFTP 지원 서버의 호스트 키 관리를 참조하세요.
- 추가 세부 정보에서 다음 정보를 편집할 수 있습니다.
  - 로깅 역할을 변경하려면 로깅 구성 편집를 참조하세요.
  - 보안 정책을 변경하려면 보안 정책 편집를 참조하세요.
  - 서버 호스트 키를 변경하려면 SFTP 지원 서버의 호스트 키 관리를 참조하세요.
  - 서버의 관리 워크플로를 변경하려면 서버의 관리형 워크플로 변경를 참조하세요.
  - 서버의 디스플레이 배너를 편집하려면 서버의 디스플레이 배너 변경를 참조하세요.
- 추가 구성 에서 다음 정보를 편집할 수 있습니다.
  - SetStat 옵션: Amazon S3 SETSTAT 버킷에 업로드하는 파일에서 클라이언트가 사용을 시도할 때 생성되는 오류를 무시하려면 이 옵션을 활성화합니다. 자세한 내용은 해당 ProtocolDetails주제의 `SetStatOption` 설명서를 참조하십시오.
  - TLS 세션 재개: FTPS 세션에 대한 통제 및 데이터 연결 간에 협상된 암호 키를 재개하거나 공유하는 메커니즘을 제공합니다. 자세한 내용은 해당 ProtocolDetails항목의 `TlsSessionResumptionMode` 설명서를 참조하십시오.
  - 수동 IP: FTP 및 FTPS 프로토콜에 대한 수동 모드를 나타냅니다. 방화벽, 라우터 또는 로드 밸런서의 퍼블릭 IP 주소와 같은 단일 IPv4 주소를 입력합니다. 자세한 내용은 해당 ProtocolDetails항목의 `PassiveIp` 설명서를 참조하십시오.
- 서버를 시작 또는 중지하려면 서버를 온라인이나 오프라인으로 전환를 참조하세요.
- 서버를 삭제하려면 서버 삭제를 참조하세요.
- 사용자 속성을 편집하려면 액세스 통제 관리를 참조하세요.



> **Note**
>
> 서버 호스트 키 설명 및 가져온 날짜 값은 2022년 9월 기준으로 새로 추가되었습니다. 이러한 값은 다중 호스트 키 기능을 지원하기 위해 도입되었습니다. 이 기능을 사용하려면

여러 호스트 키가 도입되기 전에 사용 중이던 단일 호스트 키를 모두 마이그레이션해야 했습니다.
마이그레이션된 서버 호스트 키의 가져온 날짜 값은 서버의 마지막 수정 날짜로 설정됩니다. 즉, 마이그레이션된 호스트 키로 표시되는 날짜는 어떤 식으로든 서버 호스트 키 마이그레이션 전에 서버를 마지막으로 수정한 날짜와 일치합니다.
마이그레이션된 유일한 키는 가장 오래된 또는 유일한 서버 호스트 키입니다. 모든 추가 키에는 가져온 시점의 실제 날짜가 표시됩니다. 또한 마이그레이션된 키에는 마이그레이션된 키임을 쉽게 식별할 수 있는 설명이 있습니다.
마이그레이션은 9월 2일에서 9월 13일 사이에 이루어졌습니다. 이 범위 내의 실제 마이그레이션 날짜는 서버 리전에 따라 다릅니다.



File Transfer 프로토콜 편집

AWS Transfer Family 콘솔에서 파일 전송 프로토콜을 편집할 수 있습니다. File Transfer 프로토콜은 클라이언트를 서버의 엔드포인트에 연결합니다.

프로토콜을 편집하려면

1. 서버 세부 정보 페이지에서 프로토콜 옆의 편집을 선택합니다.
2. 프로토콜 편집 페이지에서 프로토콜 확인란 또는 확인란을 선택하거나 선택 취소하여 다음 파일 전송 프로토콜을 추가하거나 제거합니다.

   - Secure Shell(SSH) File Transfer 프로토콜(SFTP): SSH를 통한 파일 전송

SFTP에 대한 자세한 내용은 SFTP 지원 서버 생성를 참조하세요.

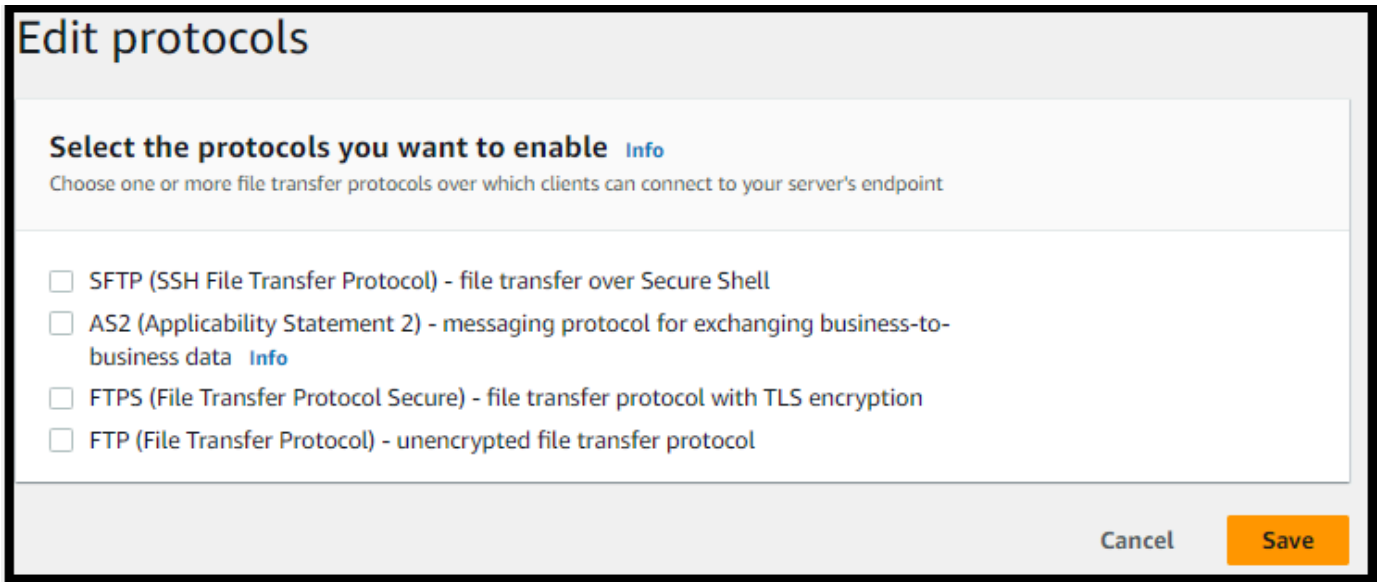
- FTPS(File Transfer 프로토콜 보안) - TLS 암호화를 사용한 파일 전송

  FTP에 대한 자세한 내용은 FTPS 지원 서버 생성를 참조하세요.

- FTP(File Transfer 프로토콜) - 암호화되지 않은 파일 전송

  FTPS에 대한 자세한 내용은 FTP 지원 서버 생성를 참조하세요.

> **Note**
>
> 기존 서버에서 SFTP만 사용하도록 설정한 상태에서 FTPS 및 FTP를 추가하려는 경우 FTPS 및 FTP와 호환되는 올바른 ID 제공자 및 엔드포인트 타입 설정이 있는지 확인해야 합니다.



FTPS를 선택하는 경우 클라이언트가 FTPS를 통해 서버에 연결할 때 서버를 식별하는 데 사용할 인증서 저장 AWS Certificate Manager (ACM) 를 선택해야 합니다.

새 퍼블릭 인증서를 요청하려면 AWS Certificate Manager 사용 설명서의 퍼블릭 인증서 요청을 참조하세요.

기존 인증서를 ACM으로 가져오려면 AWS Certificate Manager 사용 설명서의 ACM으로 인증서 가져오기를 참조하세요.

프라이빗 IP 주소를 통해 FTPS를 사용하도록 프라이빗 인증서를 요청하려면 AWS Certificate Manager 사용 설명서의 프라이빗 인증서 요청을 참조하세요.

다음 암호화 알고리즘 및 키 크기를 사용하는 인증서가 지원됩니다:

- 2048비트 RSA(RSA_2048)
- 4096비트 RSA(RSA_4096)
- 타원 프라임 곡선 256비트(EC_prime256v1)
- 타원 프라임 곡선 384 비트(EC_secp384r1)
- 타원 프라임 곡선 521비트(EC_secp521r1)

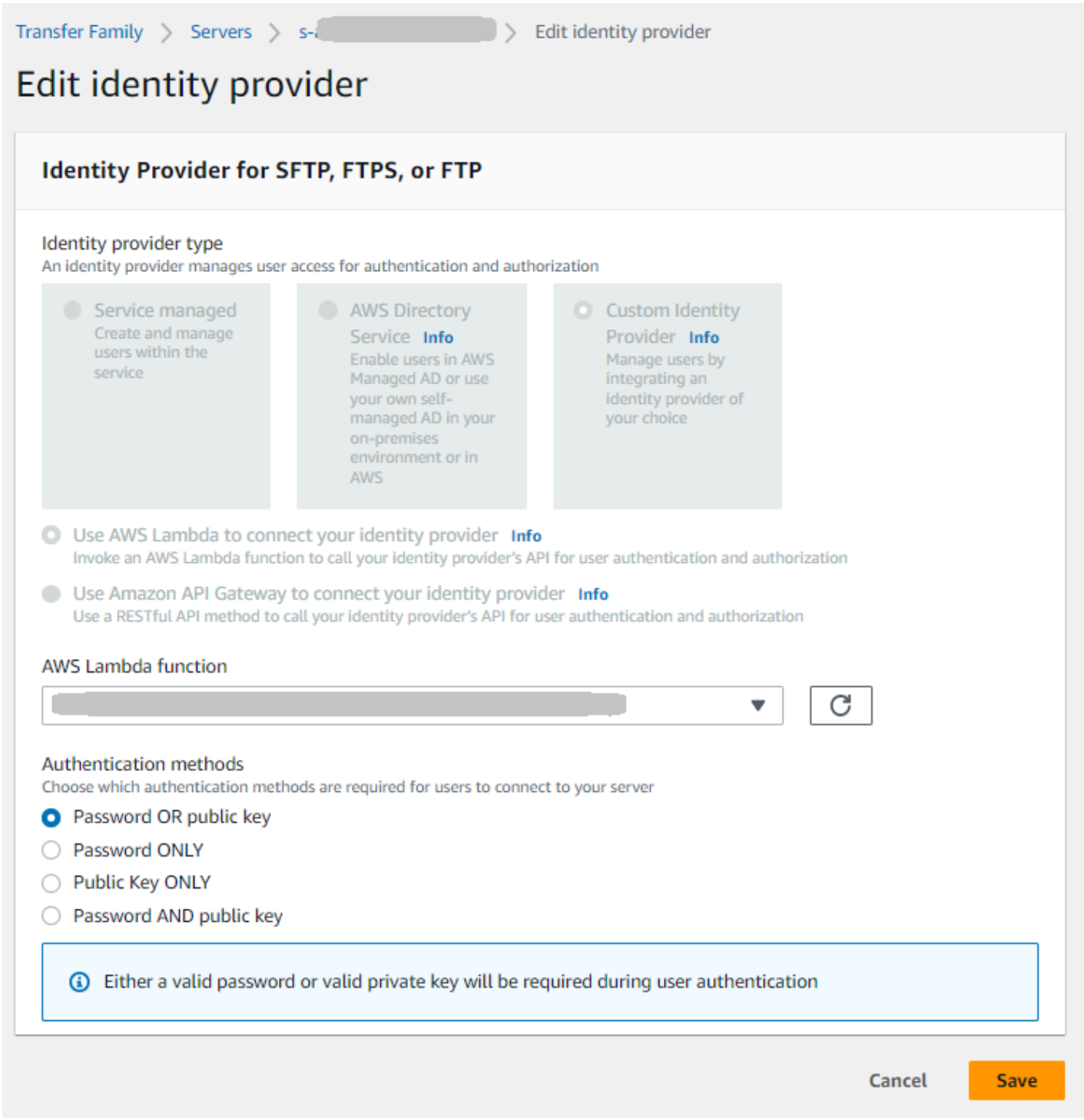
> **Note**
>
> 인증서는 FQDN 또는 IP 주소가 지정된 유효한 SSL/TLS X.509 버전 3 인증서여야 하며 발급자에 대한 정보가 포함되어 있어야 합니다.

3. 저장을 선택합니다. 서버 세부 정보 페이지로 돌아갑니다.

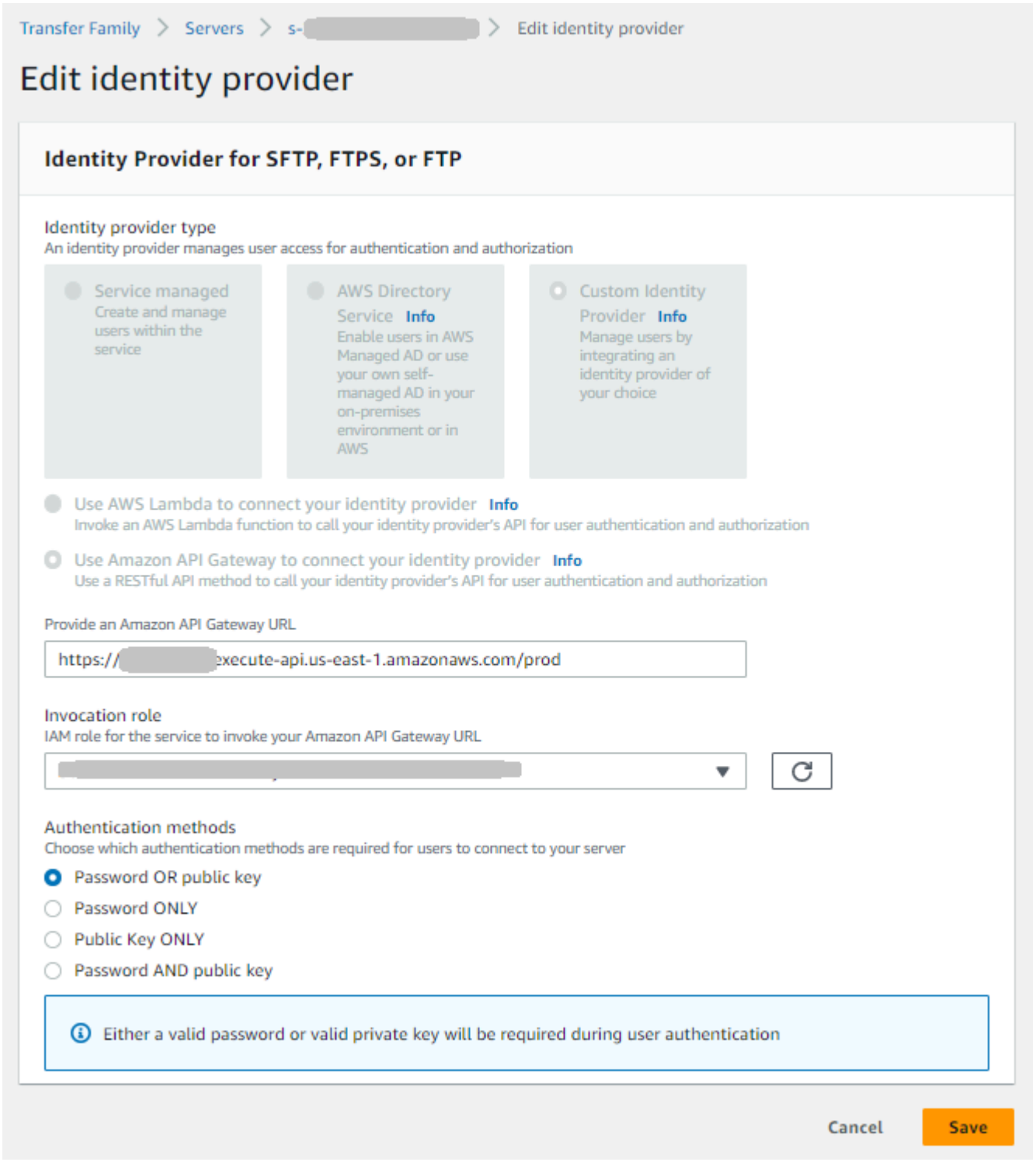
사용자 지정 자격 증명 제공자 파라미터 편집

AWS Transfer Family 콘솔에서 사용자 지정 ID 공급자의 경우 Lambda 함수를 사용하는지 또는 API Gateway를 사용하는지에 따라 일부 설정을 변경할 수 있습니다. 어느 경우든 서버가 SFTP 프로토콜을 사용하는 경우 인증 방법을 편집할 수 있습니다.

- Lambda를 자격 증명 제공자로 사용하는 경우 기본 Lambda 함수를 변경할 수 있습니다.



- API Gateway를 자격 증명 제공자로 사용하는 경우 게이트웨이 URL이나 호출 역할 또는 둘 다를 업데이트할 수 있습니다.



서버 엔드포인트 편집

AWS Transfer Family 콘솔에서 서버 엔드포인트 유형과 사용자 지정 호스트 이름을 수정할 수 있습니다. 또한 VPC 엔드포인트의 경우 가용 영역 정보를 편집할 수 있습니다.

서버 엔드포인트 세부 정보를 편집하려면

1. 서버 세부 정보 페이지에서 엔드포인트 세부 정보 옆의 편집을 선택합니다.
2. 엔드포인트 유형을 편집하려면 먼저 서버를 중지해야 합니다. 그런 다음 엔드포인트 구성 편집 페이지에서 엔드포인트 유형에 대해 다음 값 중 하나를 선택할 수 있습니다.

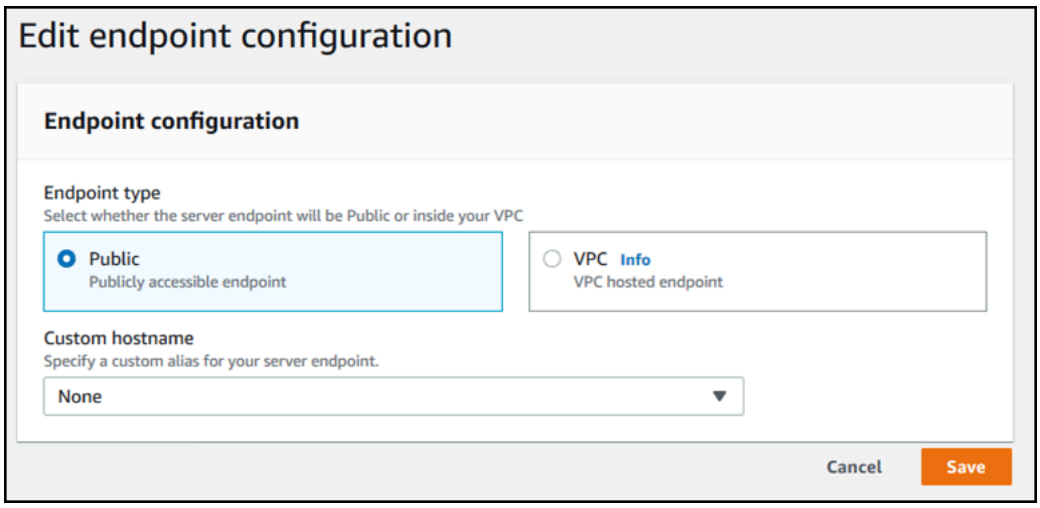
   - 퍼블릭 - 이 옵션을 사용하면 인터넷을 통해 서버에 액세스할 수 있습니다.
   - VPC - 이 옵션을 사용하면 VPC(VPC) 의 서버에 액세스할 수 있습니다. VPC에 대한 자세한 내용은 Virtual Private Cloud(VPC)에 서버 생성를 참조하세요.
3. 사용자 지정 호스트 이름의 경우 다음 중 하나를 선택합니다.

   - 없음 - 사용자 지정 도메인을 사용하지 않으려면 없음을 선택합니다.

     에서 제공한 AWS Transfer Family서버 호스트 이름을 가져옵니다. 서버 호스트 이름은 *`serverId`*`.server.transfer.`*`regionId`*`.amazonaws.com` 형식을 취합니다.
   - Amazon Route 53 DNS 별칭 - Route 53에서 자동으로 생성된 DNS 별칭을 사용하려면 이 옵션을 선택합니다.
   - 기타 DNS - 외부 DNS 서비스에서 이미 소유하고 있는 호스트 이름을 사용하려면 기타 DNS를 선택합니다.

   Amazon Route 53 DNS 별칭 또는 기타 DNS를 선택하면 서버의 엔드포인트와 연결할 이름 확인 방법이 지정됩니다.

   예를 들어 사용자 지정 도메인이 `sftp.inbox.example.com`일 수 있습니다. 사용자 지정 호스트 이름은 사용자가 입력했고 DNS 서비스가 해석할 수 있는 DNS 이름을 사용합니다. Route 53을 DNS 해석기로 사용하거나, 자체 DNS 서비스 공급자를 사용할 수 있습니다. AWS Transfer Family 가 Route 53을 이용해 트래픽을 사용자 지정 도메인에서 SFTP 엔드포인트로 라우팅하는 방법은 사용자 지정 호스트 이름으로 작업를 참조하세요.



4. VPC 엔드포인트의 경우 가용 영역 창에서 정보를 변경할 수 있습니다.
5. 저장을 선택합니다. 서버 세부 정보 페이지로 돌아갑니다.

로깅 구성 편집

AWS Transfer Family 콘솔에서 로깅 구성을 변경할 수 있습니다.

> **Note**
>
> 서버를 생성할 때 Transfer Family에서 자동으로 CloudWatch 로깅 IAM 역할을 생성한 경우 IAM 역할이 호출됩니다. `AWSTransferLoggingAccess` 모든 Transfer Family 서버에 사용할 수 있습니다.

로깅 구성 편집

1. 서버 세부 정보 페이지에서 추가 세부 정보 옆의 편집을 선택합니다.
2. 구성에 따라 로깅 역할, 정형 JSON 로깅 또는 둘 다 선택할 수 있습니다. 자세한 내용은 서버의 로깅 업데이트를 참조하세요.

보안 정책 편집

이 절차에서는 AWS Transfer Family 콘솔 또는 를 사용하여 Transfer Family 서버의 보안 정책을 변경하는 방법에 대해 설명합니다 AWS CLI.

> **Note**
>
> 엔드포인트가 FIPS를 지원하는 경우 FIPS 보안 정책을 비 FIPS 보안 정책으로 변경할 수 없습니다.

Console

콘솔을 사용하여 보안 정책을 편집하려면

1. 서버 세부 정보 페이지에서 추가 세부 정보 옆의 편집을 선택합니다.
2. 암호화 알고리즘 옵션의 경우 서버에서 사용하도록 설정된 암호화 알고리즘이 포함된 보안 정책을 선택합니다.

   보안 정책에 대한 자세한 설명은 섹션을 참조하세요서버 보안 정책 AWS Transfer Family.
3. 저장을 선택합니다.

   업데이트된 보안 정책을 볼 수 있는 서버 세부 정보 페이지로 돌아옵니다.

AWS CLI

CLI를 사용하여 보안 정책을 편집하려면

1. 다음 명령을 실행하여 서버에 연결된 현재 보안 정책을 확인합니다.

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

   이 `describe-server` 명령은 다음 줄을 포함하여 서버의 모든 세부 정보를 반환합니다.

   ```
   "SecurityPolicyName": "TransferSecurityPolicy-2018-11"
   ```

   이 경우 서버의 보안 정책은 입니다`TransferSecurityPolicy-2018-11`.
2. 명령에 보안 정책의 정확한 이름을 제공해야 합니다. 예를 들어 다음 명령을 실행하여 서버를 `TransferSecurityPolicy-2023-05` 업데이트합니다.

   ```
   aws transfer update-server --server-id your-server-id --security-policy-name
    "TransferSecurityPolicy-2023-05"
   ```

> **Note**
>
> 사용 가능한 보안 정책의 이름은 에 나열되어 서버 보안 정책 AWS Transfer Family 있습니다.

성공하면 명령은 다음 코드를 반환하고 서버의 보안 정책을 업데이트합니다.

```
{
    "ServerId": "your-server-id"
}
```

서버의 관리형 워크플로 변경

AWS Transfer Family 콘솔에서 서버와 관련된 관리형 워크플로를 변경할 수 있습니다.

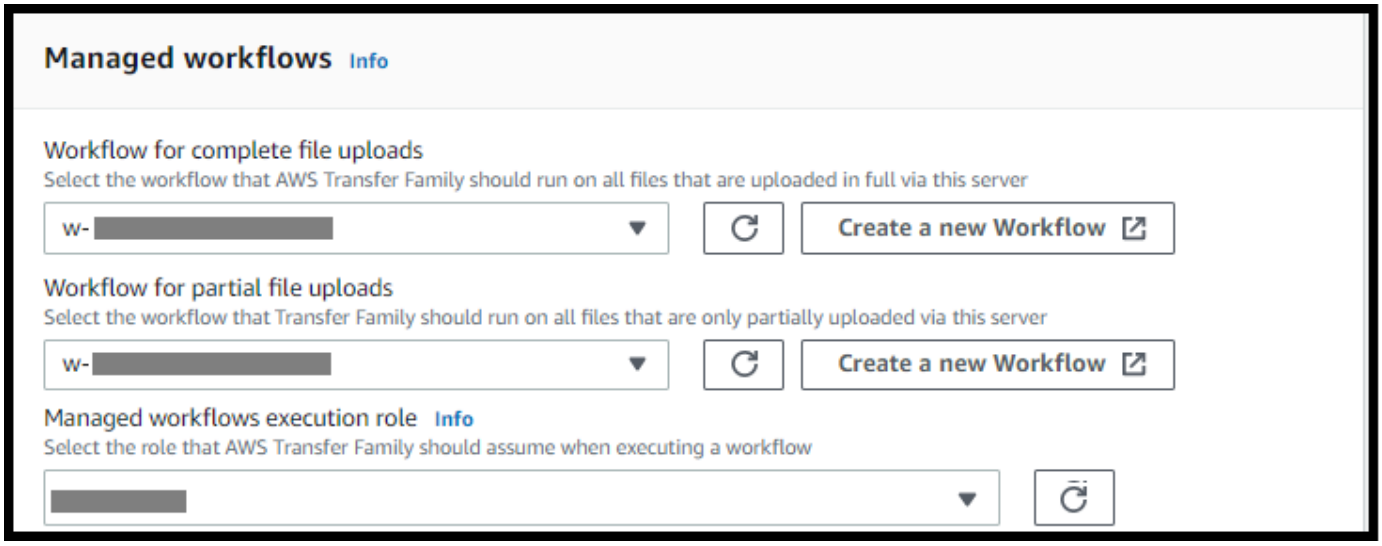
관리형 워크플로 변경

1. 서버 세부 정보 페이지에서 추가 세부 정보 옆의 편집을 선택합니다.
2. 추가 세부 정보 편집 페이지의 관리형 워크플로 섹션에서 모든 업로드에서 실행할 워크플로를 선택합니다.

   > **Note**
   >
   > 워크플로가 아직 없는 경우 새 워크플로 생성을 선택하여 워크플로를 생성합니다.

   a. 사용할 워크플로 ID를 선택합니다.

   b. 실행 역할을 선택합니다. 이 역할은 워크플로의 단계를 실행할 때 Transfer Family가 맡는 역할입니다. 자세한 내용은 워크플로에 대한 IAM 정책를 참조하세요. 저장(Save)을 선택합니다.



3. 저장을 선택합니다. 서버 세부 정보 페이지로 돌아갑니다.

서버의 디스플레이 배너 변경

AWS Transfer Family 콘솔에서 서버와 관련된 디스플레이 배너를 변경할 수 있습니다.

디스플레이 배너를 변경하려면

1. 서버 세부 정보 페이지에서 추가 세부 정보 옆의 편집을 선택합니다.
2. 추가 세부정보 편집 페이지의 디스플레이 배너 섹션에서 사용 가능한 디스플레이 배너의 텍스트를 입력합니다.
3. 저장을 선택합니다. 서버 세부 정보 페이지로 돌아갑니다.

서버를 온라인이나 오프라인으로 전환

AWS Transfer Family 콘솔에서 서버를 온라인 상태로 전환하거나 오프라인 상태로 전환할 수 있습니다.

서버를 온라인 상태로 전환하려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 탐색 창에서 서버를 선택합니다.
3. 오프라인 상태인 서버의 확인란을 선택합니다.

4. 작업에서 시작을 선택합니다.

서버가 오프라인에서 온라인으로 전환되는 데는 몇 분 정도 걸릴 수 있습니다.

> **Note**
>
> 오프라인으로 전환하기 위해 서버를 중단해도, 해당 서버에 대한 서비스 요금은 계속 발생합니다. 추가 서버 관련 요금을 방지하려면, 해당 서버를 삭제해야 합니다.

서버를 오프라인 상태로 전환하려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 탐색 창에서 서버를 선택합니다.
3. 온라인 상태인 서버의 확인란을 선택합니다.
4. 작업에서 중지를 선택합니다.

서버가 시작 또는 종료 중일 때는 서버가 파일 작업을 수행할 수 없습니다. 콘솔은 시작 및 중단 상태를 표시하지 않습니다.

오류 상태를 발견한 `STOP_FAILED` 경우 `START_FAILED` 또는 AWS Support 문의하여 문제를 해결하십시오.

SFTP 지원 서버의 호스트 키 관리

> **Important**
>
> 기존 사용자를 기존 SFTP 지원 서버에서 새 SFTP 지원 서버로 마이그레이션할 계획이 아니라면 이 섹션을 무시하세요.
> 실수로 서버의 호스트 키를 변경하면 작업에 영향을 줄 수 있습니다. SFTP 클라이언트의 구성 방식에 따라 신뢰할 수 있는 호스트 키가 없다는 메시지와 함께 SFTP 클라이언트가 즉시 실패하거나 위협적인 메시지가 표시될 수 있습니다. 연결을 자동화하는 스크립트가 있는 경우 이 스크립트도 실패할 가능성이 높습니다.

기본적으로 SFTP 지원 서버의 호스트 키를 AWS Transfer Family 제공합니다. 기본 호스트 키를 다른 서버의 호스트 키로 바꿀 수 있습니다. 기존 사용자를 기존 SFTP 지원 서버에서 새 SFTP 지원 서버로 이동하려는 경우에만 이 작업을 수행하세요.

사용자가 SFTP 지원 서버의 신뢰성 확인 메시지를 받지 않으려면 온프레미스 서버의 호스트 키를 SFTP 지원 서버로 가져옵니다. 또한 이렇게 하면 사용자에게 잠재적 공격에 대한 경고가 표시되지 않습니다. man-in-the-middle

추가 보안 조치로 호스트 키를 주기적으로 교체할 수도 있습니다.

> **Note**
>
> Transfer Family 콘솔에서 모든 서버의 서버 호스트 키를 지정하고 추가할 수 있지만 이러한 키는 SFTP 프로토콜을 사용하는 서버에만 유용합니다.

주제

추가 서버 호스트 키 추가

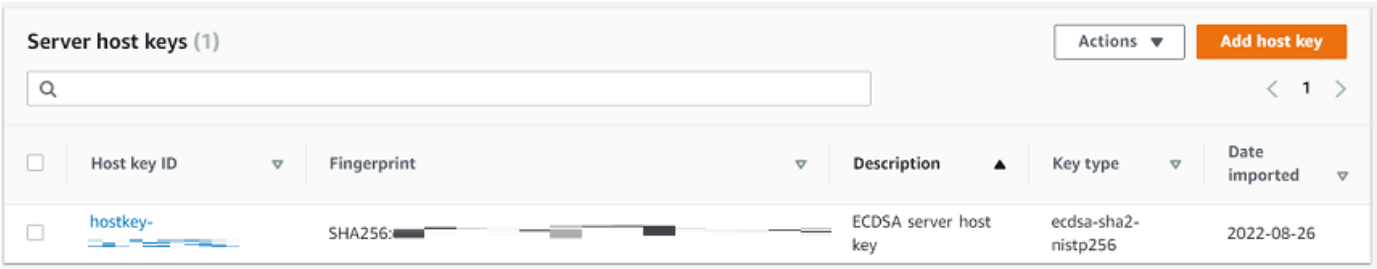
AWS Transfer Family 콘솔에서 서버 호스트 키를 더 추가할 수 있습니다. 형식이 다른 호스트 키를 추가하면 클라이언트가 서버에 연결할 때 서버를 식별하고 보안 프로필을 개선하는 데 유용할 수 있습니다. 예를 들어 원래 키가 RSA 키인 경우 ECDSA 키를 추가할 수 있습니다.

> **Note**
>
> SFTP 클라이언트는 활성 서버 키 중 하나와 일치할 수 있는 첫 번째 퍼블릭 키를 사용하여 연결합니다.

추가 서버 호스트 키 추가

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.

2. 왼쪽 탐색 창에서 서버를 선택한 다음 SFTP 프로토콜을 사용하는 서버를 선택합니다.
3. 서버 세부 정보 페이지에서 서버 호스트 키 섹션까지 아래로 스크롤합니다.



4. 호스트 키 추가를 선택합니다.

   서버 호스트 키 추가 페이지가 표시됩니다.

5. (옵션) 서버 호스트 키로는 클라이언트가 SFTP 지원 서버를 통해 서버에 연결할 때 서버를 식별하는 데 사용할 RSA, ED25519 또는 ECDSA 프라이빗 키를 입력합니다.

   > **Note**
   >
   > 서버 호스트 키를 생성할 때는 반드시 `-N ""`를 지정해야 합니다(암호 없음). 키 쌍을 생성하는 방법에 대한 자세한 내용은 macOS, Linux 또는 Unix에서 SSH 키 생성를 참조하세요.

6. (옵션) 설명을 추가하여 여러 서버 호스트 키를 구분합니다. 키에 태그를 추가할 수도 있습니다.
7. 키 추가를 선택합니다. 서버 세부 정보 페이지로 돌아갑니다.

AWS Command Line Interface (AWS CLI) 를 사용하여 호스트 키를 추가하려면 the section called “ImportHostKey” API 작업을 사용하고 새 호스트 키를 제공하십시오. 새 서버를 생성하는 경우 호스트 키를 the section called “CreateServer” API 작업의 파라미터로 제공합니다. AWS CLI 를 사용하여 기존 호스트 키의 설명을 업데이트할 수도 있습니다.

다음 예제 `import-host-key` AWS CLI 명령은 지정된 SFTP 지원 서버의 호스트 키를 가져옵니다.

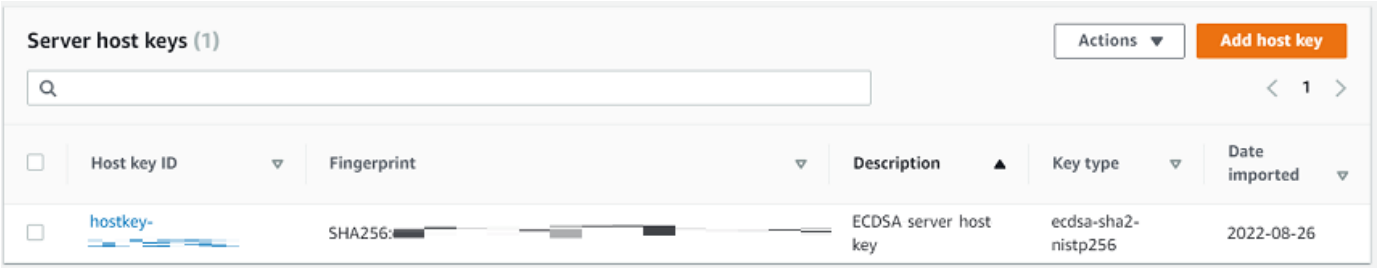
```
aws transfer import-host-key --description key-description --server-id your-server-id
 --host-key-body file://my-host-key
```

서버 호스트 키 삭제

AWS Transfer Family 콘솔에서 서버 호스트 키를 삭제할 수 있습니다.

서버 호스트 키를 삭제하려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 서버를 선택한 다음 SFTP 프로토콜을 사용하는 서버를 선택합니다.
3. 서버 세부 정보 페이지에서 서버 호스트 키 섹션까지 아래로 스크롤합니다.



4. 서버 호스트 키 섹션에서 키를 선택한 다음 작업에서 삭제를 선택합니다.
5. 표시되는 확인 대화 상자에 단어 **delete**을 입력한 다음 삭제를 선택한 다음 삭제를 선택합니다.

서버 페이지에서 호스트 키가 삭제됩니다.

를 사용하여 호스트 키를 삭제하려면 the section called “DeleteHostKey” API 작업을 사용하고 서버 ID와 호스트 키 ID를 제공하십시오. AWS CLI

다음 예제 `delete-host-key` AWS CLI 명령은 지정된 SFTP 지원 서버의 호스트 키를 삭제합니다.

```
aws transfer delete-host-key --server-id your-server-id --host-key-id your-host-key-id
```

서버 호스트 키 교체

서버 호스트 키를 주기적으로 교체할 수 있습니다.

클라이언트가 서버 호스트 키를 선택하는 방법

Transfer Family가 적용할 서버 키를 선택하는 방법은 여기에 설명된 대로 SFTP 클라이언트의 조건에 따라 달라집니다. 이전 키와 새 키가 각각 하나씩 있다고 가정합니다.

- SFTP 클라이언트에는 서버의 이전 공개 호스트 키가 없습니다. 클라이언트가 서버에 처음 연결되면 다음 중 하나가 발생합니다.
  - 클라이언트는 연결에 실패합니다 (그렇게 하도록 구성된 경우).
  - 또는 클라이언트가 사용 가능한 알고리즘과 일치하는 첫 번째 키를 선택하고 해당 키를 신뢰할 수 있는지 사용자에게 묻습니다. 이 경우 클라이언트는 `known_hosts` 파일 (또는 클라이언트가 신뢰

결정을 기록하는 데 사용하는 로컬 구성 파일 또는 리소스) 을 자동 업데이트하고 해당 키를 입력합니다.

- SFTP 클라이언트의 파일에는 이전 키가 있습니다. known_hosts 클라이언트는 새 키가 있더라도 이 키의 알고리즘이나 다른 알고리즘에 이 키를 사용하는 것을 선호합니다. 이는 클라이언트가 파일에 있는 키에 대해 더 높은 수준의 신뢰를 갖기 때문입니다. known_hosts
- SFTP 클라이언트의 키 파일에는 사용 가능한 모든 알고리즘의 새 known_hosts 키가 있습니다. 클라이언트는 신뢰할 수 없기 때문에 이전 키를 무시하고 새 키를 사용합니다.
- SFTP 클라이언트의 파일에는 두 키가 모두 있습니다. known_hosts 클라이언트는 서버에서 제공하는 사용 가능한 키 목록과 일치하는 첫 번째 키를 인덱스별로 선택합니다.

Transfer Family는 SFTP 클라이언트가 known_hosts 파일에 모든 키를 포함하는 것을 선호합니다. 이렇게 하면 Transfer Family 서버에 연결할 때 가장 유연하게 연결할 수 있기 때문입니다. 키 교체는 동일한 Transfer Family 서버의 known_hosts 파일에 여러 항목이 존재할 수 있다는 사실을 기반으로 합니다.

서버 호스트 키 교체 절차

예를 들어 Transfer Family 서버에 다음과 같은 서버 호스트 키 세트를 추가했다고 가정해 보겠습니다.

서버 호스트 키

호스트 키 타입	서버에 추가된 날짜
RSA	2020년 4월 1일
ECDSA	2020년 2월 1일
ED25519	2019년 12월 1일
RSA	2019년 10월 1일
ECDSA	2019년 6월 1일
ED25519	2019년 3월 1일

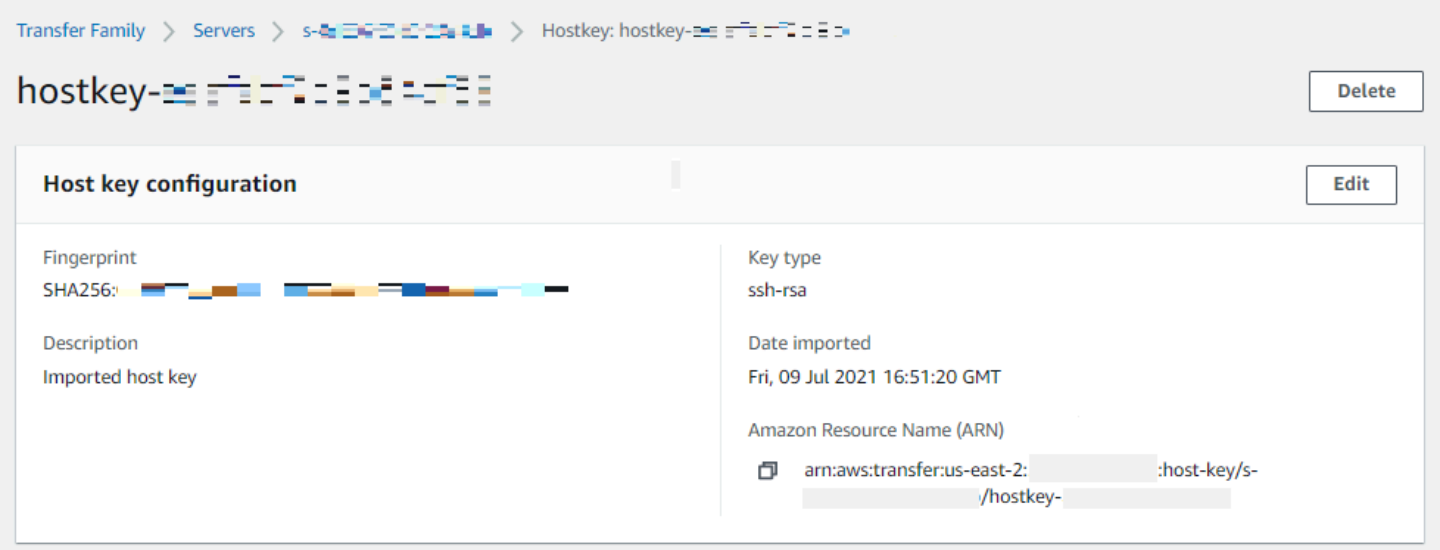
서버 호스트 키를 교체하려면

1. 새 서버 호스트 키를 추가합니다. 이 절차는 추가 서버 호스트 키 추가에 설명되어 있습니다.

2. 이전에 추가한 것과 같은 타입의 호스트 키를 하나 이상 삭제합니다. 이 절차는 서버 호스트 키 삭제에 설명되어 있습니다.
3. 앞에서 설명한 동작에 따라 모든 키가 표시되고 활성화될 수 클라이언트가 서버 호스트 키를 선택하는 방법 있습니다.

추가 서버 호스트 키 정보

호스트 키를 선택하여 해당 키에 대한 세부 정보를 표시할 수 있습니다.



서버 세부 정보 화면의 작업 메뉴에서 호스트 키를 삭제하거나 설명을 편집할 수 있습니다. 호스트 키를 선택한 다음 메뉴에서 적절한 작업을 선택합니다.

콘솔 내 사용량 모니터링

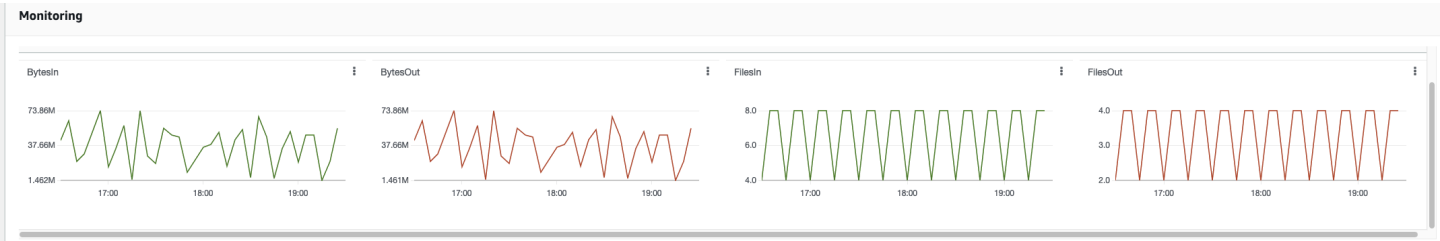
서버 세부 정보 페이지에서 서버 메트릭에 대한 정보를 얻을 수 있습니다. 이를 통해 파일 전송 워크로드를 한 곳에서 모니터링할 수 있습니다. 중앙 집중식 대시보드를 사용하여 파트너와 교환한 파일 수를

추적하고 파트너의 사용량을 면밀히 추적할 수 있습니다. 자세한 내용은 SFTP, FTPS 및 FTP 서버 세부 정보 보기를 참조하세요. 다음 표에서는 Transfer Family에 사용할 수 있는 측정치를 설명합니다.

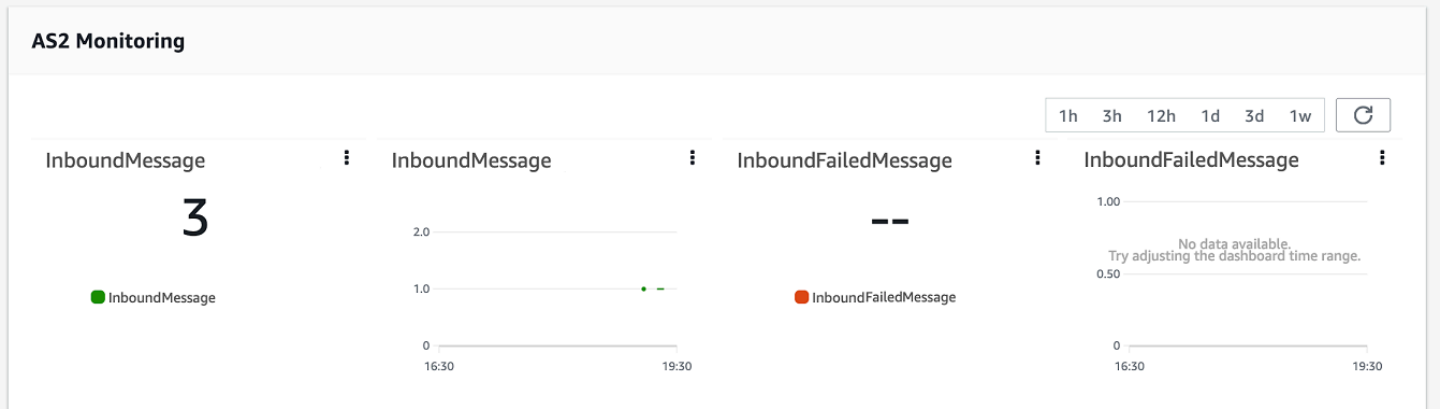
네임스페이스	지표	설명
AWS/Transfer	BytesIn	서버로 전송된 총 바이트 수. 단위: 개 기간: 5분
	BytesOut	서버에서 전송된 총 바이트 수입니다. 단위: 개 기간: 5분
	FilesIn	서버로 전송된 총 파일 수. AS2 프로토콜을 사용하는 서버의 경우 이 지표는 수신된 메시지 수를 나타냅니다. 단위: 개 기간: 5분
	FilesOut	서버에서 전송된 총 파일 수. 단위: 개 기간: 5분
	InboundMessage	거래 파트너로부터 성공적으로 수신한 AS2 메시지의 총 개수. 단위: 개 기간: 5분
	InboundFailedMessage	거래 파트너로부터 성공적으로 수신되지 못한 AS2 메시지의 총 개수. 즉, 거래 파트너가 메시지를 보냈지만

네임스페이스	지표	설명
		Transfer Family 서버가 메시지를 성공적으로 처리하지 못했습니다. 단위: 개 기간: 5분
	`OnUploadExecutionsStarted`	서버에서 시작된 워크플로를 실행한 총 횟수. 단위: 개 기간: 1분
	`OnUploadExecutionsSuccess`	서버에서 성공적인 워크플로를 실행한 총 횟수. 단위: 개 기간: 1분
	`OnUploadExecutionsFailed`	서버에서 실패한 워크플로를 실행한 총 횟수. 단위: 개 기간: 1분

모니터링 섹션에는 4개의 개별 그래프가 있습니다. 이 그래프는 입력 바이트, 출력 바이트, 파일 입력 및 출력 파일을 보여 줍니다.



AS2 프로토콜이 활성화된 서버의 경우 모니터링 정보 아래에 AS2 모니터링 섹션이 있습니다. 이 섹션에는 성공 및 실패 인바운드 메시지 수에 대한 세부 정보가 포함되어 있습니다.



선택한 그래프를 자체 창에서 열려면 확장 아이콘
(⤢)
을 선택합니다. 그래프의 세로 줄임표 아이콘
(⋮)
을 클릭하여 다음 항목이 포함된 드롭다운 메뉴를 열 수도 있습니다.

- 확대 - 선택한 그래프를 자체 창에서 엽니다.
- 새로 고침 - 그래프를 최신 데이터로 다시 로드합니다.
- 지표에서 보기 — Amazon에서 해당 지표 세부 정보를 엽니다 CloudWatch.
- 로그 보기 — 해당 로그 그룹을 엽니다 CloudWatch.

액세스 통제 관리

AWS Identity and Access Management (IAM) 정책을 사용하여 AWS Transfer Family 리소스에 대한 사용자의 액세스를 제어할 수 있습니다. IAM 정책은 명령문으로, 대부분 리소스에 대한 특정 수준의 액세스를 허용하는 JSON 형식을 취합니다. IAM 정책을 사용하여 SFTP 사용자가 수행하거나 수행하지 않게 할 파일 작업을 정의합니다. 또한 IAM 정책을 이용해 사용자에게 액세스를 허용할 S3 버킷 또는 일반 버킷을 정의할 수도 있습니다. 사용자에 대해 이러한 정책을 지정하려면, IAM 정책 및 이와 관련된 신뢰 관계가 있는 AWS Transfer Family 에 대한 IAM 역할을 만들어야 합니다.

각 사용자에게 IAM 역할이 할당됩니다. AWS Transfer Family 사용하는 IAM 역할 유형을 서비스 역할이라고 합니다. 사용자가 서버에 로그인하면 해당 사용자에게 매핑된 IAM 역할을 AWS Transfer Family 가정합니다. Amazon S3 버킷에 대한 사용자 액세스를 제공하는 IAM 역할을 생성하는 방법에 대해 알아보려면 IAM 사용 설명서의 AWS 서비스에 권한을 위임하는 역할 생성을 참조하십시오.

IAM 정책 내의 특정 권한을 사용하여 Amazon S3 객체에 대한 쓰기 전용 액세스 권한을 부여할 수 있습니다. 자세한 내용은 파일을 쓰고 나열할 수 있는 권한만 부여단원을 참조하세요.

AWS 스토리지 블로그에는 최소 권한 액세스를 설정하는 방법을 자세히 설명하는 게시물이 포함되어 있습니다. 자세한 내용은 워크플로우에서 최소 권한 액세스 구현을 참조하십시오. AWS Transfer Family

> **Note**
>
> Amazon S3 버킷을 AWS Key Management Service (AWS KMS) 사용하여 암호화한 경우 정책에 추가 권한을 지정해야 합니다. 자세한 내용은 Amazon S3의 데이터 암호화단원을 참조하세요. 또한 IAM 사용자 가이드에서 세션 정책에 대한 자세한 정보를 확인할 수 있습니다.

주제

- Amazon S3 버킷에 대한 읽기 및 쓰기 액세스 허용
- Amazon S3 버킷을 위한 세션 정책 생성
- 사용자가 S3 버킷에서 mkdir를 실행하지 못하도록 방지

Amazon S3 버킷에 대한 읽기 및 쓰기 액세스 허용

이 섹션에서는 특정 Amazon S3 버킷에 대한 읽기 및 쓰기 액세스를 허용하는 IAM 정책 생성 방법을 설명합니다. 이 IAM 정책이 있는 IAM 역할을 사용자에게 할당하면, 해당 사용자는 지정한 Amazon S3 버킷에 대한 읽기 및 쓰기 액세스 권한을 얻게 됩니다.

다음 정책은 Amazon S3 버킷에 대한 프로그래밍 방식의 읽기 및 쓰기 액세스 권한을 제공합니다. `GetObjectACL` 및 `PutObjectACL` 명령문은 크로스 계정 액세스를 활성화할 필요가 있는 경우에만 요구됩니다. 예컨대, Transfer Family 서버가 다른 계정의 버킷에 액세스할 필요가 있는 경우입니다.

```
{
  "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` 작업은 버킷 자체에 대한 권한을 요구합니다. `PUT`, `GET`, `DELETE` 작업은 객체 권한을 요구합니다. 이들은 각각 다른 리소스이며, 따라서 서로 다른 Amazon 리소스 이름(ARN)을 이용해 지정됩니다.

사용자의 액세스를 지정된 Amazon S3 버킷의 home 접두사로만 제약하는 방법은 Amazon S3 버킷을 위한 세션 정책 생성 섹션을 참조하세요.

Amazon S3 버킷을 위한 세션 정책 생성

세션 정책은 사용자를 Amazon S3 버킷의 특정 부분으로 제한하는 AWS Identity and Access Management (IAM) 정책입니다. 이 정책은 액세스를 실시간으로 평가해 이를 수행합니다.

> **Note**
>
> 세션 정책은 Amazon S3에서만 사용됩니다. Amazon EFS의 경우 POSIX 파일 권한을 사용하여 액세스를 제한합니다.

세션 정책은 사용자 그룹 전원이 Amazon S3 버킷의 특정 부분에만 액세스하도록 제한해야 할 때 사용합니다. 예를 들어 사용자 그룹이 home 디렉터리에만 액세스해야 할 수도 있습니다. 해당 사용자 그룹은 같은 IAM 역할을 공유합니다.

> **Note**
>
> 세션 정책의 최대 길이는 2,048자입니다. 자세한 내용은 API 참조의 `CreateUser` 작업에 대한 정책 요청 파라미터를 참조하세요.

세션 정책을 생성하려면, IAM 정책에서 다음 정책 변수를 사용하세요.

- `${transfer:HomeBucket}`
- `${transfer:HomeDirectory}`
- `${transfer:HomeFolder}`
- `${transfer:UserName}`

> **Important**
>
> 관리형 정책에서는 위의 변수를 사용할 수 없습니다. 또한 위의 변수를 IAM 역할 정의에서 정책 변수로 사용할 수도 없습니다. IAM 정책에 있는 이러한 변수는 사용자를 설정할 때 생성하고 바로 공급합니다. 그리고 이 세션 정책에서는 `${aws:Username}` 변수는 사용할 수 없습니

다. 이 변수는 AWS Transfer Family이(가) 요구하는 사용자 이름이 아닌 IAM 사용자 이름을 의미합니다.

다음 코드는 예 세션 정책을 보여줍니다.

```
{
  "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**
>
> 위의 정책 예에서는 사용자의 홈 디렉터리가 디렉터리임을 나타내기 위해 후행 슬래시를 포함하도록 설정된 것으로 가정합니다. 그 반면에 후행 슬래시가 없이 사용자의 HomeDirectory를 설정하는 경우에는 이를 정책의 일부로 포함해야 합니다.

이전 예 정책에서, `transfer:HomeFolder`, `transfer:HomeBucket`, `transfer:HomeDirectory` 정책 파라미터의 사용을 기억해 두세요. 이러한 파라미터는 및 에 HomeDirectory 설명된 대로 사용자를 위해 구성된 항목에 대해 설정됩니다. HomeDirectoryAPI Gateway 메서드 구현 이러한 파라미터의 정의는 다음과 같습니다.

- `transfer:HomeBucket` 파라미터는 HomeDirectory의 첫 번째 구성 요소로 대체됩니다.
- `transfer:HomeFolder` 파라미터가 HomeDirectory 파라미터의 나머지 부분으로 대체됩니다.
- `transfer:HomeDirectory` 파라미터는 Resource 명령문에서 S3 Amazon 리소스 이름 (ARN)의 일부로 사용할 수 있도록 선행 슬래시(/)가 제거되었습니다.

> **Note**
>
> 귀하가 논리적 디렉터리를 사용하는 경우—즉 사용자의 homeDirectoryType가 LOGICAL인 경우—이러한 정책 파라미터(HomeBucket, HomeDirectory, 및 HomeFolder)는 지원되지 않습니다.

예를 들어, Transfer Family 사용자에 대해 구성된 HomeDirectory 파라미터가 /home/bob/amazon/stuff/(이)라고 가정해 보겠습니다.

- `transfer:HomeBucket`를 /home(으)로 설정합니다.
- `transfer:HomeFolder`를 /bob/amazon/stuff/(으)로 설정합니다.
- `transfer:HomeDirectory`는 home/bob/amazon/stuff/가 됩니다.

첫 번째 "Sid"은(는) 사용자가 /home/bob/amazon/stuff/부터 시작하여 모든 디렉터리를 나열할 수 있게 해 줍니다.

두 번째 "Sid"은(는) 동일한 경로인 /home/bob/amazon/stuff/에 대한 사용자 put 및 get 액세스를 제한합니다.

앞의 정책을 활성화하면, 로그인 시 사용자는 자신의 홈 디렉터리에 있는 객체만 액세스할 수 있습니다. 연결 시 이러한 변수를 사용자에게 적합한 값으로 AWS Transfer Family 바꿉니다. 이렇게 하면 같은 정책 설명서를 다수의 사용자에게 쉽게 적용할 수 있습니다. 이 방법은 Amazon S3 버킷에 대한 사용자의 액세스 관리에 필요한 IAM 역할과 정책 관리 오버헤드를 줄입니다.

세션 정책을 사용하면 업무 요건에 따라 각 사용자의 액세스를 조절할 수도 있습니다. 자세한 내용은 IAM 사용 설명서의 권한 AssumeRole, AssumeRoleWith SAML을 참조하십시오. AssumeRoleWithWebIdentity

> **Note**
>
> AWS Transfer Family 정책의 Amazon 리소스 이름 (ARN) 대신 정책 JSON을 저장합니다. 따라서 IAM 콘솔에서 정책을 변경한 경우 AWS Transfer Family 콘솔로 돌아가서 최신 정책 콘텐츠로 사용자를 업데이트해야 합니다. 사용자 구성 섹션의 정책 정보 탭에서 사용자를 업데이트할 수 있습니다.
> 를 사용하는 경우 다음 명령을 사용하여 정책을 업데이트할 수 있습니다. AWS CLI
>
> ```
> aws transfer update-user --server-id server --user-name user --policy \
>    "$(aws iam get-policy-version --policy-arn policy --version-id version --
> output json)"
> ```

사용자가 S3 버킷에서 **mkdir**를 실행하지 못하도록 방지

Amazon S3 버킷에 디렉터리를 생성할 수 있는 모든 기능을 제한할 수 있습니다. 이렇게 하려면 s3:PutObject 작업을 허용하지만 "/"(선행 슬래시)로 끝나는 키도 거부하는 IAM 정책을 생성합니다. 다음 예 정책은 사용자가 Amazon S3 버킷에 파일을 업로드하도록 허용하지만 Amazon S3 버킷에서는 mkdir 명령을 거부합니다.

```
{
   "Sid":"DenyMkdir",
   "Action":[
      "s3:PutObject"
   ],
   "Effect":"Deny",
   "Resource":[
```

```
      "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*/",
      "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*/*"
    ]
}
```

> **Note**
>
> 두 번째 리소스 라인을 사용하면 사용자가 `put my-file DOC-EXAMPLE-BUCKET/new-folder/my-file`와(과) 같은 명령을 실행하여 하위 폴더를 생성할 수 없습니다.

AWS Transfer Family의 로깅

AWS Transfer FamilyAWS CloudTrail및 Amazon CloudWatch 모두와 통합됩니다. CloudTrail 서로 다르지만 상호 보완적인 용도로 CloudWatch 사용됩니다.

- CloudTrail 사용자 AWS 계정 내에서 수행된 작업을 기록하는 AWS 서비스입니다. 콘솔 로그인, AWS Command Line Interface 명령, SDK/API 호출과 같은 활동에 대한 API 호출을 지속적으로 모니터링하고 기록합니다. 이를 통해 누가, 언제, 어디서 어떤 조치를 취했는지 기록할 수 있습니다. CloudTrail 사용자 환경의 모든 활동에 대한 기록을 제공하여 감사, 액세스 관리 및 규정 준수에 도움이 AWS 됩니다. 자세한 내용은 AWS CloudTrail사용 설명서를 참조하십시오.
- CloudWatch AWS리소스 및 애플리케이션에 대한 모니터링 서비스입니다. 지표와 로그를 수집하여 리소스 사용률, 애플리케이션 성능 및 전체 시스템 상태에 대한 가시성을 제공합니다. CloudWatch 문제 해결, 경보 설정 및 자동 크기 조정과 같은 운영 작업을 지원합니다. 자세한 내용은 Amazon CloudWatch 사용 설명서를 참조하십시오.

주제

AWS CloudTrail로깅 대상 AWS Transfer Family

AWS Transfer Family에서 AWS Transfer Family 사용자AWS CloudTrail, 역할 또는 서비스가 수행한 작업의 기록을 제공하는 AWS 서비스와 통합됩니다. CloudTrail 모든 API 호출을 AWS Transfer Family 이벤트로 캡처합니다. 캡처되는 호출에는 AWS Transfer Family 콘솔로부터의 호출과 AWS Transfer Family API 작업에 대한 코드 호출이 포함됩니다.

트레일은 지정한 Amazon S3 버킷에 이벤트를 로그 파일로 전송할 수 있는 구성입니다. CloudTrail 로그 파일에는 하나 이상의 로그 항목이 포함되어 있습니다. 이벤트는 모든 소스의 단일 요청을 나타내며 요청된 작업, 작업 날짜 및 시간, 요청 매개 변수 등에 대한 정보를 포함합니다. CloudTrail 로그 파일은 공개 API 호출의 정렬된 스택 트레이스가 아니므로 특정 순서로 표시되지 않습니다.

AWS Transfer Family에 대한 이벤트를 포함하여 AWS 계정에 이벤트를 지속적으로 기록하려면 추적을 생성합니다. 트레일을 사용하면 CloudTrail Amazon S3 버킷에 로그 파일을 전송할 수 있습니다. 콘솔에서 추적을 생성하면 기본적으로 모든 AWS 지역에 추적이 적용됩니다. 추적은 AWS 파티션에 있는 모든 지역의 이벤트를 로깅하고 지정된 Amazon S3 버킷으로 로그 파일을 전송합니다. 또한

CloudTrail 로그에서 수집된 이벤트 데이터를 추가로 분석하고 이에 따라 조치를 취하도록 다른 AWS 서비스를 구성할 수 있습니다. 자세한 내용은 다음 자료를 참조하십시오.

- 추적 생성 개요
- CloudTrail 지원되는 서비스 및 통합
- 에 대한 Amazon SNS 알림 구성 CloudTrail
- 여러 지역에서 CloudTrail 로그 파일 수신 및 여러 계정으로부터 CloudTrail 로그 파일 수신

모든 AWS Transfer Family 작업은 에 의해 CloudTrail 기록되고 문서화됩니다. ActionsAPI reference 예를 들어`CreateServer`, 에 대한 호출 `ListUsers` 및 `StopServer` 작업은 CloudTrail 로그 파일에 항목을 생성합니다.

모든 이벤트 및 로그 항목에는 요청을 생성한 사용자에 대한 정보가 들어 있습니다. 보안 인증 정보를 이용하면 다음을 쉽게 판단할 수 있습니다.

- 요청을 루트로 했는지 아니면 AWS Identity and Access Management 사용자 보안 인증으로 했는지 여부.
- 역할 또는 페더레이션 사용자에 대한 임시 보안 인증을 사용하여 요청이 생성되었는지 여부.
- 다른 AWS 서비스에서 요청했는지.

자세한 내용은 CloudTrail userIdentity 요소를 참조하십시오.

트레일을 생성하면 에 대한 이벤트를 포함하여 Amazon S3 버킷으로 CloudTrail 이벤트를 지속적으로 전송할 수 AWS Transfer Family 있습니다. 트레일을 구성하지 않아도 CloudTrail 콘솔의 이벤트 기록에서 가장 최근 이벤트를 계속 볼 수 있습니다.

에서 수집한 CloudTrail 정보를 사용하여 요청을 받은 사람AWS Transfer Family, 요청한 IP 주소, 요청한 사람, 요청 시기 및 추가 세부 정보를 확인할 수 있습니다.

자세한 CloudTrail 내용은 AWS CloudTrail사용 설명서를 참조하십시오.

주제

- AWS CloudTrail 로깅 활성화
- 서버 생성을 위한 로그 항목 예제

AWS CloudTrail 로깅 활성화

AWS CloudTrail을(를) 사용하여 AWS Transfer Family API 호출을 모니터링할 수 있습니다. API 호출을 모니터링하면, 유용한 보안 및 운영 정보를 얻을 수 있습니다. Amazon S3 객체 수준 로깅을 활성화한 경우, 요청자 필드에 `RoleSessionName`이(가) `[AWS:Role Unique Identifier]/username.sessionid@server-id`(으)로 포함됩니다. AWS Identity and Access Management (IAM) 역할에 대한 자세한 설명은 AWS Identity and Access Management사용자 가이드의 고유 식별자를 참조하세요.

> **Important**
>
> `RoleSessionName`의 최대 길이는 64자입니다. `RoleSessionName` 길이가 더 길면 `server-id`이(가) 잘립니다.

서버 생성을 위한 로그 항목 예제

다음 예제는 작업을 보여주는 CloudTrail 로그 항목 (JSON 형식) 을 보여줍니다. `CreateServer`

```
{
    "eventVersion": "1.09",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AAAA4FFF5HHHHH6NNWWWW:user1",
        "arn": "arn:aws:sts::123456789102:assumed-role/Admin/user1",
        "accountId": "123456789102",
        "accessKeyId": "AAAA52C2WWWWW3BB4Z",
        "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"
}
```

아마존 CloudWatch 로깅 대상 AWS Transfer Family

Amazon은 실행 중인 AWS Transfer Family 리소스와 애플리케이션을 AWS 실시간으로 CloudWatch 모니터링합니다. 리소스와 애플리케이션에 대해 측정할 수 있는 변수인 지표를 수집하고 추적하는 데 사용할 CloudWatch 수 있습니다.

CloudWatch 홈 페이지에는 Transfer Family 및 사용하는 기타 모든 AWS 서비스에 대한 메트릭이 자동으로 표시됩니다. 사용자 지정 대시보드를 추가로 생성해 사용자 지정 애플리케이션에 대한 지표를 표시하고, 선택한 지표의 사용자 지정 집합을 표시할 수 있습니다.

지표를 감시해 알림을 보내거나 임계값을 위반한 경우 모니터링 중인 리소스를 자동으로 변경하는 경보를 생성할 수 있습니다. 예를 들어 Transfer Family 서버로 전송되는 파일을 모니터링하고 해당 데이터를 사용하여 증가된 부하를 처리하기 위해 추가 서버를 배포해야 하는지 여부를 결정할 수 있습니다. 또한 이 데이터를 사용하여 사용량이 적은 인스턴스를 중지하거나 삭제하여 비용을 절감할 수 있습니다.

Transfer Family의 CloudWatch 로깅 유형

Transfer Family는 CloudWatch 다음과 같은 두 가지 방법으로 이벤트를 기록할 수 있습니다.

- JSON 구조적 로깅
- 로깅 역할을 통한 로깅

Transfer Family 서버의 경우 원하는 로깅 메커니즘을 선택할 수 있습니다. 커넥터 및 워크플로의 경우 로깅 역할만 지원됩니다.

JSON 구조적 로깅

서버 이벤트를 로깅하려면 JSON 구조적 로깅을 사용하는 것이 좋습니다. 이는 CloudWatch 로그 쿼리를 가능하게 하는 보다 포괄적인 로깅 형식을 제공합니다. 이러한 유형의 로깅의 경우 서버를 생성 (또는 서버의 로깅 구성을 편집하는) 사용자에 대한 IAM 정책에는 다음 권한이 포함되어야 합니다.

- `logs:CreateLogDelivery`
- `logs:DeleteLogDelivery`
- `logs:DescribeLogGroups`
- `logs:DescribeResourcePolicies`
- `logs:GetLogDelivery`

- logs:ListLogDeliveries
- logs:PutResourcePolicy
- logs:UpdateLogDelivery

다음은 예제 정책입니다.

```
{
    "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/*"
        }
    ]
}
```

JSON 구조적 로깅 설정에 대한 자세한 내용은 을 참조하십시오. 서버의 로깅 생성, 업데이트 및 보기

로깅 역할

커넥터뿐 아니라 서버에 연결된 관리 워크플로우에 대한 이벤트를 기록하려면 로깅 역할을 지정해야 합니다. 액세스 권한을 설정하려면 리소스 기반 IAM 정책과 해당 액세스 정보를 제공하는 IAM 역할을 만듭니다. 다음은 서버 이벤트를 기록할 수 AWS 계정 있는 정책의 예입니다.

```
{
    "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/*"
        }
    ]
}
```

워크플로 이벤트를 기록하도록 로깅 역할을 구성하는 방법에 대한 자세한 내용은 을 참조하십시오워크플로의 로깅 관리.

주제

- 서버의 로깅 생성, 업데이트 및 보기
- 워크플로의 로깅 관리
- CloudWatch 로깅 역할을 구성합니다.
- Transfer Family 로그 스트림 보기
- 아마존 CloudWatch 알람 생성
- S3 액세스 로그에 Amazon S3 API 호출 로깅
- 혼동된 대리자 문제를 제한하는 예
- CloudWatch Transfer Family의 로그 구조
- 예제 CloudWatch 로그 항목
- Transfer Family에 대한 CloudWatch 측정항목 사용
- AWS 사용자 알림 와 함께 사용 AWS Transfer Family
- 쿼리를 사용하여 로그 항목을 필터링합니다.

서버의 로깅 생성, 업데이트 및 보기

모든 AWS Transfer Family 서버의 경우 두 가지 로깅 옵션, 즉 `LoggingRole` (서버에 연결된 워크플로를 로깅하는 데 사용) 또는 두 가지 로깅 옵션 중에서 선택할 수 `StructuredLogDestinations` 있습니다. `StructuredLogDestinations` 사용의 이점은 다음과 같습니다.

- 구조화된 JSON 형식으로 로그를 수신합니다.

- JSON 형식의 필드를 자동으로 검색하는 Amazon CloudWatch Logs Insights로 로그를 쿼리합니다.
- AWS Transfer Family 리소스 간에 로그 그룹을 공유하면 여러 서버의 로그 스트림을 단일 로그 그룹으로 결합하여 모니터링 구성 및 로그 보존 설정을 더 쉽게 관리할 수 있습니다.
- 대시보드에 추가할 수 있는 집계된 지표와 시각화를 만들 수 있습니다. CloudWatch
- 로그 그룹을 사용하여 통합된 로그 지표, 시각화 및 대시보드를 생성하여 사용량 및 성능 데이터를 추적할 수 있습니다.

`LoggingRole` 또는 `StructuredLogDestinations`에 대한 옵션은 개별적으로 구성 및 제어됩니다. 각 서버에 대해 하나 또는 두 가지 로깅 방법을 모두 설정하거나, 로깅이 전혀 없도록 서버를 구성할 수 있습니다(권장되지는 않음).

Transfer Family 콘솔을 사용하여 새 서버를 생성할 경우 기본값으로 로깅이 활성화됩니다. 서버를 생성한 후 `UpdateServer` API 직접 호출을 사용하여 로깅 구성을 변경할 수 있습니다. 자세한 내용은 대상을 참조하십시오. StructuredLog

현재 워크플로의 경우 로깅을 활성화하려면 로깅 역할을 지정해야 합니다.

- `CreateServer`또는 `UpdateServer` API 직접적 호출을 사용하여 워크플로를 서버에 연결하는 경우 시스템에서 자동으로 로깅 역할을 만들지 않습니다. 워크플로 이벤트를 기록하려면 서버에 로깅 역할을 명시적으로 연결해야 합니다.
- Transfer Family 콘솔을 사용하여 서버를 생성하고 워크플로를 연결하는 경우 이름에 서버 ID가 포함된 로그 그룹에 로그가 전송됩니다. 형식은 `/aws/transfer/`*`server-id`*이며, 예를 들면 `/aws/transfer/s-1111aaaa2222bbbb3`와 같습니다. 서버 로그는 동일한 로그 그룹이나 다른 로그 그룹으로 전송될 수 있습니다.

콘솔에서 서버를 만들고 편집할 때 고려할 로깅 고려 사항

- 콘솔을 통해 만든 새 서버는 워크플로가 서버에 연결되어 있지 않는 한 구조화된 JSON 로깅만 지원합니다.
- 콘솔에서 새로 만든 서버의 경우 로깅 없음을 선택할 수 없습니다.
- 기존 서버는 언제든지 콘솔을 통해 구조화된 JSON 로깅을 활성화할 수 있습니다.
- 콘솔을 통해 구조화된 JSON 로깅을 활성화하면 기존 로깅 방법이 비활성화되므로 고객에게 요금이 두 배로 청구되지 않습니다. 단, 워크플로가 서버에 연결된 경우는 예외입니다.
- 구조화된 JSON 로깅을 활성화하면 나중에 콘솔을 통해 비활성화할 수 없습니다.
- 구조화된 JSON 로깅을 활성화하면 언제든지 콘솔을 통해 로그 그룹 대상을 변경할 수 있습니다.

- 구조화된 JSON 로깅을 활성화한 경우 API를 통해 두 로깅 타입을 모두 활성화한 경우 콘솔을 통해 로깅 역할을 편집할 수 없습니다. 단, 워크플로가 서버에 연결된 경우는 예외입니다. 하지만 로깅 역할은 추가 세부 정보에 계속 표시됩니다.

API 또는 SDK를 사용하여 서버를 만들고 편집할 때 로깅 고려 사항

- API를 통해 새 서버를 만드는 경우 로깅 타입 중 하나 또는 두 가지를 모두 구성하거나 로깅 없음을 선택할 수 있습니다.
- 기존 서버의 경우 언제든지 구조화된 JSON 로깅을 활성화 및 비활성화할 수 있습니다.
- API를 통해 언제든지 로그 그룹을 변경할 수 있습니다.
- API를 통해 언제든지 로깅 역할을 변경할 수 있습니다.

구조적 로깅을 활성화하려면 다음 권한이 있는 계정에 로그인해야 합니다

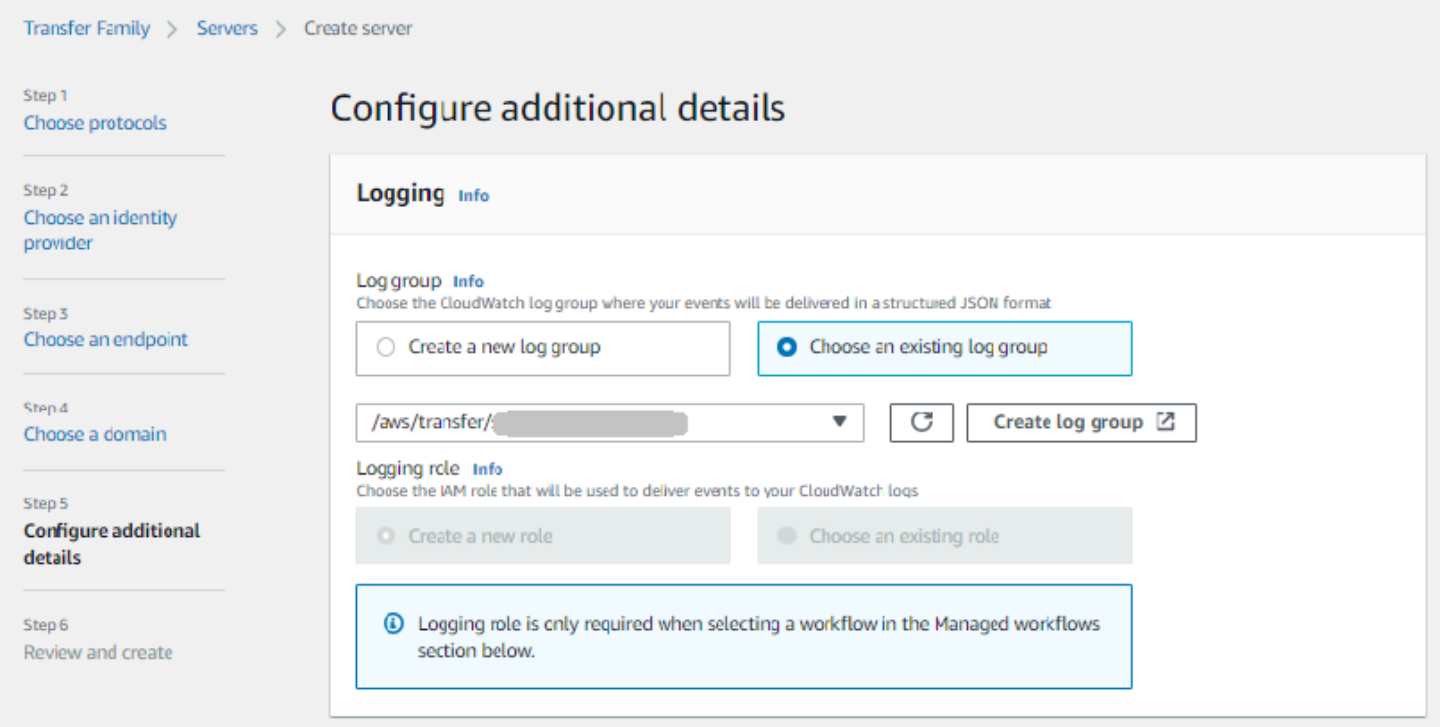
- `logs:CreateLogDelivery`
- `logs:DeleteLogDelivery`
- `logs:DescribeLogGroups`
- `logs:DescribeResourcePolicies`
- `logs:GetLogDelivery`
- `logs:ListLogDeliveries`
- `logs:PutResourcePolicy`
- `logs:UpdateLogDelivery`

정책 예는 섹션에서 확인할 수 CloudWatch 로깅 역할을 구성합니다. 있습니다.

주제

서버에 대한 로깅 생성

새 서버를 만들 때 추가 세부 정보 구성 페이지에서 기존 로그 그룹을 지정하거나 새 로그 그룹을 만들 수 있습니다.



로그 그룹 생성을 선택하면 CloudWatch 콘솔 (https://console.aws.amazon.com/cloudwatch/) 에서 로그 그룹 생성 페이지가 열립니다. 자세한 내용은 로그에 CloudWatch 로그 그룹 만들기를 참조하십시오.

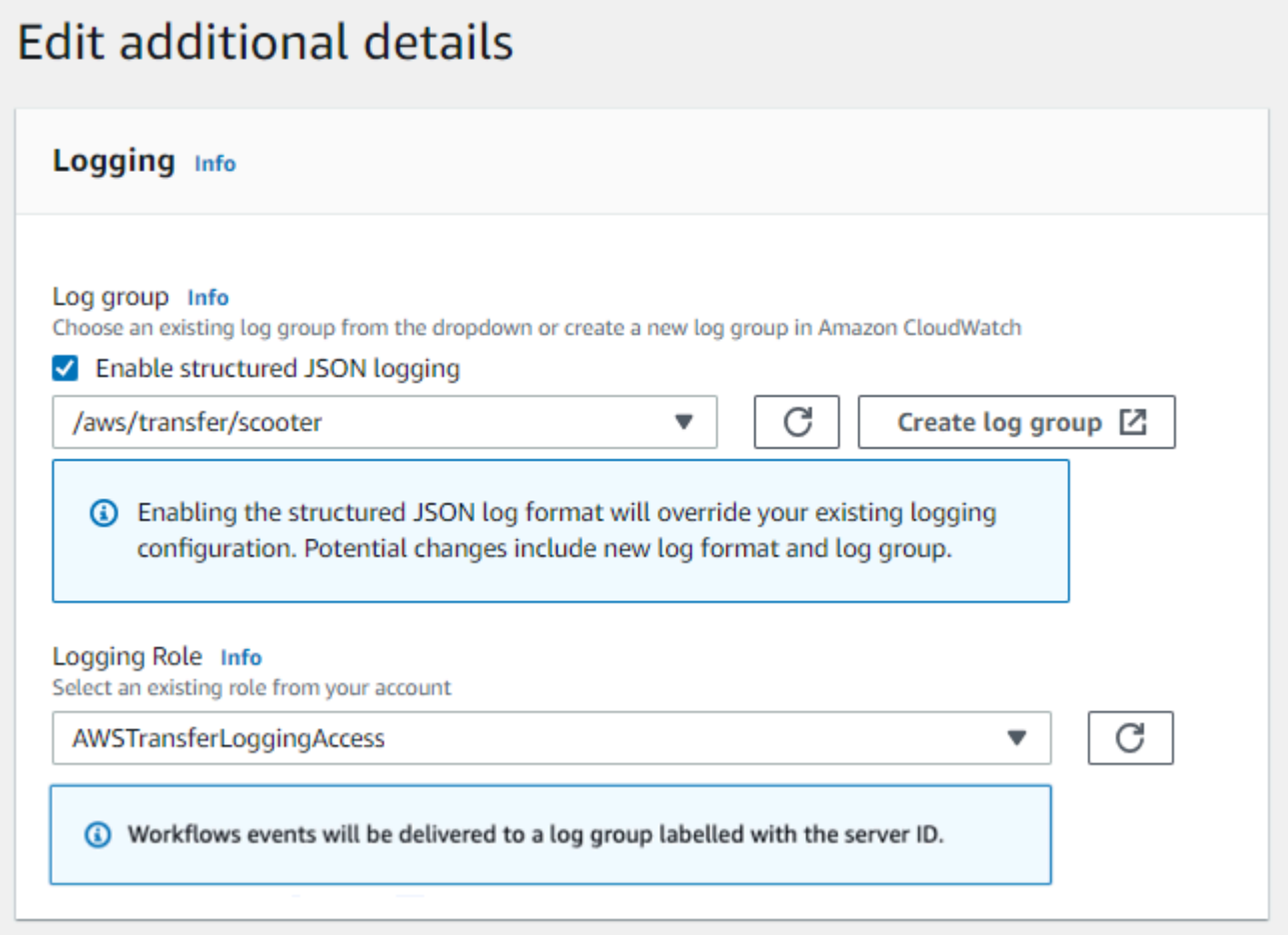
서버의 로깅 업데이트

로깅 세부 정보는 업데이트 시나리오에 따라 다릅니다.

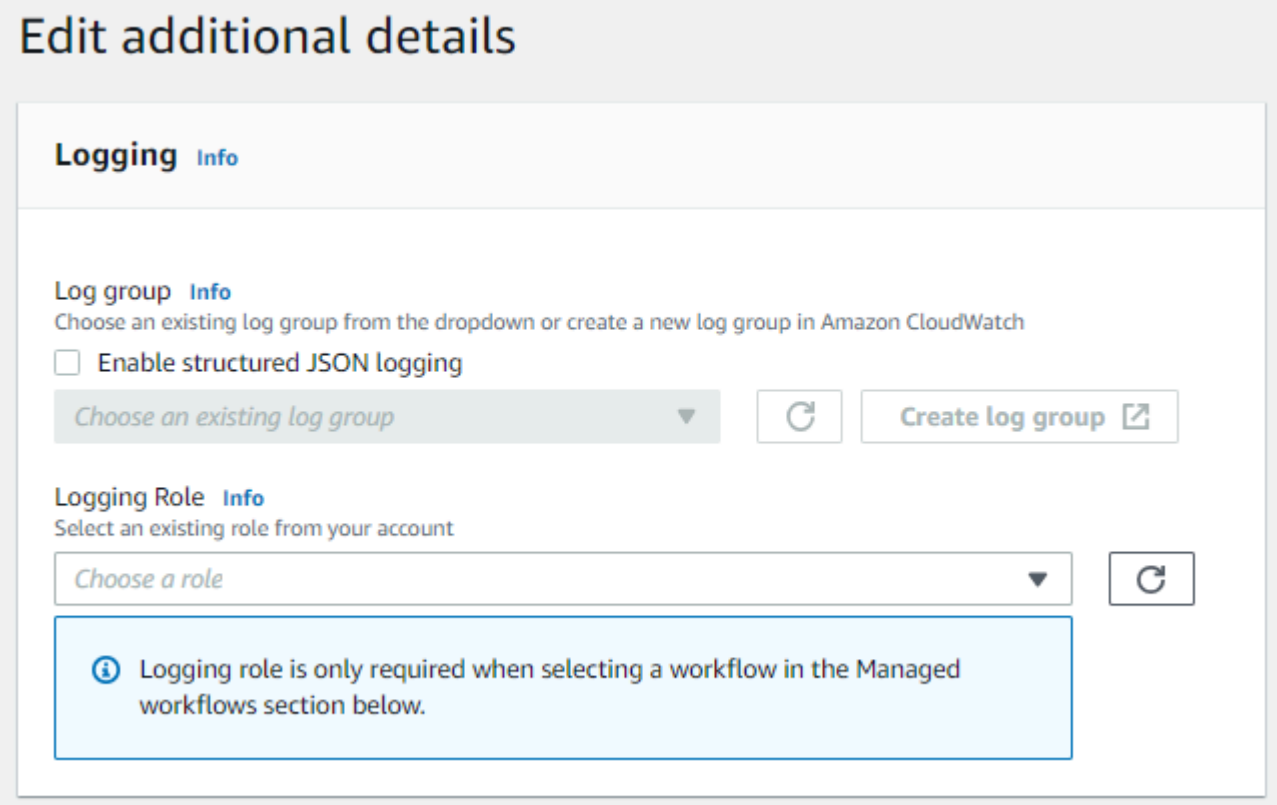
> **Note**
>
> 구조화된 JSON 로깅을 선택하면 드문 경우지만 Transfer Family에서 이전 형식의 로깅을 중지하지만 새 JSON 형식으로 로깅을 시작하는 데 시간이 걸리는 지연이 발생할 수 있습니다. 이로 인해 이벤트가 로그되지 않을 수 있습니다. 서비스 중단은 없겠지만 로깅 방법을 변경한 후 처음 1시간 동안은 로그가 삭제될 수 있으므로 파일 전송에 주의해야 합니다.

기존 서버를 편집하는 경우 옵션은 서버 상태에 따라 달라집니다.

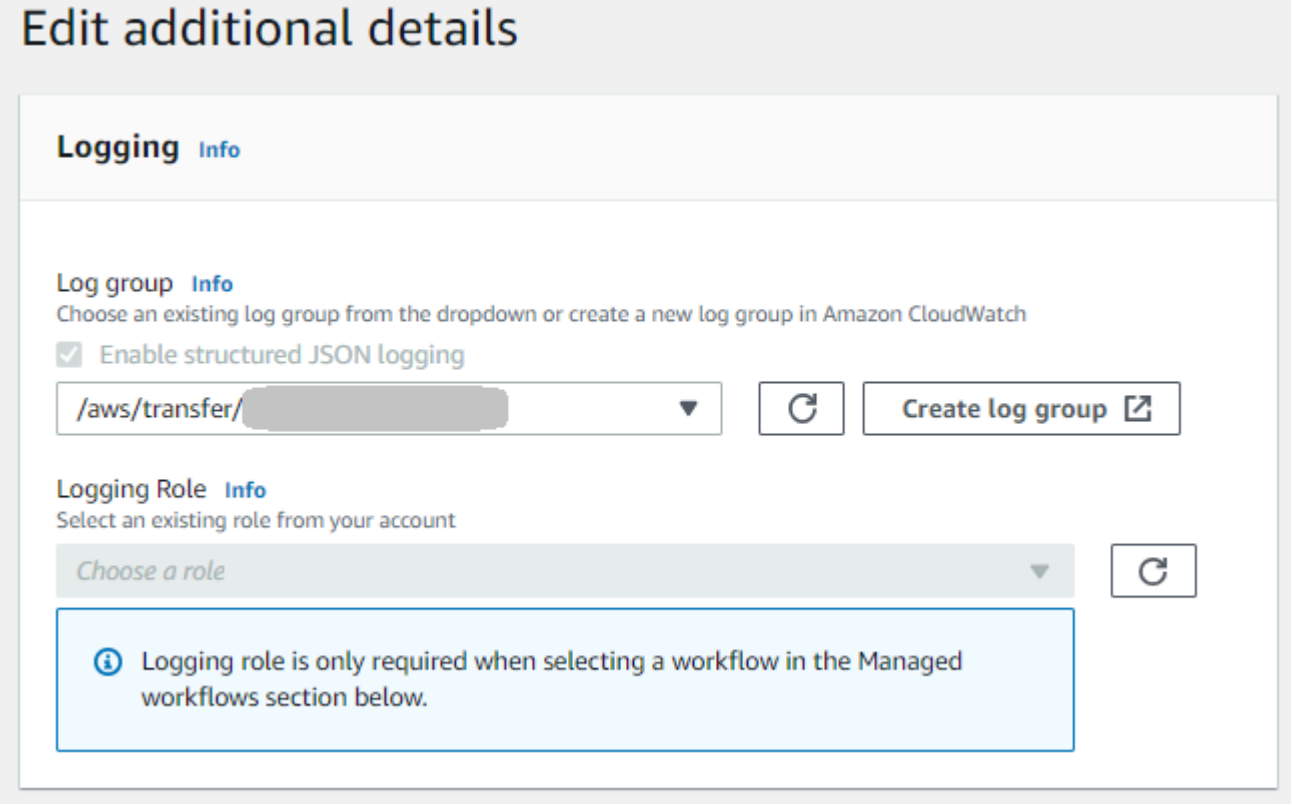
- 서버에는 이미 로깅 역할이 활성화되어 있지만 구조화된 JSON 로깅은 활성화되어 있지 않습니다.



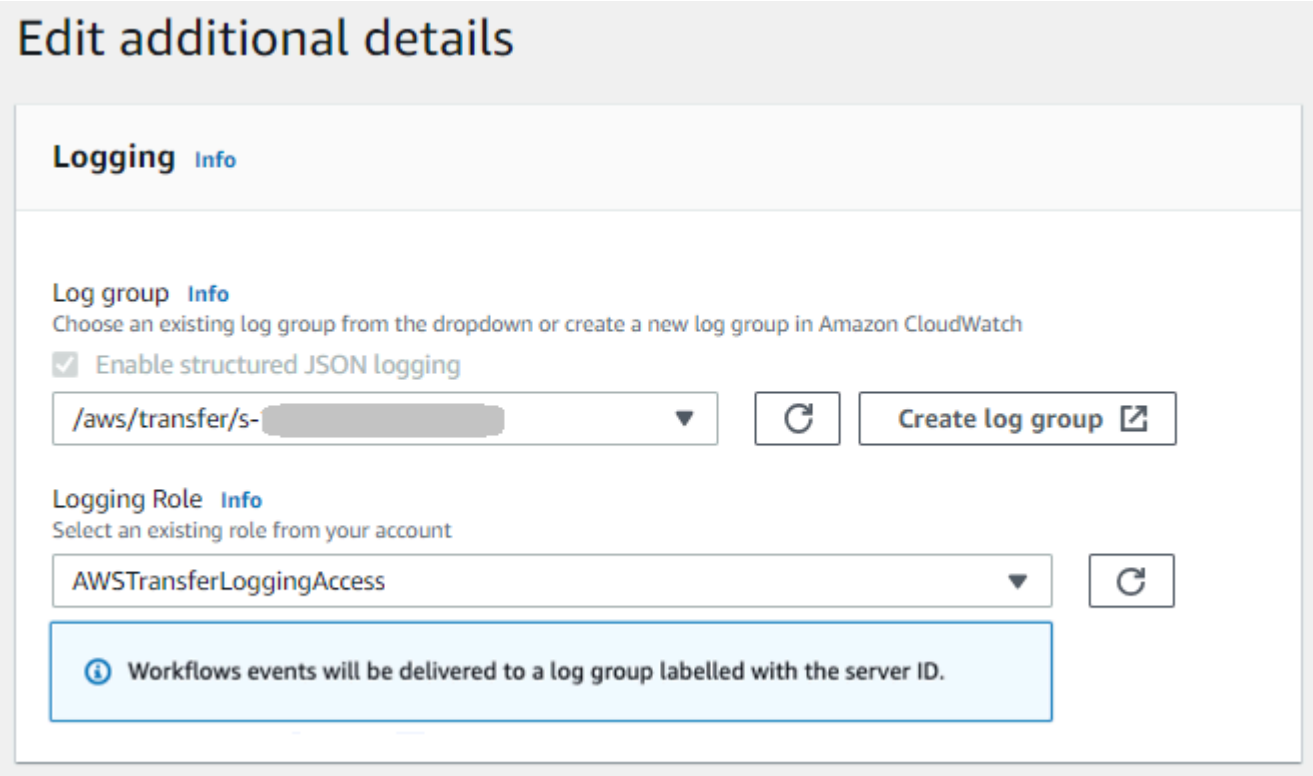
- 서버에 로깅이 활성화되지 않았습니다.



- 서버에는 이미 로깅 역할이 활성화되어 있지만 구조화된 JSON 로깅은 활성화되어 있지 않습니다.



- 서버에는 이미 구조화된 JSON 로깅이 활성화되어 있으며 지정된 로딩 역할도 있습니다.

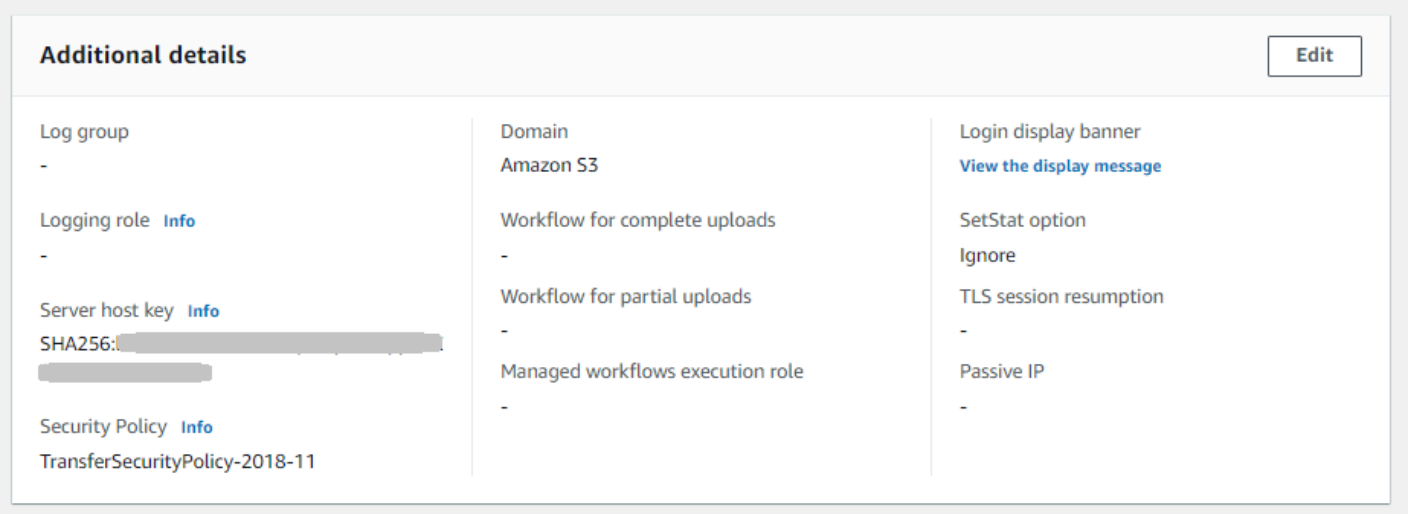


서버 구성 보기

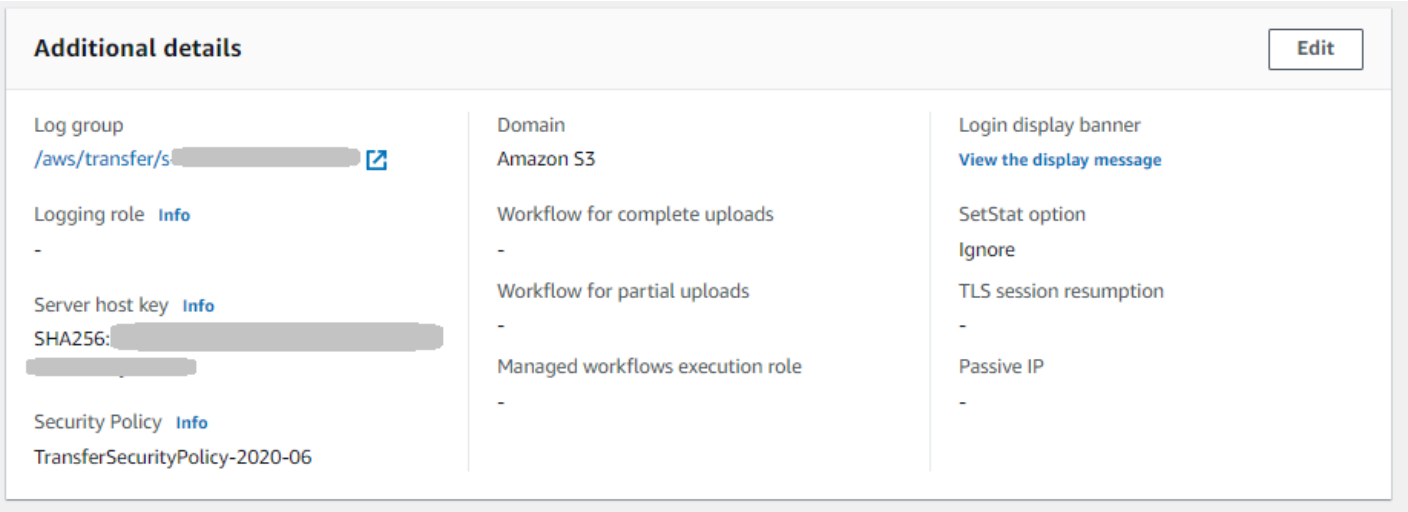
서버 구성 페이지의 세부 정보는 시나리오에 따라 다릅니다.

시나리오에 따라 서버 구성 페이지는 다음 예 중 하나와 비슷할 수 있습니다.

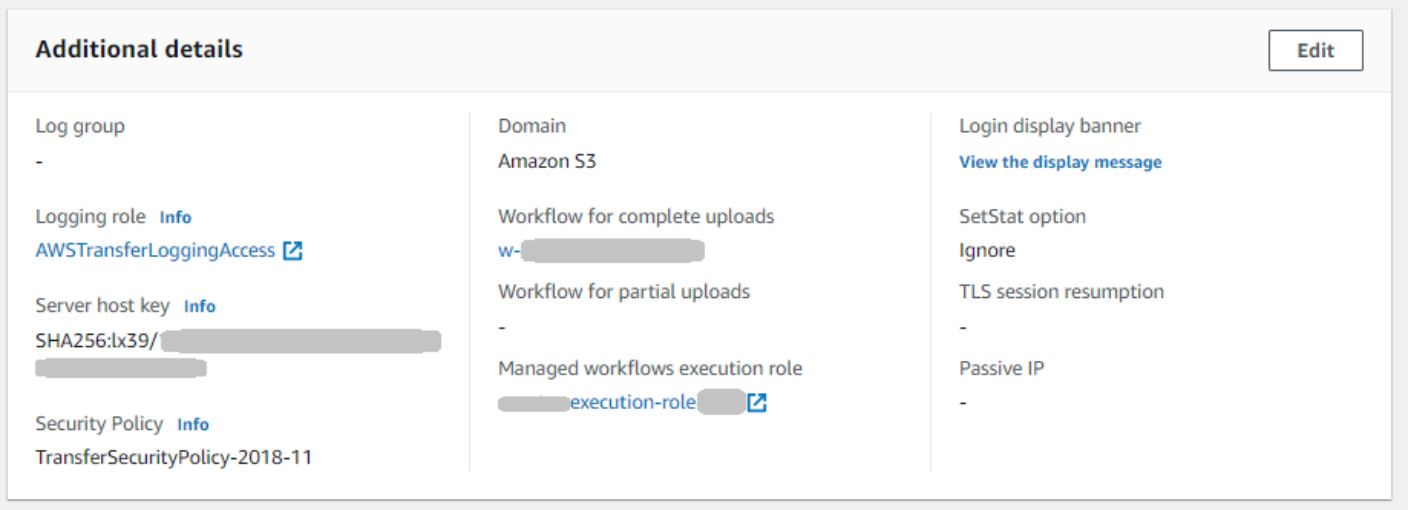
- 로깅이 활성화되지 않았습니다.



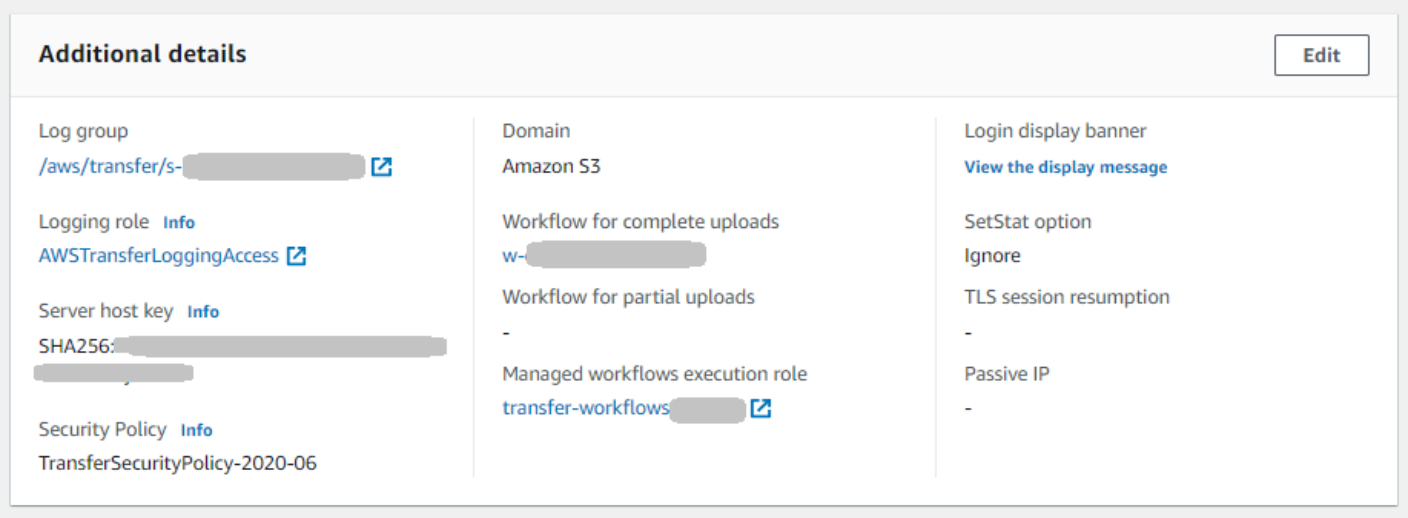
- 구조화된 JSON 로깅이 활성화되었습니다.



- 로깅 역할은 활성화되지만 구조화된 JSON 로깅은 활성화되지 않습니다.



- 두 가지 타입의 로깅(로깅 역할 및 구조화된 JSON 로깅)이 모두 활성화됩니다.



워크플로의 로깅 관리

CloudWatch 워크플로우 진행 상황 및 결과에 대한 통합 감사 및 로깅을 제공합니다. 또한 워크플로우에 AWS Transfer Family 대한 몇 가지 지표를 제공합니다. 이전 1분 간 시작, 성공적으로 완료, 및 실패한 워크플로 실행 수에 대한 지표를 볼 수 있습니다. Transfer Family에 대한 모든 CloudWatch 지표는 에 설명되어 Transfer Family에 대한 CloudWatch 측정항목 사용 있습니다.

워크플로에 대한 Amazon CloudWatch 로그 보기

1. https://console.aws.amazon.com/cloudwatch/ 에서 아마존 CloudWatch 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 로그를 선택한 다음, 로그 그룹을 선택합니다.
3. 로그 그룹 페이지의 탐색 표시줄에서 AWS Transfer Family 서버에 적합한 지역을 선택합니다.
4. 서버에 해당하는 로그 그룹을 선택합니다.

   예를 들어 서버 ID가 `s-1234567890abcdef0`인 경우 로그 그룹은 `/aws/transfer/s-1234567890abcdef0`입니다.
5. 서버의 로그 그룹 세부 정보 페이지에는 가장 최근의 로그 스트림이 표시됩니다. 탐색 중인 사용자에 대한 로그 스트림은 두 가지입니다.

   - 각 Secure Shell(SSH) File Transfer 프로토콜(SFTP) 세션별 로그 스트림
   - 서버에서 실행 중인 워크플로의 로그 스트림 워크플로의 로그 스트림 형식은 *`username`*`.`*`workflowID`*`.`*`uniqueStreamSuffix`*입니다.

   예를 들어 사용자가 `mary-major`이면 다음과 같은 로그 스트림이 있습니다.

```
mary-major-east.1234567890abcdef0
mary.w-abcdef01234567890.021345abcdef6789
```

> **Note**
>
> 이 예에 나열된 16자리 영숫자 식별자는 가상의 식별자입니다. Amazon에서 보는 CloudWatch 값은 다릅니다.

mary-major-usa-east.1234567890abcdef0의 로그 이벤트 페이지에는 각 사용자 세션의 세부 정보가 표시되고 mary.w-abcdef01234567890.021345abcdef6789 로그 스트림에는 워크플로에 대한 세부 정보가 포함됩니다.

다음은 복사 단계가 포함된 워크플로(w-abcdef01234567890)를 기반으로 하는 mary.w-abcdef01234567890.021345abcdef6789의 샘플 로그 스트림입니다.

```
{
    "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 로깅 역할을 구성합니다.

액세스 권한을 설정하려면 리소스 기반 IAM 정책과 해당 액세스 정보를 제공하는 IAM 역할을 만듭니다.

Amazon CloudWatch 로깅을 활성화하려면 먼저 CloudWatch 로깅을 활성화하는 IAM 정책을 생성해야 합니다. 그런 다음 IAM 역할을 만들고 이 역할에 정책을 첨부합니다. 이렇게 하려면 서버를 생성하거나 기존 서버를 편집해야 합니다. 에 대한 CloudWatch 자세한 내용은 Amazon이란 무엇입니까 CloudWatch? 를 참조하십시오. 그리고 아마존 CloudWatch 로그란 무엇입니까? Amazon CloudWatch 사용 설명서에서 확인할 수 있습니다.

다음 예제 IAM 정책을 사용하여 CloudWatch 로깅을 허용하십시오.

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/*"
        }
    ]
}
```

위의 예 정책에서, **Resource**의 경우 *region-id* 및 *AWS ##*를 맞춤 값으로 대체하세요. 예제: **"Resource": "arn:aws::logs:us-east-1:111122223333:log-group:/aws/transfer/*"**

그런 다음 역할을 생성하고 생성한 CloudWatch 로그 정책을 연결합니다.

IAM 역할을 생성하여 정책을 연결하려면

1. 탐색 창에서 역할을 선택한 후 역할 생성을 선택합니다.

   역할 생성 페이지에서 AWS 서비스가 선택되어 있는지 확인합니다.
2. 서비스 목록에서 전송을 선택하고, 다음: 권한을 선택합니다. 이렇게 하면 IAM 역할 AWS Transfer Family 간에 신뢰 관계가 설정됩니다. 또한 `aws:SourceAccount` 및 `aws:SourceArn` 조건 키를 추가하여 혼동되는 부정 문제로부터 자신을 보호하세요. 자세한 설명은 다음 설명서를 참조하세요:

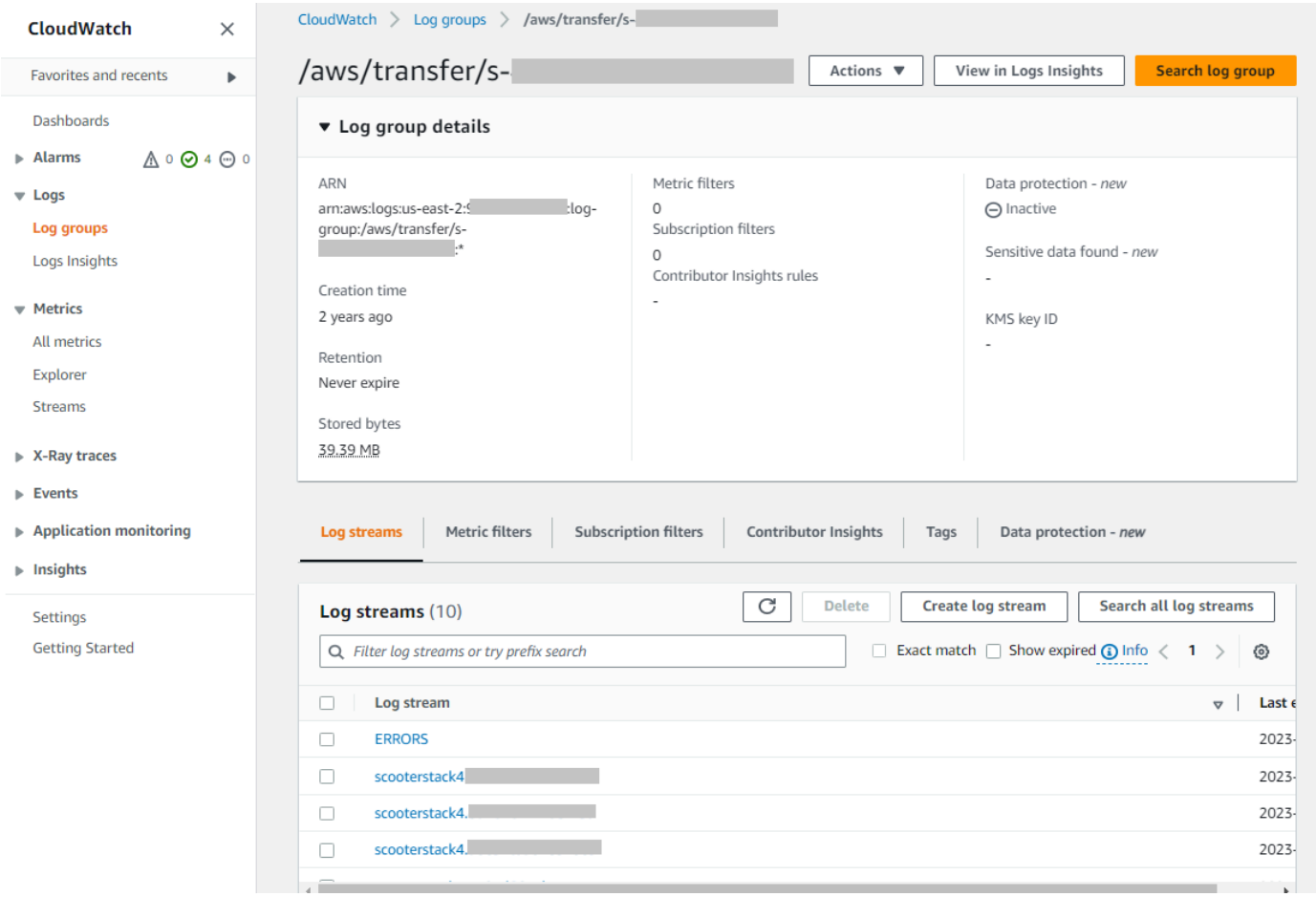
   - 신뢰 관계를 구축하는 절차: AWS Transfer Family신뢰 관계를 구축하기 위해
   - 혼동된 대리인 문제에 대한 설명: 혼동된 대리인 문제
3. 권한 정책 연결 섹션에서 방금 만든 CloudWatch 로그 정책을 찾아 선택하고 다음: 태그를 선택합니다.
4. (옵션) 태그의 키와 값을 입력하고 다음: 검토를 선택합니다.
5. 검토 페이지에 새 역할의 명칭과 설명을 입력한 다음 역할 생성을 선택합니다.
6. 로그를 보려면 Server ID(서버 ID)를 선택하여 서버 구성 페이지를 열고 로그 보기를 선택합니다. 로그 스트림을 볼 수 있는 CloudWatch 콘솔로 리디렉션됩니다.

서버 CloudWatch 페이지에서 사용자 인증 (성공 및 실패), 데이터 업로드 (작업), 데이터 다운로드 (PUTGET작업) 기록을 볼 수 있습니다.

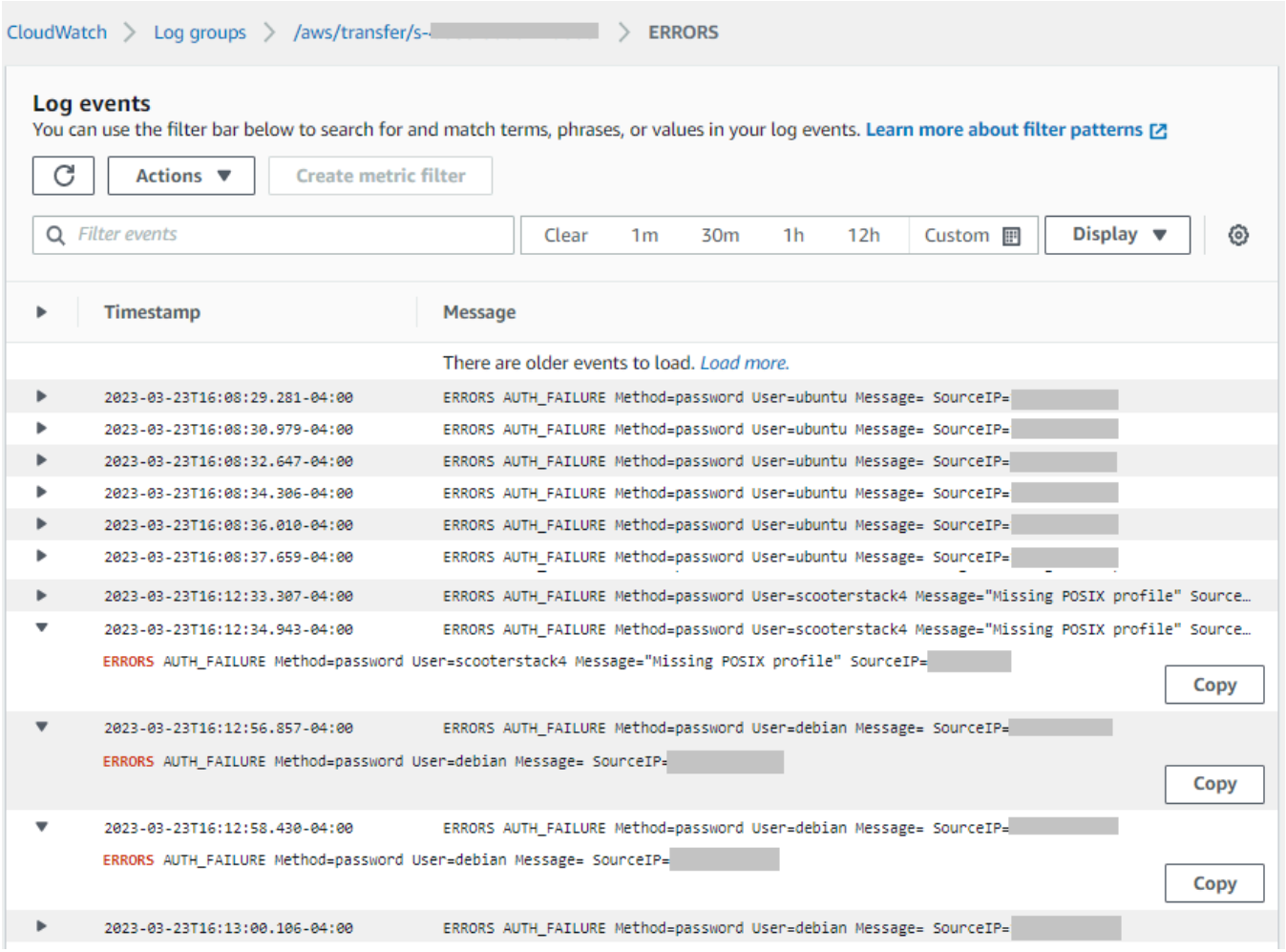
Transfer Family 로그 스트림 보기

Transfer Family 서버 로그를 보려면

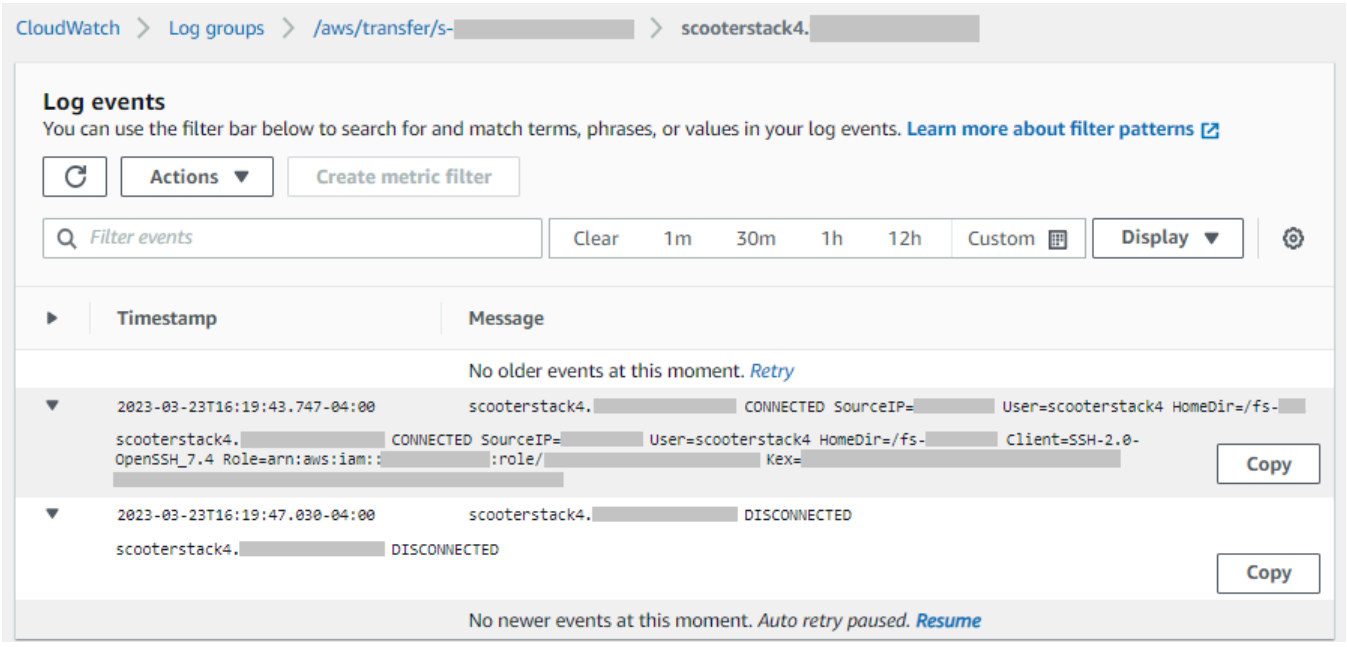
1. 서버의 세부 정보 페이지로 이동합니다.
2. 로그 보기를 선택합니다. 그러면 아마존이 열립니다 CloudWatch.
3. 선택한 서버의 로그 그룹이 표시됩니다.



4. 로그 스트림을 선택하여 스트림의 세부 정보 및 개별 항목을 표시할 수 있습니다.
   - 오류 목록이 있는 경우 해당 목록을 선택하여 서버의 최신 오류에 대한 세부 정보를 볼 수 있습니다.

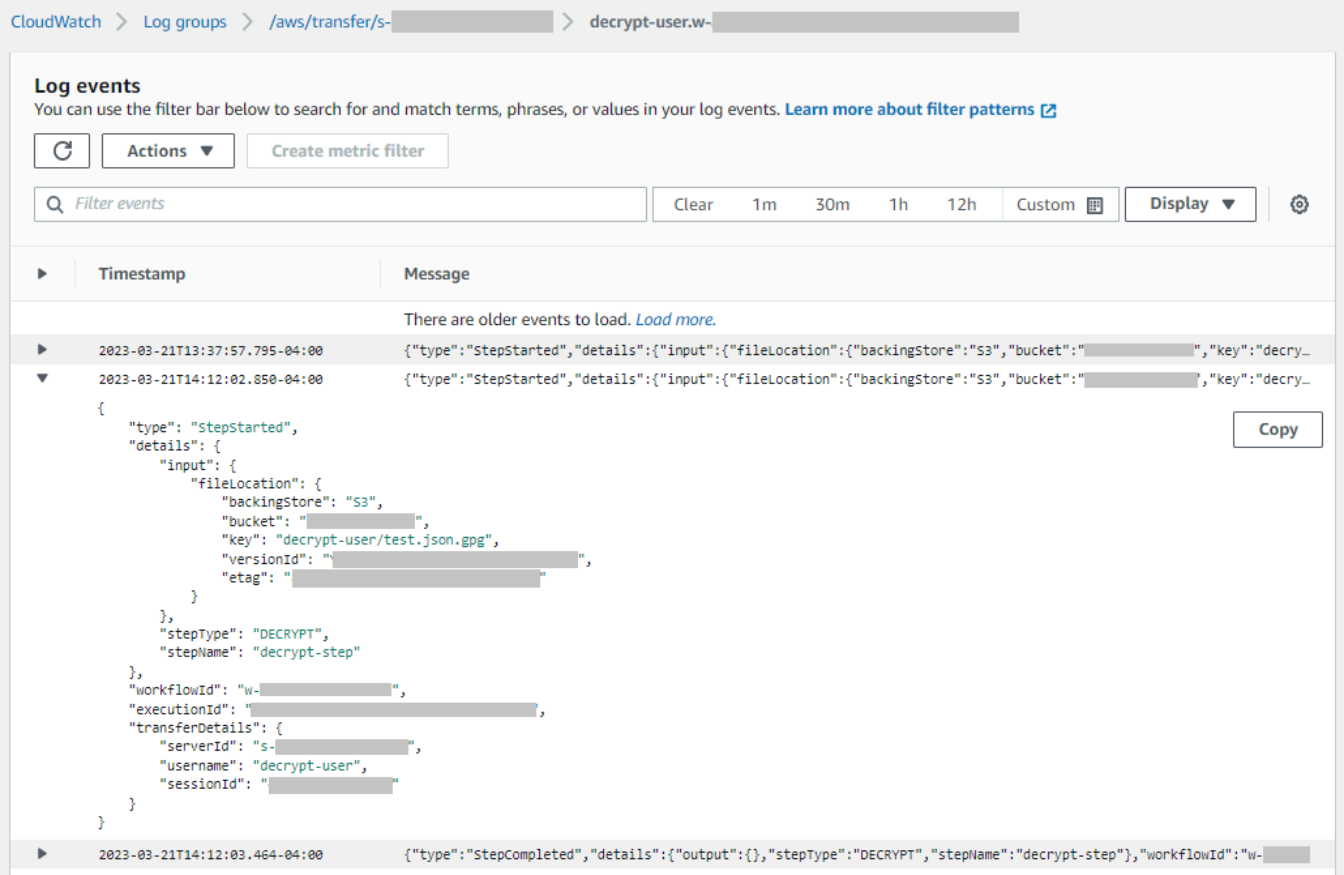


- 예 로그 스트림을 보려면 다른 항목을 선택하세요.



- 서버에 연결된 관리형 워크플로가 있는 경우 워크플로 실행에 대한 로그를 볼 수 있습니다.

> **Note**
>
> 워크플로의 로그 스트림 형식은 *username*.*workflowId*.*uniqueStreamSuffix*입니다. 예를 들어 decrypt-user.w-a1111222233334444.aa1111bb2222는 사용자 **decrypt-user** 및 워크플로 **w-a1111222233334444**의 로그 스트림 명칭일 수 있습니다.



> **Note**
>
> 확장된 로그 항목의 경우 복사를 선택하여 항목을 클립보드에 복사할 수 있습니다. CloudWatch 로그에 대한 자세한 내용은 로그 데이터 보기를 참조하십시오.

아마존 CloudWatch 알람 생성

다음 예는 AWS Transfer Family 측정치를 사용하여 Amazon CloudWatch 경보를 생성하는 방법을 보여줍니다. `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
```

S3 액세스 로그에 Amazon S3 API 호출 로깅

파일 전송 사용자를 대신하여 이루어진 S3 요청을 식별하기 위해 Amazon S3 액세스 로그를 사용하는 경우, 파일 전송을 처리하는 데 맡겨진 IAM 역할을 표시하는 데 `RoleSessionName`이(가) 사용

됩니다. 또한 전송에 사용된 사용자 이름, 세션 ID, 서버 ID와 같은 추가 정보도 표시됩니다. 형식은 `[AWS:Role Unique Identifier]/username.sessionid@server-id`(이)며, 요청자 필드에 포함되어 있습니다. 예를 들어, 다음은 S3 버킷으로 복사된 파일에 대한 S3 액세스 로그의 샘플 요청자 필드 콘텐츠입니다.

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

위의 요청자 필드에는 `IamRoleName`(이)라고 불리는 IAM 역할이 표시됩니다. IAM 역할 고유 식별자에 대한 자세한 설명은 AWS Identity and Access Management 사용자 가이드에서 고유 식별자를 참조하세요.

혼동된 대리자 문제를 제한하는 예

혼동된 대리자 문제는 작업을 수행할 권한이 없는 엔터티가 권한이 더 많은 엔터티에게 작업을 수행하도록 강요할 수 있는 보안 문제입니다. 에서 AWS서비스 간 사칭으로 인해 대리인 문제가 혼동될 수 있습니다. 자세한 내용은 교차 서비스 혼동된 대리인 방지를 참조하세요.

> **Note**
>
> 다음 예에서는 자신의 정보로 각각의 *### ## ## ###*를 바꿉니다.
> 이 예시에서는 서버에 연결된 워크플로가 없는 경우 워크플로의 ARN 세부 정보를 제거할 수 있습니다.

다음 예제 로깅/호출 정책은 계정의 모든 서버 (및 워크플로) 가 역할을 맡을 수 있도록 허용합니다.

```
{
    "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/*"
                   ]
                }
            }
        }
    ]
}
```

다음 예제 로깅/호출 정책은 특정 서버 (및 워크플로) 가 역할을 맡도록 허용합니다.

```
{
    "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 Family의 로그 구조

이 항목에서는 Transfer Family 로그에 입력되는 필드 (JSON 구조화된 로그 항목과 기존 로그 항목 모두에 대해) 에 대해 설명합니다.

주제

Transfer Family를 위한 JSON 구조화된 로그

다음 표에는 새로운 JSON 구조 로그 형식의 Transfer Family SFTP/FTP/FTPS 작업에 대한 로그 입력 필드에 대한 세부 정보가 포함되어 있습니다.

필드	설명	입력 예
activity-type	The action by the user	열기 | 닫기 | 부분_닫기 | 연결 끊김 | 연결됨
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 암호화 알고리즘)	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 PATH: 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 암호화 알고리즘)	diffie-hellman-group14-sha256

필드	설명	입력 예
message	Provides more information related to the error	*<string>*
method	The authentication method	publickey
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:서버/s -1234567890 akeu2js2
role	The IAM role of the user	arn:aws:iam: :0293883675: 역할/테스트 사용자 역할
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 Family의 레거시 로그

다음 표에는 다양한 Transfer Family 작업에 대한 로그 항목의 세부 정보가 나와 있습니다.

> **Note**
>
> 이러한 항목은 새로운 JSON 구조화된 로그 형식이 아닙니다.

다음 표에는 다양한 Transfer Family 작업에 대한 로그 항목의 세부 정보가 새로운 JSON 구조 로그 형식으로 포함되어 있습니다.

작업	Amazon 로그 내의 해당 CloudWatch 로그
인증 실패 횟수	ERRORS AUTH_FAILURE Method=publickey User=lhr Message="RSA SHA256:Lfz3R2nmLY4raK+b7Rb1rSvUIbAE+a+Hxg0c7l1JIZ0" SourceIP=3.8.172.211
복사/태그 지정/삭제/복호화 워크플로우	{"유형":" StepStarted “, “세부 정보”: {"입력”: {"파일 위치”: {"백업 저장소”: “EFS”, “파일 시스템 ID”: “fs-12345678", “경로”:” /lhr/regex.py “}}, “단계 유형”: “태그”, “단계 이름”: “성공_태그_단계"}, “워크플로 ID”: “w-1111aaaa22bbbb3", “실행 ID”: “81234abcd-1234-efgh-5678-ijklmnopqr90", “전송 세부 정보”: {"서버 ID”: “s-1234abcd5678efghi”, “사용자 이름”: “lhr”, “세션 ID”: “1234567890" abcdef0"}}
맞춤 단계 워크플로우	{"유형":" “, “세부 정보”: {"출력”: {"토큰”: CustomStepInvoked “MZM4MJG5YWUTYTEzMy 00 YjIz LWI3OG MtYz U4OGI2 ZjQyMzE5"}, “단계 유형”: “사용자 정의”, “단계 이름”: “efs-s3_copy_2"}, “워크플로우 ID”: “w-9283e49d33297c3f7", “실행 ID”: “1234abcd-1234-efgh-5678-ijklmnopqr90", “전송 세부 정보”: {"서버 ID”: “s-zzzz11aaaa22223", “사용자 이름”: “lhr”, “세션 ID”: “1234567890abcdef0"}
삭제	lhr.33a8fb495ffb383b DELETE Path=/bucket/user/123.jpg

작업	Amazon 로그 내의 해당 CloudWatch 로그
다운로드	lhr.33a8fb495ffb383b OPEN Path=/bucket/user/123.jpg Mode=READ llhr.33a8fb495ffb383b 닫기 경로=/버킷/사용자/123.jpg BytesOut =3618546
로그인/로그아웃	사용자.914984e553bcddb6 커넥티드 소스 IP=1.22.111.222 사용자=LHR =논리적 클라이언트=SSH-2.0-opensh_7.4 역할=arn:AWS: :iam: :123456789012:role/sftp-s3-access HomeDir user.914984e553bcddb6 DISCONNECTED
명칭 바꾸기	lhr.33a8fb495ffb383b 이름 바꾸기 경로=/버킷/사용자/Lambo.png =/버킷/사용자/페라리.png NewPath
샘플 워크플로우 오류 로그	{"type”:” “, “details”: {"오류 유형”: StepErrored “BAD_REQUEST”, “오류 메시지”: “Ef 파일에 태그를 지정할 수 없음”, “단계 유형”: “TAG”, “단계 이름”: “성공_tag_step"}, “워크플로 ID”: “w-1234abcd5678efghi”, “실행 ID”: “실행 ID”: 81234abcd-1234-efgh-5678-ijklmnopqr90", “전송 세부 정보”: {"서버 ID”: “s-1234abcd5678efghi”, “사용자 이름”: “lhr”, “세션 ID”: “1234567890abcdef0"}
Symlinks	lhr.eb49cf7b8651e6d5 CREATE_SYMLINK LinkPath =/fs-12345678/lhr/pqr.jpg TargetPath =abc.jpg
업로드	lhr.33a8fb495ffb383b OPEN Path=/bucket/user/123.jpg Mode=CREATE|TRUNCATE|WRITE lhr.33a8fb495ffb383b 닫기 경로=/버킷/사용자/123.jpg =3618546 BytesIn

작업	Amazon 로그 내의 해당 CloudWatch 로그
워크플로	{"유형":" ", "세부 정보": {"입력": {" Execution Started ": {" 백업 스토어": "EFS", "파일 시스템 ID": "fs-12345678", initialFileLocation "경로":" /lhr/regex.py "}}, "워크플로 ID": "w-1111aaaa22bb3", "실행 ID": "1234abcd-12cd-127" 4-efgh-5678-ijklmnopqr90", "전송 세부 정보": {"서버 ID": "s-zzzz11aaaa2223", "사용자 이름": "lhr", "세션 ID": "1234567890abcdef0"} {"유형":" StepStarted ", "세부 정보": {"입력": {"파일 위치": {"백업 저장소": "EFS", "파일 시스템 ID": "fs-12345678", "경로":" /lhr/regex.py "}}, "단계 유형": "사용자 지정", "단계 이름": "efs-s3_copy_2"}, "워크플로우 ID": "w-9283e49d33297c3f7", "실행 ID": "1234abcd-1234-efgh-5678-ijklmnopqr90", "전송 세부 정보": {"서버 ID": "s-18ca49dce5d842e0b", "사용자 이름": "lhr", "세션 ID": "1234567890abcdef0"}}

예제 CloudWatch 로그 항목

이 항목에서는 예제 로그 항목을 제공합니다.

주제

- 전송 세션 로그 항목의 예
- SFTP 커넥터의 예제 로그 항목
- 키 교환 알고리즘 실패에 대한 예제 로그 항목

전송 세션 로그 항목의 예

이 예에서 SFTP 사용자는 Transfer Family 서버에 연결하고 파일을 업로드한 다음 세션과의 연결을 끊습니다.

다음 로그 항목은 Transfer Family 서버에 연결하는 SFTP 사용자를 반영합니다.

```
{
   "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"
}
```

다음 로그 항목은 SFTP 사용자가 Amazon S3 버킷에 파일을 업로드하는 모습을 반영합니다.

```
{
   "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"
}
```

다음 로그 항목은 SFTP 사용자가 SFTP 세션에서 연결을 끊은 상황을 반영합니다. 먼저 클라이언트가 버킷 연결을 닫은 다음 SFTP 세션 연결을 끊습니다.

```
{
   "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 커넥터의 예제 로그 항목

이 섹션에는 전송 성공 및 실패 모두에 대한 예제 로그가 포함되어 있습니다. 라는 이름의 `/aws/transfer/connector-id` 로그 그룹에 로그가 생성됩니다. 여기서 *connector-id#* SFTP 커넥터의 식별자입니다.

> **Note**
>
> SFTP 커넥터의 로그 항목은 명령을 실행할 때만 생성됩니다. `StartFileTransfer`

이 로그 항목은 성공적으로 완료된 전송을 위한 것입니다.

```
{
    "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"
    "bytes": 514
}
```

이 로그 항목은 제한 시간이 초과되어 성공적으로 완료되지 않은 전송에 대한 것입니다.

```
{
    "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"
}
```

이 로그 항목은 성공한 SEND 작업을 위한 것입니다.

```
{
    "operation": "SEND",
    "timestamp": "2024-04-24T18:16:12.513207284Z",
    "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": "/DOC-EXAMPLE-BUCKET/my-test-folder/connector-metrics-us-
east-1-2024-01-02.csv",
    "status-code": "COMPLETED",
    "start-time": "2024-04-24T18:16:12.295235884Z",
    "end-time": "2024-04-24T18:16:12.461840732Z",
    "account-id": "255443218509",
    "connector-arn": "arn:aws:transfer:us-east-1:255443218509:connector/connector-id",
    "bytes": 275
}
```

이전 로그 예제의 일부 키 필드에 대한 설명.

- `timestamp`로그가 추가되는 시기를 나타냅니다 CloudWatch. `start-time`커넥터가 실제로 전송을 시작하고 완료하는 시간에 `end-time` 해당합니다.
- `transfer-id`각 `start-file-transfer` 요청에 할당되는 고유 식별자입니다. 사용자가 단일 `start-file-transfer` API 호출로 여러 파일 경로를 전달하면 모든 파일이 동일하게 `transfer-id` 공유됩니다.
- `file-transfer-id`전송된 각 파일에 대해 생성되는 고유한 값입니다. 참고로 의 첫 부분은 와 같습니다`transfer-id`. `file-transfer-id`

키 교환 알고리즘 실패에 대한 예제 로그 항목

이 섹션에는 KEX (키 교환 알고리즘) 가 실패한 예제 로그가 포함되어 있습니다. 다음은 구조화된 로그에 대한 ERRORS 로그 스트림의 예입니다.

이 로그 항목은 호스트 키 유형 오류가 있는 예입니다.

```
{
    "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"
}
```

이 로그 항목은 KEX 불일치가 있는 예입니다.

```
{
    "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 key exchange method found",
    "kex": "diffie-hellman-group1-sha1,diffie-hellman-group14-sha1,diffie-hellman-
group14-sha256"
}
```

Transfer Family에 대한 CloudWatch 측정항목 사용

> **Note**
>
> Transfer Family 콘솔 자체에서도 Transfer Family에 대한 지표를 가져올 수 있습니다. 자세한 내용은 콘솔 내 사용량 모니터링 를 참조하세요.

CloudWatch 지표를 사용하여 서버에 대한 정보를 얻을 수 있습니다. 지표는 게시되는 시간순으로 정렬된 데이터 요소 집합을 CloudWatch 나타냅니다. 지표를 사용할 때는 Transfer Family 네임스페이스,

지표 명칭 및 차원을 지정해야 합니다. 지표에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서의 지표를 참조하십시오.

다음 표에는 Transfer CloudWatch Family의 측정 기준이 설명되어 있습니다.

네임스페이스	지표	설명
AWS/Transfer	BytesIn	서버로 전송된 총 바이트 수. 단위: 개 기간: 5분
	BytesOut	서버에서 전송된 총 바이트 수입니다. 단위: 개 기간: 5분
	FilesIn	서버로 전송된 총 파일 수. AS2 프로토콜을 사용하는 서버의 경우 이 지표는 수신된 메시지 수를 나타냅니다. 단위: 개 기간: 5분
	FilesOut	서버에서 전송된 총 파일 수. 단위: 개 기간: 5분
	InboundMessage	거래 파트너로부터 성공적으로 수신한 AS2 메시지의 총 개수. 단위: 개 기간: 5분

네임스페이스	지표	설명
	InboundFailedMessage	거래 파트너로부터 성공적으로 수신되지 못한 AS2 메시지의 총 개수. 즉, 거래 파트너가 메시지를 보냈지만 Transfer Family 서버가 메시지를 성공적으로 처리하지 못했습니다. 단위: 개 기간: 5분
	OnUploadExecutionsStarted	서버에서 시작된 워크플로를 실행한 총 횟수. 단위: 개 기간: 1분
	OnUploadExecutionsSuccess	서버에서 성공적인 워크플로를 실행한 총 횟수. 단위: 개 기간: 1분
	OnUploadExecutionsFailed	서버에서 실패한 워크플로를 실행한 총 횟수. 단위: 개 기간: 1분

Transfer Family 크기

차원은 지표의 보안 인증에 속하는 명칭/값 쌍입니다. 치수에 대한 자세한 내용은 Amazon CloudWatch 사용 설명서의 치수를 참조하십시오.

다음 표에는 Transfer CloudWatch Family의 차원이 설명되어 있습니다.

측정기준	설명
ServerId	사용자의 고유 ID입니다.

AWS 사용자 알림 와 함께 사용 AWS Transfer Family

를 사용하여 다양한 전송 채널을 AWS 사용자 알림설정하여 AWS Transfer Family 이벤트에 대한 알림을 받을 수 있습니다. 이벤트가 지정한 규칙과 일치하면 알림을 받습니다.

이메일, AWS Chatbot 채팅 알림 또는 AWS Console Mobile Application 푸시 알림을 비롯한 여러 채널을 통해 이벤트에 대한 알림을 받을 수 있습니다. 콘솔 알림 센터에서도 알림을 볼 수 있습니다. 사용자 알림 집계를 지원하므로 특정 이벤트 중에 받는 알림 수를 줄일 수 있습니다.

자세한 내용은 AWS Transfer Family 관리형 워크플로를 사용하여 파일 배달 알림 사용자 지정 블로그 게시물 및 AWS 사용자 알림무엇입니까를 참조하십시오. AWS 사용자 알림 사용 설명서에서

쿼리를 사용하여 로그 항목을 필터링합니다.

CloudWatch 쿼리를 사용하여 Transfer Family의 로그 항목을 필터링하고 식별할 수 있습니다. 이 섹션에는 몇 가지 예가 포함되어 있습니다.

1. https://console.aws.amazon.com/cloudwatch/ 에서 AWS Management Console 로그인하고 CloudWatch 콘솔을 엽니다.
2. 쿼리 또는 규칙을 생성할 수 있습니다.
   - Logs Insights 쿼리를 만들려면 왼쪽 탐색 패널에서 Logs Insights를 선택한 다음 쿼리의 세부 정보를 입력합니다.
   - 기여자 인사이트 규칙을 만들려면 왼쪽 탐색 패널에서 Insights > Contributor Insights를 선택한 다음 규칙의 세부 정보를 입력합니다.
3. 생성한 쿼리 또는 규칙을 실행합니다.

인증 실패에 가장 많이 기여한 원인 보기

구조화된 로그에서 인증 실패 로그 항목은 다음과 비슷합니다.

```
{
  "method":"password",
  "activity-type":"AUTH_FAILURE",
  "source-ip":"999.999.999.999",
  "resource-arn":"arn:aws:transfer:us-east-1:999999999999:server/s-0123456789abcdef",
  "message":"Invalid user name or password",
  "user":"exampleUser"
}
```

다음 쿼리를 실행하여 인증 실패의 주요 원인을 확인하세요.

```
filter @logStream = 'ERRORS'
| filter `activity-type` = 'AUTH_FAILURE'
| stats count() as AuthFailures by user, method
| sort by AuthFailures desc
| limit 10
```

CloudWatch 로그 인사이트를 사용하는 대신 CloudWatch 기여자 인사이트 규칙을 만들어 인증 실패를 확인할 수 있습니다. 다음과 비슷한 규칙을 만드세요.

```
{
    "AggregateOn": "Count",
    "Contribution": {
        "Filters": [
            {
                "Match": "$.activity-type",
                "In": [
                    "AUTH_FAILURE"
                ]
            }
        ],
        "Keys": [
            "$.user"
        ]
    },
    "LogFormat": "JSON",
    "Schema": {
        "Name": "CloudWatchLogRule",
        "Version": 1
    },
    "LogGroupARNs": [
        "arn:aws:logs:us-east-1:999999999999:log-group:/customer/structured_logs"
    ]
}
```

파일이 열린 로그 항목 보기

구조화된 로그에서 파일 읽기 로그 항목은 다음과 비슷합니다.

```
{
  "mode":"READ",
```

```
  "path":"/fs-0df669c89d9bf7f45/avtester/example",
  "activity-type":"OPEN",
  "resource-arn":"arn:aws:transfer:us-east-1:999999999999:server/s-0123456789abcdef",
  "session-id":"0049cd844c7536c06a89"
}
```

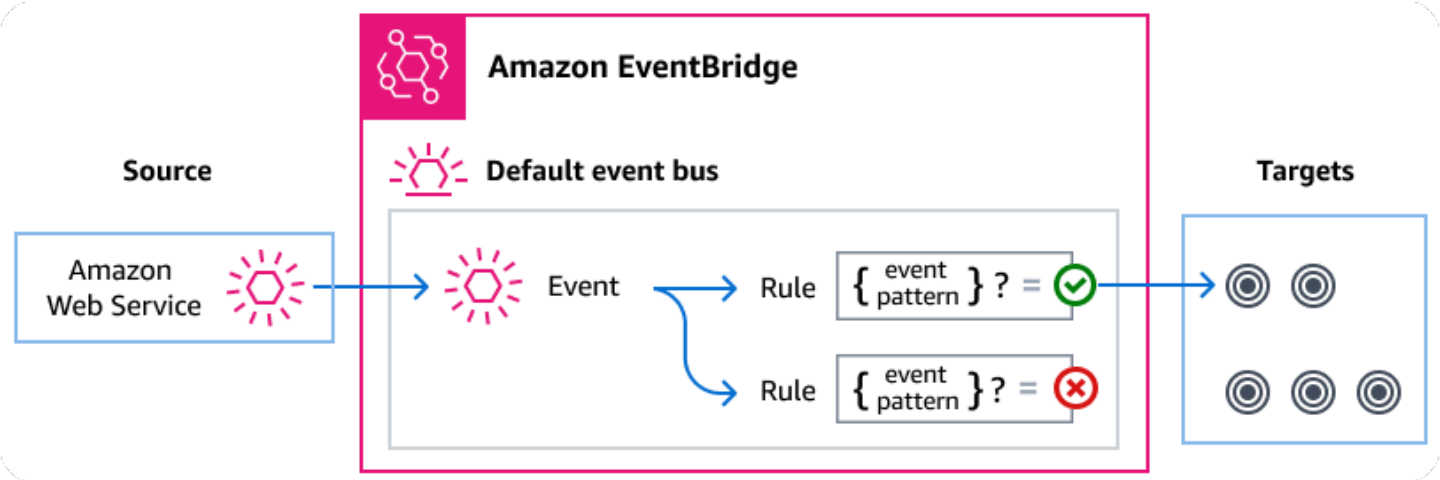
다음 쿼리를 실행하여 파일이 열렸음을 나타내는 로그 항목을 확인하세요.

```
filter `activity-type` = 'OPEN'
| display @timestamp, @logStream, `session-id`, mode, path
```

를 사용하여 Transfer Family 이벤트 관리 Amazon EventBridge

Amazon EventBridge 이벤트를 사용하여 애플리케이션 구성 요소를 서로 연결하는 서버리스 서비스로, 이를 통해 확장 가능한 이벤트 기반 애플리케이션을 보다 쉽게 구축할 수 있습니다. 이벤트 기반 아키텍처는 이벤트를 내보내고 이에 응답하여 함께 작동하는 느슨하게 결합된 소프트웨어 시스템을 구축하는 스타일입니다. 이벤트는 리소스나 환경의 변화를 나타냅니다.

대부분의 AWS 서비스와 마찬가지로 이벤트를 Transfer Family 생성하여 기본 이벤트 버스로 전송합니다. EventBridge 기본 이벤트 버스는 모든 AWS 계정에 자동으로 프로비저닝된다는 점에 유의하십시오. 이벤트 버스는 이벤트를 수신하여 0개 이상의 목적지 또는 대상에 전달하는 라우터입니다. 이벤트가 도착하면 이를 평가하는 이벤트 버스 규칙을 지정합니다. 각 규칙은 이벤트가 규칙의 이벤트 패턴과 일치하는지 여부를 확인합니다. 이벤트가 일치하면 이벤트 버스가 이벤트를 하나 이상의 지정된 대상으로 보냅니다.



주제

- Transfer Family 이벤트
- 규칙을 Transfer Family 사용하여 이벤트 전송 EventBridge
- Amazon EventBridge 권한
- 추가 EventBridge 리소스
- Transfer Family 이벤트 세부 정보 참조

Transfer Family 이벤트

Transfer Family 이벤트를 기본 EventBridge 이벤트 버스로 자동 전송합니다. 각 규칙이 이벤트 패턴과 하나 이상의 대상을 포함하는 규칙을 이벤트 버스에 생성할 수 있습니다. 규칙의 이벤트 패턴과 일치하는 이벤트는 최선을 다해 지정된 대상에 전달되지만 일부 이벤트는 순서가 맞지 않게 전달될 수 있습니다.

에서 생성되는 이벤트는 다음과 같습니다 Transfer Family. 자세한 내용은 Amazon EventBridge 사용 설명서의EventBridge 이벤트를 참조하십시오.

SFTP, FTPS 및 FTP 서버 이벤트

이벤트 세부 정보 유형	설명
FTP 파일 서버 다운로드 완료	FTP 프로토콜용 파일이 성공적으로 다운로드되었습니다.
FTP 파일 서버 다운로드 실패	FTP 프로토콜에 대한 파일 다운로드 시도가 실패했습니다.
FTP 파일 서버 업로드 완료	FTP 프로토콜용 파일이 성공적으로 업로드되었습니다.
FTP 파일 서버 업로드 실패	FTP 프로토콜에 대한 파일 업로드 시도가 실패했습니다.
FTPS 파일 서버 다운로드 완료	FTPS 프로토콜용 파일이 성공적으로 다운로드되었습니다.
FTPS 파일 서버 다운로드 실패	FTPS 프로토콜에 대한 파일 다운로드 시도가 실패했습니다.
FTPS 파일 서버 업로드 완료	FTPS 프로토콜용 파일이 성공적으로 업로드되었습니다.
FTPS 파일 서버 업로드 실패	FTPS 프로토콜에 대한 파일 업로드 시도가 실패했습니다.
SFTP 서버 파일 다운로드 완료	SFTP 프로토콜용 파일이 성공적으로 다운로드되었습니다.
SFTP 서버 파일 다운로드 실패	SFTP 프로토콜에 대한 파일 다운로드 시도가 실패했습니다.
SFTP 서버 파일 업로드 완료	SFTP 프로토콜용 파일이 성공적으로 업로드되었습니다.
SFTP 서버 파일 업로드 실패	SFTP 프로토콜에 대한 파일 업로드 시도가 실패했습니다.

SFTP 커넥터 이벤트

이벤트 세부 정보 유형	설명
SFTP 커넥터 파일 전송 완료	커넥터에서 원격 SFTP 서버로의 파일 전송이 성공적으로 완료되었습니다.
SFTP 커넥터 파일 전송 실패	커넥터에서 원격 SFTP 서버로의 파일 전송이 실패했습니다.
SFTP 커넥터 파일 검색이 완료되었습니다.	원격 SFTP 서버에서 커넥터로의 파일 전송이 성공적으로 완료되었습니다.
SFTP 커넥터 파일 검색에 실패했습니다.	원격 SFTP 서버에서 커넥터로의 파일 전송이 실패했습니다.
SFTP 커넥터 디렉터리 목록 완료	시작 파일 디렉터리 목록 호출이 성공적으로 완료되었습니다.
SFTP 커넥터 디렉토리 목록 실패	시작 파일 디렉토리 목록이 실패했습니다.

A2S 이벤트

이벤트 세부 정보 유형	설명
AS2 페이로드 수신 완료	AS2 메시지의 페이로드가 수신되었습니다.
AS2 페이로드 수신 실패	AS2 메시지의 페이로드가 수신되지 않았습니다.
AS2 페이로드 전송 완료	AS2 메시지의 페이로드가 성공적으로 전송되었습니다.
AS2 페이로드 전송 실패	AS2 메시지의 페이로드를 전송하지 못했습니다.
AS2 MDN 수신 완료	AS2 메시지에 대한 메시지 처리 알림이 수신되었습니다.
AS2 MDN 수신 실패	AS2 메시지에 대한 메시지 처리 알림이 수신되지 않았습니다.

이벤트 세부 정보 유형	설명
AS2 MDN 전송 완료	AS2 메시지에 대한 메시지 처리 알림이 성공적으로 전송되었습니다.
AS2 MDN 전송 실패	AS2 메시지에 대한 메시지 처리 알림을 보내지 못했습니다.

규칙을 Transfer Family 사용하여 이벤트 전송 EventBridge

EventBridge 기본 이벤트 버스가 대상으로 Transfer Family 이벤트를 보내도록 하려면 원하는 Transfer Family 이벤트의 데이터와 일치하는 이벤트 패턴을 포함하는 규칙을 만들어야 합니다.

다음과 같은 일반 단계에 따라 규칙을 생성할 수 있습니다.

1. 다음을 지정하는 규칙의 이벤트 패턴을 생성합니다.
   - Transfer Family 규칙에 의해 평가되는 이벤트의 소스입니다.
   - (선택 사항) 해당 데이터와 일치시킬 기타 모든 이벤트 데이터.

   자세한 정보는 ???을 참조하세요.
2. (선택 사항) 규칙 대상으로 정보를 EventBridge 보내기 전에 이벤트의 데이터를 사용자 지정하는 입력 변환기를 만드십시오.

   자세한 내용은 EventBridge 사용 설명서의 입력 변환을 참조하세요.
3. 이벤트 패턴과 일치하는 이벤트를 EventBridge 전송할 대상을 지정합니다.

   대상은 다른 AWS 서비스, SaaS (Software as a Service) 애플리케이션, API 대상 또는 기타 사용자 지정 엔드포인트일 수 있습니다. 자세한 내용은 EventBridge 사용 설명서의 대상을 참조하세요.

이벤트 버스 규칙 생성에 대해 자세히 알아보려면 EventBridge 사용 설명서의 이벤트에 대응하는 규칙 생성을 참조하세요.

이벤트에 대한 이벤트 패턴 생성 Transfer Family

기본 이벤트 버스에 이벤트를 전달할 때 Transfer Family 는 각 규칙에 정의된 이벤트 패턴을 EventBridge 사용하여 이벤트를 규칙의 대상에 전달할지 여부를 결정합니다. 이벤트 패턴이 원하는 Transfer Family 이벤트의 데이터와 일치합니다. 각 이벤트 패턴은 다음을 포함하는 JSON 객체입니다.

- 이벤트를 전송하는 서비스를 식별하는 `source` 속성입니다. Transfer Family 이벤트의 경우 소스는 입니다. `aws.transfer`
- (선택 사항) 일치시킬 이벤트 유형의 배열이 포함된 `detail-type` 속성입니다.
- (선택 사항) 일치시킬 다른 이벤트 데이터가 포함된 `detail` 속성입니다.

예를 들어, 다음 이벤트 패턴은 의 모든 이벤트와 일치합니다 Transfer Family.

```
{
  "source": ["aws.transfer"]
}
```

다음 이벤트 패턴 예제는 모든 SFTP 커넥터 이벤트와 일치합니다.

```
{
  "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"]
}
```

다음 이벤트 패턴 예는 모든 Transfer Family 실패 이벤트와 일치합니다.

```
{
  "source": ["aws.transfer"],
  "detail-type": [{"wildcard", "*Failed"}]
}
```

다음 이벤트 패턴 예시는 *### ####* 대한 성공적인 SFTP 다운로드와 일치합니다.

```
{
  "source": ["aws.transfer"],
  "detail-type": ["SFTP Server File Download Completed"],
  "detail": {
    "username": [username]
  }
}
```

자세한 내용은 EventBridge 사용 설명서의 이벤트 패턴을 참조하세요.

에서 이벤트의 이벤트 패턴을 테스트합니다 Transfer Family . EventBridge

EventBridge 샌드박스를 사용하면 규칙을 만들거나 편집하는 광범위한 프로세스를 완료하지 않고도 이벤트 패턴을 빠르게 정의하고 테스트할 수 있습니다. 샌드박스를 사용하면 이벤트 패턴을 정의하고 샘플 이벤트를 사용하여 패턴이 원하는 이벤트와 일치하는지 확인할 수 있습니다. EventBridge 샌드박스에서 직접 해당 이벤트 패턴을 사용하여 새 규칙을 생성할 수 있는 옵션을 제공합니다.

자세한 내용은 사용 EventBridge 설명서의 EventBridge 샌드박스를 사용한 이벤트 패턴 테스트를 참조하십시오.

Amazon EventBridge 권한

Transfer Family 이벤트를 전달하는 데 추가 권한이 필요하지 않습니다 Amazon EventBridge.

지정하는 대상에는 특정 권한 또는 구성이 필요할 수 있습니다. 대상에 특정 서비스를 사용하는 방법에 대한 자세한 내용은 Amazon EventBridge 사용 설명서의 Amazon EventBridge 대상을 참조하세요.

추가 EventBridge 리소스

를 사용하여 이벤트를 처리하고 관리하는 방법에 대한 자세한 내용은 사용 Amazon EventBridge EventBridge 설명서의 다음 항목을 참조하십시오.

- 이벤트 버스의 작동 방식에 대한 자세한 내용은 Amazon EventBridge 이벤트 버스를 참조하세요.
- 이벤트 구조에 대해 자세히 알아보려면 이벤트를 참조하세요.
- 규칙과 이벤트를 일치시킬 때 사용할 이벤트 패턴을 구성하는 방법에 대한 자세한 내용은 이벤트 패턴을 참조하십시오. EventBridge
- EventBridge 에서 처리하는 이벤트를 지정하는 규칙을 생성하는 방법에 대한 자세한 내용은 규칙을 참조하세요.
- 일치하는 이벤트를 EventBridge 전송할 서비스 또는 기타 대상을 지정하는 방법에 대한 자세한 내용은 대상을 참조하십시오.

Transfer Family 이벤트 세부 정보 참조

AWS 서비스의 모든 이벤트에는 이벤트에 대한 메타데이터가 포함된 공통 필드 집합이 있습니다. 이러한 메타데이터에는 이벤트 소스인 AWS 서비스, 이벤트가 생성된 시간, 이벤트가 발생한 계정 및 지역 등이 포함될 수 있습니다. 이러한 일반 필드에 대한 정의는 Amazon EventBridge 사용 설명서의 이벤트 구조 참조를 참조하세요.

또한 각 이벤트에는 해당 특정 이벤트와 관련된 데이터를 포함하는 `detail` 필드가 있습니다. 다음 참조는 다양한 Transfer Family 이벤트의 세부 정보 필드를 정의합니다.

를 EventBridge 사용하여 Transfer Family 이벤트를 선택하고 관리할 때는 다음 사항을 고려하십시오.

- 의 모든 이벤트 `source` 필드는 로 Transfer Family 설정되어 `aws.transfer` 있습니다.
- `detail-type` 필드는 이벤트 유형을 지정합니다.

  예를 들어 `FTP File Server Download Completed`입니다.
- `detail` 필드는 해당 특정 이벤트와 관련된 데이터를 포함합니다.

Transfer Family 이벤트와 일치하는 규칙을 활성화하는 이벤트 패턴을 구성하는 방법에 대한 자세한 내용은 Amazon EventBridge 사용 안내서의 이벤트 패턴을 참조하세요.

이벤트 및 이벤트 EventBridge 처리 방법에 대한 자세한 내용은 Amazon EventBridge 사용 설명서의 Amazon EventBridge 이벤트를 참조하십시오.

주제

- SFTP, FTPS 및 FTP 서버 이벤트
- SFTP 커넥터 이벤트
- AS2 이벤트

SFTP, FTPS 및 FTP 서버 이벤트

SFTP, FTPS 및 FTP 서버 이벤트의 세부 정보 필드는 다음과 같습니다.

- FTP 파일 서버 다운로드 완료
- FTP 파일 서버 다운로드 실패
- FTP 파일 서버 업로드 완료
- FTP 파일 서버 업로드 실패
- FTPS 파일 서버 다운로드 완료
- FTPS 파일 서버 다운로드 실패
- FTPS 파일 서버 업로드 완료
- FTPS 파일 서버 업로드 실패

- SFTP 서버 파일 다운로드 완료
- SFTP 서버 파일 다운로드 실패
- SFTP 서버 파일 업로드가 완료되었습니다.
- SFTP 서버 파일 업로드 실패

source및 detail-type 필드는 Transfer Family 이벤트에 대한 특정 값을 포함하므로 아래에 포함됩니다. 모든 이벤트에 포함되는 다른 메타데이터 필드의 정의는 Amazon EventBridge 사용 설명서의 이벤트 구조 참조를 참조하십시오.

```
{
  . . .,
  "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`

이벤트의 유형을 식별합니다.

이 이벤트의 값은 이전에 나열된 SFTP, FTPS 또는 FTP 서버 이벤트 이름 중 하나입니다.

`source`

이벤트를 생성한 서비스를 식별합니다. Transfer Family 이벤트의 경우 이 값은 입니다`aws.transfer`.

`detail`

이벤트에 대한 정보를 포함하는 JSON 객체입니다. 이벤트를 생성하는 서비스에 따라 이 필드의 내용이 결정됩니다.

이 이벤트의 데이터에는 다음이 포함됩니다.

`failure-code`

전송이 실패한 이유에 대한 카테고리. 값: `PARTIAL_UPLOAD` | `PARTIAL_DOWNLOAD` | `UNKNOWN_ERROR`

`status-code`

전송 성공 여부. 값:`COMPLETED` | `FAILED`.

`protocol`

전송에 사용된 프로토콜. 값: `SFTP` | `FTPS` | `FTP`

`bytes`

전송되는 바이트 수입니다.

`client-ip`

전송에 관련된 클라이언트의 IP 주소

`failure-message`

전송이 실패한 경우 전송이 실패한 이유에 대한 세부 정보

`end-timestamp`

성공적인 전송의 경우 파일 처리가 완료되는 시점의 타임스탬프입니다.

`etag`

엔티티 태그 (Amazon S3 파일에만 사용됨).

`file-path`

전송 중인 파일의 경로.

`server-id`

Transfer Family 서버의 고유 ID입니다.

`username`

전송을 수행하는 사용자.

`session-id`

전송 세션의 고유 식별자입니다.

`start-timestamp`

성공적인 전송의 경우 파일 처리가 시작되는 시점의 타임스탬프입니다.

Example SFTP 서버 파일 다운로드 실패 예제 이벤트

다음 예는 SFTP 서버 (Amazon EFS 사용 중인 스토리지) 에서 다운로드가 실패한 이벤트를 보여줍니다.

```
{
    "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 파일 서버 업로드 완료 예제 이벤트

다음 예제는 FTP 서버 (Amazon S3 사용 중인 스토리지) 에서 업로드가 성공적으로 완료된 이벤트를 보여줍니다.


```
{
    "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 커넥터 이벤트

SFTP 커넥터 이벤트의 세부 정보 필드는 다음과 같습니다.

- SFTP 커넥터 파일 전송 완료
- SFTP 커넥터 파일 전송 실패
- SFTP 커넥터 파일 검색이 완료되었습니다.
- SFTP 커넥터 파일 검색 실패
- SFTP 커넥터 디렉토리 목록 작성 완료

- SFTP 커넥터 디렉토리 목록 실패

`source`및 `detail-type` 필드는 Transfer Family 이벤트에 대한 특정 값을 포함하므로 아래에 포함됩니다. 모든 이벤트에 포함되는 다른 메타데이터 필드의 정의는 Amazon EventBridge 사용 설명서의 이벤트 구조 참조를 참조하십시오.

```
{
  . . .,
  "detail-type": "string",
  "source": "aws.transfer",
  . . .,
  "detail": {
    "operation" : "string",
    "max-items" : "number",
    "connector-id" : "string",
    "output-directory-path" : "string",
    "listing-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"
    "item-count" : "number"
    "truncated" : "boolean"
    "bytes" : "number",
    "local-file-location" : {
      "domain" : "string",
      "bucket" : "string",
      "key" : "string"
    },
    "output-file-location" : {
      "domain" : "string",
      "bucket" : "string",
      "key" : "string"
    }
  }
```

```
}
```

`detail-type`

이벤트의 유형을 식별합니다.

이 이벤트의 값은 이전에 나열된 SFTP 커넥터 이벤트 이름 중 하나입니다.

`source`

이벤트를 생성한 서비스를 식별합니다. Transfer Family 이벤트의 경우 이 값은 입니다. `aws.transfer`

`detail`

이벤트에 대한 정보를 포함하는 JSON 객체입니다. 이벤트를 생성하는 서비스에 따라 이 필드의 내용이 결정됩니다.

이 이벤트의 데이터에는 다음이 포함됩니다.

`max-items`

반환할 디렉터리/파일 이름의 최대 수입니다.

`operation`

`StartFileTransfer`요청이 파일을 보내는 것인지 아니면 가져오는 것인지 여부. 값:. `SEND|RETRIEVE`

`connector-id`

사용 중인 SFTP 커넥터의 고유 식별자입니다.

`output-directory-path`

파일/디렉터리 목록 결과를 저장하는 Amazon S3의 경로 (버킷 및 접두사).

`listing-id`

API 호출의 고유 식별자. `StartDirectoryListing` 이 식별자를 사용하여 CloudWatch 로그를 확인하여 리스팅 요청 상태를 확인할 수 있습니다.

`transfer-id`

전송 이벤트 (`StartFileTransfer`요청) 의 고유 식별자입니다.

`file-transfer-id`

전송 중인 파일의 고유 식별자입니다.

`url`

파트너의 AS2 또는 SFTP 엔드포인트의 URL입니다.

`file-path`

전송 또는 검색 중인 위치 및 파일.

`status-code`

전송 성공 여부. 값:`FAILED | COMPLETED`.

`failure-code`

전송이 실패한 경우 전송이 실패한 사유 코드입니다.

`failure-message`

전송이 실패한 경우 전송이 실패한 이유에 대한 세부 정보

`start-timestamp`

성공적인 전송의 경우 파일 처리가 시작되는 시점의 타임스탬프입니다.

`end-timestamp`

성공적인 전송의 경우 파일 처리가 완료되는 시점의 타임스탬프입니다.

`local-directory-path`

`RETRIEVE`요청의 경우, 검색된 파일을 배치할 위치.

`remote-directory-path`

`SEND`요청의 경우 파트너의 SFTP 서버에 파일을 배치할 파일 디렉토리입니다. 사용자가 `StartFileTransfer` 요청에 `RemoteDirectoryPath` 전달한 값의 값입니다. 파트너의 SFTP 서버에서 기본 디렉터리를 지정할 수 있습니다. 그렇다면 이 필드는 비어 있습니다.

`item-count`

리스팅 요청으로 반환된 항목 (디렉터리 및 파일) 수.

`truncated`

목록 출력에 원격 디렉터리에 포함된 모든 항목이 포함되는지 여부

`bytes`

전송 중인 바이트 수입니다. 전송 실패의 값은 0입니다.

`local-file-location`

이 매개변수에는 AWS 저장소 파일 위치의 세부 정보가 포함됩니다.

`domain`

사용 중인 스토리지. 현재 유일한 값은 입니다S3.

`bucket`

Amazon S3에 있는 객체를 위한 컨테이너입니다.

`key`

Amazon S3에 있는 객체에 할당된 이름입니다.

`output-file-location`

이 파라미터는 디렉토리 목록 결과를 AWS 스토리지에 저장할 위치의 세부 정보를 포함합니다.

`domain`

사용 중인 스토리지. 현재 유일한 값은 입니다S3.

`bucket`

Amazon S3에 있는 객체를 위한 컨테이너입니다.

`key`

Amazon S3에 있는 객체에 할당된 이름입니다.

Example SFTP 커넥터 파일 전송 실패 예제 이벤트

다음 예에서는 원격 SFTP 서버로 파일을 보내려고 시도하는 동안 SFTP 커넥터가 실패한 이벤트를 보여 줍니다.

```
{
    "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 커넥터 파일 검색 완료 예제 이벤트

다음 예에서는 SFTP 커넥터가 원격 SFTP 서버에서 전송된 파일을 성공적으로 검색한 이벤트를 보여 줍니다.

```
{
    "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"
        }
    }
}
```

Example SFTP 커넥터 디렉터리 목록 완료 예제 이벤트

다음 예에서는 디렉토리 목록 시작 호출이 원격 SFTP 서버에서 목록 파일을 검색한 이벤트를 보여 줍니다.

```
{
    "version": "0",
    "id": "event-ID",
    "detail-type": "SFTP Connector Directory Listing 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": {
        "max-items": 10000,
        "connector-id": "c-fc68000012345aa18",
        "output-directory-path": "/DOC-EXAMPLE-BUCKET/example/file-listing-output",
        "listing-id": "123456-23aa-7980-abc1-1a2b3c4d5e",
        "url": "sftp://s-21a23456789012a.server.transfer.us-east-1.amazonaws.com",

        "status-code": "COMPLETED",
        "remote-directory-path": "/home",
        "item-count": 10000,
```

```
        "truncated": true,
        "start-timestamp": "2024-01-24T18:28:07.632388Z",
        "end-timestamp": "2024-01-24T18:28:07.774898Z",
        "output-file-location": {
            "domain": "S3",
            "bucket": "DOC-EXAMPLE-BUCKET",
            "key": "c-fc1ab90fd0d047e7a-70987273-49nn-4006-bab1-1a7290cc412ba.json"
        }
    }
}
```

AS2 이벤트

AS2 이벤트의 세부 정보 필드는 다음과 같습니다.

- AS2 페이로드 수신 완료
- AS2 페이로드 수신 실패
- AS2 페이로드 전송 완료
- AS2 페이로드 전송 실패
- AS2 MDN 수신 완료
- AS2 MDN 수신이 실패했습니다.
- AS2 MDN 전송 완료
- AS2 MDN 전송 실패

`source`및 `detail-type` 필드는 이벤트에 대한 Transfer Family 특정 값을 포함하므로 아래에 포함됩니다. 모든 이벤트에 포함되는 다른 메타데이터 필드의 정의는 Amazon EventBridge 사용 설명서의 이벤트 구조 참조를 참조하십시오.

```
{
   . . .,
  "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`

이벤트의 유형을 식별합니다.

이 이벤트의 값은 앞서 나열한 AS2 이벤트 중 하나입니다.

`source`

이벤트를 생성한 서비스를 식별합니다. Transfer Family 이벤트의 경우 이 값은 입니다.

`aws.transfer`

`detail`

이벤트에 대한 정보를 포함하는 JSON 객체입니다. 이벤트를 생성하는 서비스에 따라 이 필드의 내용이 결정됩니다.

`s3-attributes`

전송 중인 파일의 Amazon S3 버킷과 키를 식별합니다. MDN 이벤트의 경우 MDN 파일의 버킷과 키도 식별합니다.

`file-bucket`

Amazon S3에 있는 객체를 위한 컨테이너입니다.

`file-key`

Amazon S3에 있는 객체에 할당된 이름입니다.

`json-bucket`

전송이 완료되거나 실패한 경우 JSON 파일의 컨테이너입니다.

`json-key`

전송이 완료되거나 실패한 경우 Amazon S3의 JSON 파일에 할당된 이름입니다.

`mdn-bucket`

MDN 이벤트의 경우 MDN 파일을 위한 컨테이너입니다.

`mdn-key`

MDN 이벤트의 경우, Amazon S3의 MDN 파일에 할당된 이름입니다.

`mdn-subject`

MDN 이벤트의 경우, 메시지 처리에 대한 텍스트 설명.

`mdn-message-id`

MDN 이벤트의 경우 MDN 메시지의 고유 ID.

`disposition`

MDN 이벤트의 경우 처리를 위한 카테고리입니다.

`bytes`

메시지의 바이트 수.

`as2-from`

메시지를 보내는 AS2 거래 파트너.

`as2-message-id`

전송 중인 AS2 메시지의 고유 식별자.

`as2-to`

메시지를 받고 있는 AS2 거래 파트너.

`connector-id`

Transfer Family 서버에서 거래 파트너로 보내는 AS2 메시지의 경우 AS2 커넥터의 고유 식별자가 사용됩니다.

`client-ip`

서버 이벤트 (거래 파트너에서 Transfer Family 서버로 이전) 의 경우 이전과 관련된 클라이언트의 IP 주소.

`agreement-id`

서버 이벤트의 경우 AS2 계약의 고유 식별자입니다.

`server-id`

서버 이벤트의 경우 Transfer Family 서버에만 사용할 수 있는 고유 ID입니다.

`requester-file-name`

페이로드 이벤트의 경우 전송 중에 받은 파일의 원래 이름입니다.

`message-subject`

메시지 제목에 대한 텍스트 설명.

`start-timestamp`

성공적인 전송의 경우 파일 처리가 시작되는 시점의 타임스탬프입니다.

`end-timestamp`

성공적인 전송의 경우 파일 처리가 완료되는 시점의 타임스탬프입니다.

`status-code`

AS2 메시지 전송 프로세스의 상태에 해당하는 코드입니다. 유효한 값: `COMPLETED | FAILED | PROCESSING`.

`failure-code`

실패한 전송의 경우 전송이 실패한 이유에 대한 범주입니다.

`failure-message`

전송이 실패한 경우 전송이 실패한 이유에 대한 세부 정보.

`transfer-id`

전송 이벤트의 고유 식별자입니다.

Example AS2 페이로드 수신 완료 예제 이벤트

```
{
    "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 수신 실패 예제 이벤트

```
{
  "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"
  }
}
```

보안 입력 AWS Transfer Family

클라우드 AWS 보안이 최우선 과제입니다. AWS 고객은 가장 보안에 민감한 조직의 요구 사항을 충족하도록 구축된 데이터 센터 및 네트워크 아키텍처의 혜택을 누릴 수 있습니다.

보안은 기업과 기업 간의 AWS 공동 책임입니다. 공동 책임 모델은 이 사항을 클라우드 내 보안 및 클라우드의 보안으로 설명합니다.

특정 규정 준수 프로그램의 범위 내에 AWS 서비스 있는지 알아보려면AWS 서비스 규정 준수 프로그램의AWS 서비스 범위별, 규정 참조하여 관심 있는 규정 준수 프로그램을 선택하십시오. 일반 정보는 AWS 규정 준수 프로그램AWS 보증 프로그램 규정AWS 참조하십시오.

를 사용하여 AWS Artifact타사 감사 보고서를 다운로드할 수 있습니다. 자세한 내용은 의 보고서 https://docs.aws.amazon.com/artifact/latest/ug/downloading-documents.html 참조하십시오 AWS Artifact.

사용 시 규정 준수 AWS 서비스 책임은 데이터의 민감도, 회사의 규정 준수 목표, 관련 법률 및 규정에 따라 결정됩니다. AWS 규정 준수에 도움이 되는 다음 리소스를 제공합니다.

- 보안 및 규정 준수 퀵 스타트 가이드 - 이 배포 가이드에서는 아키텍처 고려 사항을 설명하고 보안 및 규정 준수에 AWS 중점을 둔 기본 환경을 배포하기 위한 단계를 제공합니다.
- Amazon Web Services의 HIPAA 보안 및 규정 준수를 위한 설계 — 이 백서에서는 기업이 HIPAA 적격 애플리케이션을 만드는 AWS 데 사용할 수 있는 방법을 설명합니다.

  > **Note**
  >
  > 모든 AWS 서비스 사람이 HIPAA 자격을 갖춘 것은 아닙니다. 자세한 내용은 HIPAA 적격 서비스 참조를 참조하십시오.

- AWS 규정 준수 리소스AWS — 이 워크북 및 가이드 모음은 해당 산업 및 지역에 적용될 수 있습니다.
- AWS 고객 규정 준수 가이드 — 규정 준수의 관점에서 공동 책임 모델을 이해하십시오. 이 가이드에서는 보안을 유지하기 위한 모범 사례를 AWS 서비스 요약하고 여러 프레임워크 (미국 표준 기술 연구소 (NIST), 결제 카드 산업 보안 표준 위원회 (PCI), 국제 표준화기구 (ISO) 등) 에서 보안 제어에 대한 지침을 매핑합니다.
- AWS Config 개발자 안내서의 규칙을 사용하여 리소스 평가 — 이 AWS Config 서비스는 리소스 구성이 내부 관행, 업계 지침 및 규정을 얼마나 잘 준수하는지 평가합니다.

- AWS Security Hub— 이를 AWS 서비스 통해 내부 AWS보안 상태를 포괄적으로 파악할 수 있습니다. Security Hub는 보안 제어를 사용하여 AWS 리소스를 평가하고 보안 업계 표준 및 모범 사례에 대한 규정 준수를 확인합니다. 지원되는 서비스 및 제어 목록은 Security Hub 제어 참조를 참조하십시오.
- Amazon GuardDuty — 환경에 의심스럽고 악의적인 활동이 있는지 AWS 계정모니터링하여 워크로드, 컨테이너 및 데이터에 대한 잠재적 위협을 AWS 서비스 탐지합니다. GuardDuty 특정 규정 준수 프레임워크에서 요구하는 침입 탐지 요구 사항을 충족하여 PCI DSS와 같은 다양한 규정 준수 요구 사항을 해결하는 데 도움이 될 수 있습니다.
- AWS Audit Manager— 이를 AWS 서비스 통해 AWS 사용량을 지속적으로 감사하여 위험을 관리하고 규정 및 업계 표준을 준수하는 방법을 단순화할 수 있습니다.

이 설명서는 공동 책임 모델을 사용할 때 공동 책임 모델을 적용하는 방법을 이해하는 데 도움이 AWS Transfer Family됩니다. 다음 항목에서는 보안 및 규정 준수 목표를 AWS Transfer Family 충족하도록 구성하는 방법을 보여줍니다. 또한 AWS Transfer Family 리소스를 모니터링하고 보호하는 데 도움이 되는 다른 AWS 서비스를 사용하는 방법도 알아봅니다.

기존 애플리케이션을 수정하거나 서버 인프라를 관리할 필요 AWS 없이 확장 가능하고 안전한 파일 전송 아키텍처를 구축할 수 있는 방법에 대한 지침과 실습을 제공하는 워크숍을 제공합니다. 이 워크숍의 세부 정보는 여기에서 확인할 수 있습니다.

주제

- 서버 보안 정책 AWS Transfer Family
- SFTP AWS Transfer Family 커넥터에 대한 보안 정책
- AWS Transfer Family에서 하이브리드 포스트 퀀텀 키 교환 사용
- 데이터 보호: AWS Transfer Family
- 에 대한 ID 및 액세스 관리 AWS Transfer Family
- 규정 준수 검증: AWS Transfer Family
- 레질리언스: AWS Transfer Family
- 인프라 보안: AWS Transfer Family
- 웹 애플리케이션 방화벽 추가
- 교차 서비스 혼동된 대리인 방지
- AWSAWS Transfer Family에 대한 관리형 정책

서버 보안 정책 AWS Transfer Family

의 서버 보안 정책을 AWS Transfer Family 통해 서버와 관련된 일련의 암호화 알고리즘 (메시지 인증 코드 (MAC), 키 교환 (KEX) 및 암호 그룹) 을 제한할 수 있습니다. 지원되는 키 알고리즘 목록은 암호화 알고리즘 섹션을 참조하세요. 서버 호스트 키 및 서비스 관리 사용자 키와 함께 사용할 수 있는 지원되는 키 알고리즘 목록은 사용자 및 서버 키에 지원되는 알고리즘를 참조하세요.

> **Note**
>
> 서버를 최신 보안 정책으로 업데이트하는 것이 좋습니다. 최신 보안 정책이 기본값입니다. 기본 보안 정책을 사용하여 Transfer Family 서버를 CloudFormation 생성하고 이를 수락하는 모든 고객에게는 자동으로 최신 정책이 할당됩니다. 클라이언트 호환성이 우려되는 경우 서버를 만들거나 업데이트할 때 변경될 수 있는 기본 정책을 사용하는 대신 사용할 보안 정책을 확실하게 명시하십시오.
>
> 서버의 보안 정책을 변경하려면 을 참조하십시오. 보안 정책 편집

Transfer Family의 보안에 대한 자세한 내용은 블로그 게시물인 Transfer Family를 통해 안전하고 규정을 준수하는 관리형 파일 전송 솔루션을 구축하는 방법을 참조하세요.

주제

- 암호화 알고리즘
- TransferSecurity정책-2024-01
- TransferSecurity정책-2023-05
- TransferSecurity정책-2022-03
- TransferSecurity정책-2020-06
- TransferSecurity정책-2018-11
- TransferSecurityTransferSecurity정책-FIPS-2024-01/ 정책-FIPS-2024-05
- TransferSecurity정책-FIPS-2023-05
- TransferSecurity정책-FIPS-2020-06
- 포스트 퀀텀 보안 정책

> **Note**
>
> `TransferSecurityPolicy-2024-01`콘솔, API 또는 CLI를 사용하여 서버를 생성할 때 서버에 연결되는 기본 보안 정책입니다.

암호화 알고리즘

호스트 키의 경우 다음 알고리즘을 지원합니다.

- `rsa-sha2-256`
- `rsa-sha2-512`
- `ecdsa-sha2-nistp256`
- `ecdsa-sha2-nistp384`
- `ecdsa-sha2-nistp521`
- `ssh-ed25519`

또한 다음 보안 정책은 `ssh-rsa` 다음을 허용합니다.

- TransferSecurity정책-2018-11
- TransferSecurity정책-2020-06
- TransferSecurity정책-FIPS-2020-06
- TransferSecurity정책-FIPS-2023-05
- TransferSecurity정책-FIPS-2024-01
- TransferSecurity정책-PQ-SSH-FIPS-실험용-2023-04

> **Note**
>
> RSA 키 유형 (항상 사용 가능) 과 RSA 호스트 키 알고리즘 (지원되는 알고리즘 중 하나일 수 있음) 의 차이점을 이해하는 것이 중요합니다. `ssh-rsa`

다음은 각 보안 정책에 지원되는 암호화 알고리즘 목록입니다.

> **Note**
>
> 다음 표와 정책에서 다음과 같은 알고리즘 유형 사용에 주목하십시오.
>
> - SFTP 서버는 SshCiphersSshKexs, 및 SshMacs섹션에서만 알고리즘을 사용합니다.
> - FTPS 서버는 섹션의 TlsCiphers알고리즘만 사용합니다.
> - FTP 서버는 암호화를 사용하지 않으므로 이러한 알고리즘을 사용하지 마십시오.
> - FIPS-2024-05 및 FIPS-2024-01 보안 정책은 FIPS-2024-05 이 `ssh-rsa` 알고리즘을 지원하지 않는다는 점을 제외하면 동일합니다.

보안 정책	2024-01	2023-05	2022-03	2020-06	FIPS-2024-05 FIPS-2024-01	FIPS-2023-05	FIPS-2020-06	2018-11
SshCiphers								
aes128-ctr	♦			♦	♦		♦	♦
aes128-gcm@openssh.com	♦	♦	♦	♦	♦	♦	♦	♦
aes192-ctr	♦	♦	♦	♦	♦	♦	♦	♦
aes256-ctr	♦	♦	♦	♦	♦	♦	♦	♦
aes256-gcm@openssh.com	♦	♦	♦	♦	♦	♦	♦	♦

보안 정책	2024-01	2023-05	2022-03	2020-06	FIPS-2024-05 FIPS-2024-01	FIPS-2023-05	FIPS-2020-06	2018-11
chacha20-poly1305@openssh.com				♦				♦
SshKexs								
curve25519-sha256	♦	♦	♦					♦
curve25519-sha256@libssh.org	♦	♦	♦					♦
diffie-hellman-group14-sha1								♦
diffie-hellman-group14-sha256				♦			♦	♦

보안 정책	2024-01	2023-05	2022-03	2020-06	FIPS-2024-05 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	♦				♦			

보안 정책	2024-01	2023-05	2022-03	2020-06	FIPS-2024-05 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	♦			♦	♦		♦	♦

보안 정책	2024-01	2023-05	2022-03	2020-06	FIPS-2024-05 FIPS-2024-01	FIPS-2023-05	FIPS-2020-06	2018-11
ecdh-sha2-nistp521	♦			♦	♦		♦	♦
x25519-kyber-512r3-sha256-d00@amazon.com	♦							
SshMacs								
hmac-sha1								♦
hmac-sha1-etm@openssh.com								♦
hmac-sha2-256			♦	♦			♦	♦

보안 정책	2024-01	2023-05	2022-03	2020-06	FIPS-2024-05 FIPS-2024-01	FIPS-2023-05	FIPS-2020-06	2018-11
hmac-sha2-256-etm@openssh.com	♦	♦	♦	♦	♦	♦	♦	♦
hmac-sha2-512			♦	♦			♦	♦
hmac-sha2-512-etm@openssh.com	♦	♦	♦	♦	♦	♦	♦	♦
umac-128-etm@openssh.com				♦				♦
umac-128@openssh.com				♦				♦
umac-64-etm@openssh.com								♦

보안 정책	2024-01	2023-05	2022-03	2020-06	FIPS-2024-05 FIPS-2024-01	FIPS-2023-05	FIPS-2020-06	2018-11
umac-64@openssh.com								♦
TlsCiphers								
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		♦	♦	♦	♦	♦	♦	♦

보안 정책	2024-01	2023-05	2022-03	2020-06	FIPS-2024-05 FIPS-2024-01	FIPS-2023-05	FIPS-2020-06	2018-11
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	♦	♦	♦	♦	♦	♦	♦	♦
TLS_RSA_WITH_AES_128_CBC_SHA256								♦

보안 정책	2024-01	2023-05	2022-03	2020-06	FIPS-2024-05 FIPS-2024-01	FIPS-2023-05	FIPS-2020-06	2018-11
TLS_RSA_WITH_AES_256_CBC_SHA256								◆

TransferSecurity정책-2024-01

다음은 TransferSecurityPolicy -2024-01 보안 정책을 보여줍니다.

```
{
    "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"
        ]
    }
}
```

TransferSecurity정책-2023-05

다음은 TransferSecurityPolicy -2023-05 보안 정책을 보여줍니다.

```
{
    "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"
        ]
    }
}
```

TransferSecurity정책-2022-03

다음은 TransferSecurityPolicy -2022-03 보안 정책을 보여줍니다.

```
{
  "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"
    ]
  }
}
```

TransferSecurity정책-2020-06

다음은 TransferSecurityPolicy -2020-06 보안 정책을 보여줍니다.

```
{
  "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"
    ]
  }
}
```

TransferSecurity정책-2018-11

다음은 TransferSecurityPolicy -2018-11 보안 정책을 보여줍니다.

```
{
  "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"
    ]
  }
}
```

TransferSecurityTransferSecurity정책-FIPS-2024-01/ 정책-FIPS-2024-05

다음은 -FIPS-2024-01 및 -FIPS-2024-05 보안 정책을 보여줍니다. TransferSecurityPolicy TransferSecurityPolicy

> **Note**
>
> FIPS 서비스 엔드포인트 및 -FIPS-2024-01 및 -FIPS-2024-05 보안 정책은 일부 지역에서만 사용할 수 있습니다. TransferSecurityPolicy TransferSecurityPolicy AWS 자세한 내용은AWS 일반 참조에서 AWS Transfer Family 엔드포인트 및 할당량을 참조하세요.
> 이 두 보안 정책의 유일한 차이점은 -FIPS-2024-01은 알고리즘을 지원하지만 -FIPS-2024-05는 지원하지 않는다는 것입니다. TransferSecurityPolicy `ssh-rsa` TransferSecurityPolicy

```
{
    "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"
        ]
    }
}
```

TransferSecurity정책-FIPS-2023-05

FIPS 인증 세부 정보는 다음에서 확인할 수 있습니다. AWS Transfer Family https://csrc.nist.gov/projects/cryptographic-module-validation-program/validated-modules/search/all

다음은 TransferSecurityPolicy -FIPS-2023-05 보안 정책을 보여줍니다.

> **Note**
>
> FIPS 서비스 엔드포인트 및 TransferSecurityPolicy -FIPS-2023-05 보안 정책은 일부 지역에서만 사용할 수 있습니다. AWS 자세한 내용은AWS 일반 참조에서 AWS Transfer Family 엔드포인트 및 할당량을 참조하세요.

```
{
    "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"
        ]
    }
}
```

TransferSecurity정책-FIPS-2020-06

FIPS 인증 세부 정보는 다음에서 확인할 수 있습니다. AWS Transfer Family https://csrc.nist.gov/projects/cryptographic-module-validation-program/validated-modules/search/all

다음은 TransferSecurityPolicy -FIPS-2020-06 보안 정책을 보여줍니다.

> **Note**
>
> FIPS 서비스 엔드포인트 및 TransferSecurityPolicy -FIPS-2020-06 보안 정책은 일부 지역에서만 사용할 수 있습니다. AWS 자세한 내용은 AWS 일반 참조의 AWS Transfer Family 엔드포인트 및 할당량을 참조하세요.


```
{
  "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"
    ]
  }
}
```

포스트 퀀텀 보안 정책

이 표에는 Transfer Family 포스트 퀀텀 보안 정책의 알고리즘이 나열되어 있습니다. 이러한 정책은 AWS Transfer Family에서 하이브리드 포스트 퀀텀 키 교환 사용 에 자세히 설명되어 있습니다.

정책 목록은 표 뒤에 나와 있습니다.

보안 정책	TransferSecurity정책-PQ-SSH-실험용-2023-04	TransferSecurity정책 - PQ-SSH-FIPS-실험용 - 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	♦	♦

보안 정책	TransferSecurity정책-PQ-SSH-실험용-2023-04	TransferSecurity정책 - PQ-SSH-FIPS-실험용 - 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	♦	♦

보안 정책	TransferSecurity정책-PQ-SSH-실험용-2023-04	TransferSecurity정책 - PQ-SSH-FIPS-실험용 - 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	♦	♦

TransferSecurity정책-PQ-SSH-익스페리멘탈-2023-04

다음은 -PQ-SSH-익스페리멘탈-2023-04 보안 정책을 보여줍니다. TransferSecurityPolicy

```
{
    "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"
        ]
    }
}
```

TransferSecurity정책-PQ-SSH-FIPS-실험용-2023-04

다음은 -PQ-SSH-FIPS-익스페리멘탈-2023-04 보안 정책을 보여줍니다. TransferSecurityPolicy

```
{
```

```
    "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"
        ]
    }
}
```

SFTP AWS Transfer Family 커넥터에 대한 보안 정책

의 SFTP 커넥터 보안 정책을 AWS Transfer Family 사용하면 SFTP 커넥터와 관련된 암호화 알고리즘 세트 (메시지 인증 코드 (MAC), 키 교환 (KEX) 및 암호 그룹) 을 제한할 수 있습니다. 다음은 각 SFTP 커넥터 보안 정책에 지원되는 암호화 알고리즘 목록입니다.

> **Note**
>
> `TransferSFTPConnectorSecurityPolicy-2024-03`SFTP 커넥터에 적용되는 기본 보안 정책입니다.

커넥터의 보안 정책을 변경할 수 있습니다. Transfer Family 왼쪽 탐색 창에서 커넥터를 선택하고 커넥터를 선택합니다. 그런 다음 Sftp 구성 섹션에서 편집을 선택합니다. 암호화 알고리즘 옵션 섹션에서 보안 정책 필드의 드롭다운 목록에서 사용 가능한 보안 정책을 선택합니다.

보안 정책	TransferSFTP 정책-2024-03 ConnectorSecurity	전송 FTP ConnectorSecurity 정책-2023-07
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		♦

보안 정책	TransferSFTP 정책-2024-03 ConnectorSecurity	전송 FTP ConnectorSecurity 정책-2023-07
diffie-hellman-group16-sha512	♦	♦
diffie-hellman-group18-sha512	♦	♦
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		♦

AWS Transfer Family에서 하이브리드 포스트 퀀텀 키 교환 사용

AWS Transfer Family SSH (Secure Shell) 프로토콜을 위한 하이브리드 포스트 퀀텀 키 설정 옵션을 지원합니다. 네트워크 트래픽을 기록하고 향후 복호화를 위해 양자 컴퓨터에 저장하는 것이 이미 가능하기 때문에 포스트 퀀텀 키 설정이 필요합니다. 이를 지금 저장 나중 수확 공격이라고 합니다.

이 옵션은 Transfer Family에 연결하여 Amazon 심플 스토리지 서비스(Amazon S3) 스토리지 또는 Amazon Elastic File System(Amazon EFS)에서 안전하게 파일을 주고 받을 때 사용할 수 있습니다. SSH의 포스트 퀀텀 하이브리드 키 설정은 포스트 퀀텀 키 설정 메커니즘을 도입하여 기존 키 교환 알고리즘과 함께 사용합니다. 기존 암호군으로 생성된 SSH 키는 현재 기술로는 무차별 암호 대입 공격으로부터 안전합니다. 그러나 향후 대규모 양자 컴퓨팅이 등장하더라도 기존 암호화는 보안을 유지하지 못할 것으로 예상됩니다.

Transfer Family 연결을 통해 전달되는 데이터의 장기 기밀 유지에 의존하는 조직의 경우, 대규모 양자 컴퓨터를 사용할 수 있게 되기 전에 포스트 퀀텀 암호화로 마이그레이션하는 계획을 고려해야 합니다.

현재 암호화된 데이터를 미래의 잠재적 공격으로부터 보호하기 위해 암호화 커뮤니티와 함께 양자 내성 또는 포스트 퀀텀 알고리즘 개발에 참여하고 있습니다. AWS Transfer Family에서 클래식 및 포스트 퀀텀 요소를 결합하는 하이브리드 포스트 퀀텀 키 교환 암호 제품군을 구현했습니다.

이러한 하이브리드 암호 제품군은 대부분의 AWS 지역에서 프로덕션 워크로드에 사용할 수 있습니다. 그러나 하이브리드 암호 제품군의 성능 특성 및 대역폭 요건은 클래식 키 교환 메커니즘의 해당 요건과 다르기 때문에 Transfer FAmily 연결에 대해 이 제품군을 테스트하는 것이 좋습니다.

포스트 퀀텀 암호화 보안 블로그 게시물에서 포스트 퀀텀 암호화에 대해 자세히 알아보세요.

목차

SSH의 포스트 퀀텀 하이브리드 키 교환 소개

Transfer Family는 기존의 타원 곡선 디피-헬만(ECDH) 키 교환 알고리즘과 CRYSTALS Kyber를 모두 사용하는 포스트 퀀텀 하이브리드 키 교환 암호 제품군을 지원합니다. Kyber는 미국 국립표준기술연구소(NIST)가 최초의 표준 포스트 퀀텀 키 계약 알고리즘으로 지정한 포스트 퀀텀 퍼블릭 키 암호화 및 키 설정 알고리즘입니다.

클라이언트와 서버는 여전히 ECDH 키 교환을 수행합니다. 또한 서버는 포스트 퀀텀 공유 비밀을 클라이언트의 포스트 퀀텀 KEM 퍼블릭 키로 캡슐화하는데, 이는 클라이언트의 SSH 키 교환 메시지에 알려집니다. 이 전략은 기존 키 교환의 높은 신뢰성과 제안된 포스트 퀀텀 키 교환의 보안을 결합하여 ECDH 또는 포스트 퀀텀 공유 비밀이 침해되지 않는 한 핸드셰이크를 보호할 수 있도록 합니다.

Transfer Family에서 포스트 퀀텀 하이브리드 키 설정이 작동하는 방식

AWS 최근 SFTP 파일 전송 시 포스트 퀀텀 키 교환을 지원한다고 발표했습니다. AWS Transfer Family Transfer Family는 SFTP 및 기타 프로토콜을 사용하여 AWS 스토리지 서비스로의 business-to-business 파일 전송을 안전하게 확장합니다. SFTP는 SSH를 통해 실행되는 FTP (파일 전송 프로토콜)의 보다 안전한 버전입니다. Transfer Family의 포스트 퀀텀 키 교환 지원은 SFTP를 통한 데이터 전송의 보안 기준을 높입니다.

Transfer Family의 포스트 퀀텀 하이브리드 키 교환 SFTP 지원에는 P256, P384, P521 또는 Curve25519 곡선을 통한 ECDH와 포스트 퀀텀 알고리즘 Kyber-512, Kyber-768 및 Kyber-1024의 결합이 포함됩니다. 포스트 퀀텀 하이브리드 SSH 키 교환 초안에는 다음과 같은 해당 SSH 키 교환 방법이 명시되어 있습니다.

- `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**
>
> 이러한 새로운 키 교환 방법은 초안이 표준화를 향해 나아가거나 NIST가 Kyber 알고리즘을 승인함에 따라 변경될 수 있습니다.

Kyber를 선택하는 이유는 무엇인가?

AWS 표준화되고 상호 운용 가능한 알고리즘을 지원하기 위해 최선을 다하고 있습니다. Kyber는 NIST 포스트 퀀텀 암호화 프로젝트에서 표준화 대상으로 선택한 최초의 포스트 퀀텀 암호화 알고리즘입니다. 일부 표준 기관에서는 이미 Kyber를 프로토콜에 통합하고 있습니다. AWS 이미 일부 API 엔드포인트에서 TLS에서 Kyber를 지원하고 있습니다. AWS

이러한 노력의 일환으로 저는 Kyber와 AWS NIST가 승인한 SSH용 P256과 같은 곡선을 결합하는 포스트 퀀텀 암호화에 대한 제안서 초안을 IETF에 제출했습니다. 고객의 보안을 강화하기 위해 SFTP와 SSH에서 포스트 퀀텀 키 교환을 AWS 구현하는 것도 이 초안을 따릅니다. 우리의 제안이 IETF에서 채택되어 표준이 될 때까지 향후 업데이트를 지원할 계획입니다.

새로운 키 교환 방법(섹션 Transfer Family에서 포스트 퀀텀 하이브리드 키 설정이 작동하는 방식에 열거됨)은 초안이 표준화를 향해 나아가거나 NIST가 Kyber 알고리즘을 승인함에 따라 변경될 수 있습니다.

> **Note**
>
> 포스트 퀀텀 알고리즘 지원은 현재 TLS의 포스트 퀀텀 하이브리드 키 교환 AWS KMS (하이브리드 포스트 퀀텀 TLS 사용 참조) 및 API 엔드포인트에 사용할 수 있습니다. AWS KMSAWS Certificate Manager AWS Secrets Manager

포스트 퀀텀 하이브리드 SSH 키 교환 및 암호화 요건 (FIPS 140)

FIPS 규정 준수가 필요한 고객을 위해 Transfer Family는 FIPS 140 인증을 받은 오픈 소스 암호화 라이브러리인 -LC를 사용하여 SSH로 AWS FIPS 승인 암호화를 제공합니다. AWS TransferSecurityPolicyTransfer Family의 -PQ-SSH-FIPS-Expermental-2023-04에서 지원되는 포스트 퀀텀 하이브리드 키 교환 방법은 NIST의 SP 800-56Cr2 (섹션 2) 에 따라 FIPS 승인을 받았습니다. 독일 연방 정보보안청(BSI)과 프랑스 국립정보시스템청(ANSSI)에서도 이러한 포스트 퀀텀 하이브리드 키 교환 방법을 권장합니다.

Transfer Family의 포스트 퀀텀 하이브리드 키 교환 테스트

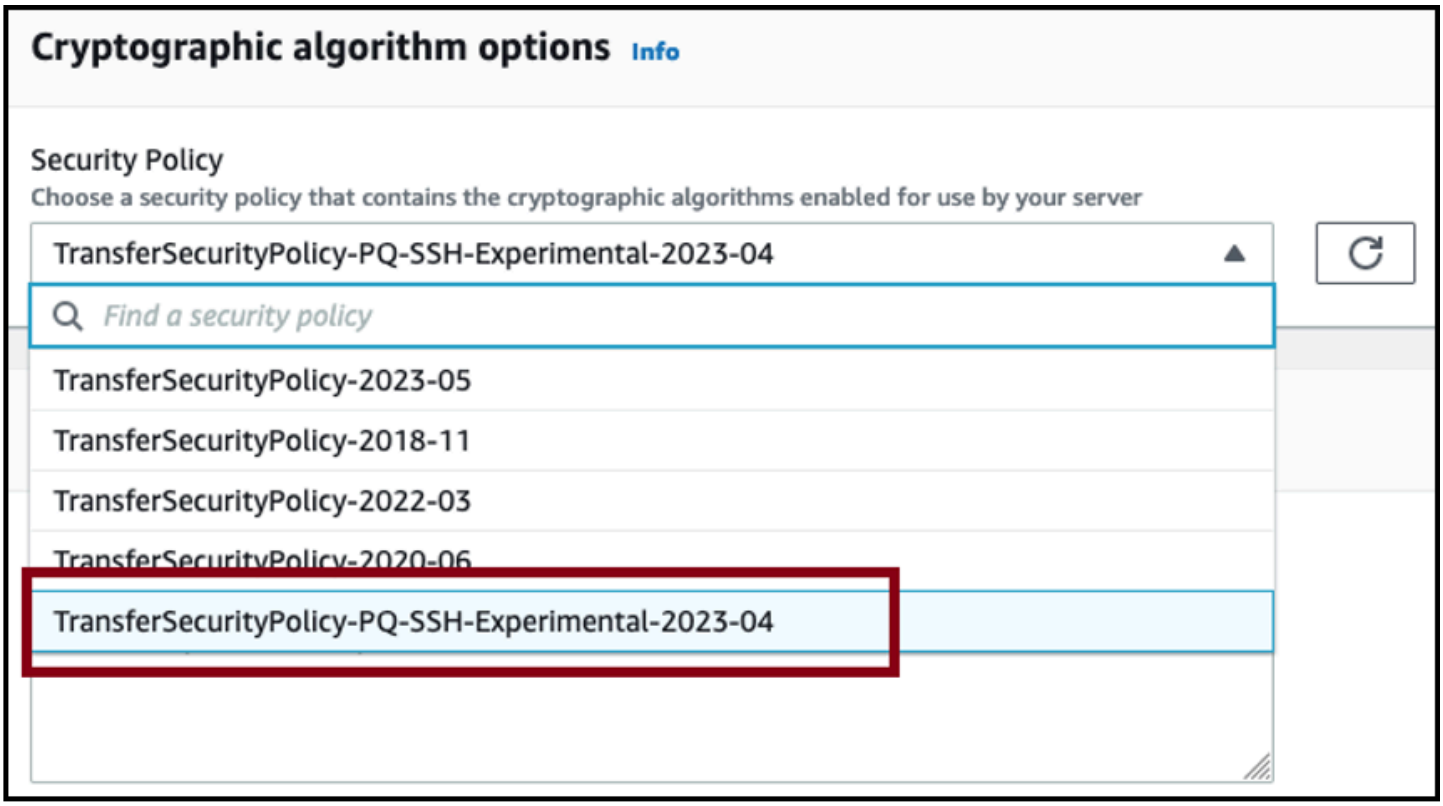
이 섹션에서는 포스트 퀀텀 하이브리드 키 교환을 테스트하기 위해 수행하는 단계를 설명합니다.

1. SFTP 엔드포인트에서 포스트 퀀텀 하이브리드 키 교환을 활성화하세요..
2. 앞서 언급한 초안 사양의 지침을 따라 포스트 퀀텀 하이브리드 키 교환을 지원하는 SFTP 클라이언트(예:포스트 퀀텀 하이브리드 키 교환을 지원하는 SFTP 클라이언트 설정)를 사용하세요.

3. Transfer Family 서버를 사용하여 파일을 전송합니다.
4. SFTP에서 포스트 퀀텀 하이브리드 키 교환 확인.

SFTP 엔드포인트에서 포스트 퀀텀 하이브리드 키 교환을 활성화하세요.

Transfer Family에서 새 SFTP 서버 엔드포인트를 생성할 때 또는 기존 SFTP 엔드포인트에서 암호화 알고리즘 옵션을 편집하여 SSH 정책을 선택할 수 있습니다. 다음 스냅샷은 SSH 정책을 업데이트하는 AWS Management Console 경우의 예를 보여줍니다.



포스트 퀀텀 키 교환을 지원하는 SSH 정책 이름은 TransferSecurity정책-PQ-SSH-익스페리멘탈-2023-04 및 TransferSecurity정책-PQ-SSH-FIPS-익스페리멘탈-2023-04입니다. Transfer Family 정책에 대한 자세한 설명은 서버 보안 정책 AWS Transfer Family를 참조하세요.

포스트 퀀텀 하이브리드 키 교환을 지원하는 SFTP 클라이언트 설정

SFTP Transfer Family 엔드포인트에서 올바른 포스트 퀀텀 SSH 정책을 선택한 후 Transfer Family에서 포스트 퀀텀 SFTP를 실험해 볼 수 있습니다. 앞서 언급한 초안 사양의 지침에 따라 포스트 퀀텀 하이브리드 키 교환을 지원하는 SFTP 클라이언트(예: OQS OpenSSH)를 사용할 수 있습니다.

OQS OpenSSH는 liboqs를 사용하여 SSH에 양자 안전 암호화를 추가하는 OpenSSH의 오픈 소스 포크입니다. liboqs은(는) 양자 저항성 암호화 알고리즘을 구현하는 오픈 소스 C 라이브러리입니다. OQS OpenSSH 및 liboqs은(는) 오픈 퀀텀 세이프 (OQS) 프로젝트의 일부입니다.

OQS OpenSSH를 사용하여 Transfer Family SFTP에서 포스트 퀀텀 하이브리드 키 교환을 테스트하려면 프로젝트의 README에 설명된 대로 OQS OpenSSH를 빌드해야 합니다. OQS OpenSSH를 구축한 후에는 다음 명령과 같이 포스트 퀀텀 하이브리드 키 교환 방법을 사용하여 예 SFTP 클라이언트를 실행하여 SFTP 엔드포인트(예: s-1111aaaa2222bbbb3.server.transfer.us-west-2.amazonaws.com)에 연결할 수 있습니다.

```
./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
```

이전 명령에서 다음 항목을 자신의 정보로 바꿉니다.

- *username_private_key_PEM_file*을 SFTP 사용자의 프라이빗 키 PEM 인코딩 파일로 대체합니다.
- *### ##*을 인스턴스의 사용자 이름으로 바꿉니다.
- *## ID*를 Transfer Family 서버 ID로 교체
- *## ID*를 Transfer Family 서버가 위치한 실제 지역으로 바꾸세요.

SFTP에서 포스트 퀀텀 하이브리드 키 교환 확인

SFTP를 Transfer Family로 연결하는 동안 포스트 퀀텀 하이브리드 키 교환이 사용되었는지 확인하려면 클라이언트 출력을 확인하세요. 선택적으로 패킷 캡처 프로그램을 사용할 수 있습니다. Open Quantum Safe OpenSSH 클라이언트를 사용하는 경우 출력은 다음과 비슷해야 합니다 (간결성을 위해 관련 없는 정보는 생략).

```
$./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>
```

출력은 포스트 퀀텀 하이브리드 ecdh-nistp384-kyber-768r3-sha384-d00@openquantumsafe.org 방법을 사용하여 클라이언트 협상이 이루어졌으며 SFTP 세션이 성공적으로 설정되었음을 보여줍니다.

데이터 보호: AWS Transfer Family

AWS 공동 책임 모델 공동 책임 모델 이 모델에 설명된 대로 AWS 는 모든 AWS 클라우드를 실행하는 글로벌 인프라를 보호할 책임이 있습니다. 사용자는 인프라에서 호스팅되는 콘텐츠를 관리해야 합니

다. 이 콘텐츠에는 사용하는 AWS 서비스의 보안 구성 및 관리 작업이 포함됩니다. 데이터 프라이버시에 대한 자세한 설명은 데이터 프라이버시 FAQ를 참조하세요. 유럽의 데이터 보호에 대한 자세한 설명은 AWS 보안 블로그의 AWS 공동 책임 모델 및 GDPR 블로그 게시물을 참조하세요.

데이터 보호를 위해 AWS 계정 자격 증명을 보호하고 AWS Identity and Access Management (IAM) 을 사용하여 개별 사용자 계정을 설정하는 것이 좋습니다. 이러한 방식에서는 각 사용자에게 자신의 직무를 충실히 이행하는 데 필요한 권한만 부여됩니다. 또한 다음과 같은 방법으로 데이터를 보호하는 것이 좋습니다.

- 각 계정에 멀티 팩터 인증 설정(MFA)을 사용하세요.
- SSL/TLS를 사용하여 리소스와 통신하세요. AWS TLS 1.2를 지원합니다.
- 를 사용하여 API 및 사용자 활동 로깅을 설정합니다. AWS CloudTrail
- AWS 서비스 내의 모든 기본 보안 제어와 함께 AWS 암호화 솔루션을 사용하십시오.
- Amazon S3에 저장된 개인 데이터를 검색하고 보호하는 데 도움이 되는 Amazon Macie와 같은 고급 관리형 보안 서비스를 사용합니다.
- 명령줄 인터페이스 또는 API를 통해 AWS 에 액세스할 때 FIPS 140-2 검증된 암호화 모듈이 필요한 경우 FIPS 엔드포인트를 사용합니다. 사용 가능한 FIPS 엔드포인트에 대한 자세한 설명은 Federal Information Processing Standard(FIPS) 140-2를 참조하세요.

명칭 필드와 같은 자유 형식 필드에 고객 계정 번호와 같은 중요 식별 정보를 절대 입력하지 마세요. 여기에는 콘솔 AWS CLI, API 또는 AWS SDK를 사용하여 Transfer Family 또는 기타 AWS 서비스를 사용하는 경우가 포함됩니다. Transfer Family 서비스 구성에 입력하는 모든 구성 데이터 또는 다른 서비스의 구성이 진단 로그에 포함되도록 선택될 수 있습니다. 외부 서버에 URL을 제공할 때 해당 서버에 대한 요청을 검증하기 위해 보안 인증 정보를 URL에 포함시키지 마세요.

반면, Transfer Family 서버를 오가는 업로드 및 다운로드 작업의 데이터는 완전히 비공개로 취급되며 SFTP 또는 FTPS 연결과 같은 암호화된 채널 외부에 존재하지 않습니다. 이 데이터는 승인된 사람만 액세스할 수 있습니다.

주제

- Amazon S3의 데이터 암호화
- Transfer Family의 SSH 및 PGP 키 관리

Amazon S3의 데이터 암호화

AWS Transfer Family Amazon S3 버킷에 설정한 기본 암호화 옵션을 사용하여 데이터를 암호화합니다. 버킷에서 암호화를 활성화한 경우 모든 객체는 버킷에 저장될 때 암호화됩니다. Amazon S3 관리 키 (SSE-S3) 또는 AWS Key Management Service () 관리 키 (SSE-KMS) 를 사용한 서버 측 암호화를 사용하여 객체를 암호화합니다.AWS KMS자세한 설명은 Amazon Simple Storage Service 사용자 가이드의 서버 측 암호화를 사용하여 데이터 보호를 참조하세요.

다음 단계는 에서 데이터를 암호화하는 방법을 보여줍니다. AWS Transfer Family

에서 암호화를 허용하려면 AWS Transfer Family

1. Amazon S3 버킷에 대한 기본 암호화를 활성화합니다. 자세한 설명은 Amazon Simple Storage Service 사용자 가이드의 Amazon S3 버킷에 대한 Amazon S3 기본 암호화를 참조하세요.
2. 사용자에게 연결된 AWS Identity and Access Management (IAM) 역할 정책을 업데이트하여 필수 AWS Key Management Service (AWS KMS) 권한을 부여하십시오.
3. 사용자에 대한 세션 정책을 사용하는 경우 세션 정책은 필요한 AWS KMS 권한을 부여해야 합니다.

다음 예는 AWS KMS 암호화가 활성화된 Amazon S3 AWS Transfer Family 버킷과 함께 사용할 때 필요한 최소 권한을 부여하는 IAM 정책을 보여줍니다. 사용하고 있는 경우, 사용자 IAM 역할 정책과 세션 정책 모두에 이 예 정책을 포함시킵니다.

```
{
 "Sid": "Stmt1544140969635",
 "Action": [
  "kms:Decrypt",
  "kms:Encrypt",
  "kms:GenerateDataKey"
 ],
 "Effect": "Allow",
 "Resource": "arn:aws:kms:region:account-id:key/kms-key-id"
}
```

> **Note**
>
> 이 정책에서 지정하는 KMS 키 ID는 1단계에서 기본 암호화에 대해 지정한 것과 같아야 합니다.

루트 또는 사용자에게 사용되는 IAM 역할은 AWS KMS 키 정책에서 허용되어야 합니다. AWS KMS 키 정책에 대한 자세한 내용은 AWS Key Management Service 개발자 안내서의 AWS KMS에서의 키 정책 사용을 참조하십시오.

Transfer Family의 SSH 및 PGP 키 관리

이 섹션에서는 SSH 키를 생성하는 방법과 키를 회전하는 방법 등 SSH 키에 대한 정보를 확인할 수 있습니다. Transfer Family를 사용하여 키를 관리하는 방법에 대한 자세한 내용은 A AWS Transfer Family 및 AWS Lambda를 사용하여 사용자 셀프 서비스 키 관리 AWS Lambda 활성화하기 블로그 게시물을 참조하십시오.

> **Note**
>
> AWS Transfer Family RSA, ECDSA 및 ED25519 키를 허용합니다.

이 섹션에서는 프리티 굿 프라이버시 (PGP) 키를 생성하고 관리하는 방법도 다룹니다.

주제

- 사용자 및 서버 키에 지원되는 알고리즘
- 서비스 관리 사용자를 위한 SSH 키 생성
- SSH 키 교체
- PGP 키 생성 및 관리
- 지원되는 PGP 클라이언트

사용자 및 서버 키에 지원되는 알고리즘

AWS Transfer Family내에서 사용자 및 서버 키 쌍에 대해 다음과 같은 키 알고리즘이 지원됩니다.

> **Note**
>
> 워크플로에서 PGP 복호화와 함께 사용할 알고리즘은 PGP 키 쌍에 지원되는 알고리즘을 참조하세요.

- ED25519의 경우: `ssh-ed25519`

- RSA의 경우:
  - `rsa-sha2-256`
  - `rsa-sha2-512`
- ECDSA의 경우:
  - `ecdsa-sha2-nistp256`
  - `ecdsa-sha2-nistp384`
  - `ecdsa-sha2-nistp521`

> **Note**
>
> 이전 보안 정책에 대해 `ssh-rsa` SHA1을 지원합니다. 자세한 내용은 암호화 알고리즘를 참조하세요.

서비스 관리 사용자를 위한 SSH 키 생성

서버를 설정해 서비스 관리형 인증 메서드를 사용하는 사용자를 인증할 수 있습니다. 이때 사용자 이름과 SSH 키는 서비스 내에 보관됩니다. 사용자의 퍼블릭 SSH 키는 서버에 사용자의 속성으로 업로드됩니다. 이 키는 서버에서 표준 키 기반 인증 프로세스의 일부로 사용됩니다. 각 사용자는 파일에 여러 퍼블릭 SSH 키를 개별 서버와 함께 보유할 수 있습니다. 사용자당 저장할 수 있는 키 수에 대한 제한은 Amazon Web Services 일반 참조의 AWS Transfer Family 엔드포인트 및 할당량을 참조하세요.

서비스 관리형 인증 방법 대신 사용자 지정 ID 공급자 또는 를 사용하여 사용자를 인증할 수 있습니다. AWS Directory Service for Microsoft Active Directory자세한 내용은 사용자 지정 자격 증명 공급자와 작업 또는 AWS 디렉터리 서비스 ID 제공자 사용를 참조하세요.

서버는 한 가지 방법 (서비스 관리형, 디렉터리 서비스 또는 맞춤 ID 제공자)을 사용해서만 사용자를 인증할 수 있으며, 서버를 만든 후에는 이 방법을 변경할 수 없습니다.

주제

- macOS, Linux 또는 Unix에서 SSH 키 생성
- Microsoft Windows에서 SSH 키 생성
- SSH2 퍼블릭 키를 PEM 형식으로 변환

macOS, Linux 또는 Unix에서 SSH 키 생성

macOS, Linux 또는 Unix 작동 시스템에서는 `ssh-keygen` 명령을 사용하여 SSH 퍼블릭 키와 키 쌍이라고도 하는 SSH 프라이빗 키를 생성합니다.

macOS, Linux 또는 Unix 작동 시스템에서 SSH 키를 만들려면

1. macOS, Linux 또는 Unix 작동 시스템에서는 명령 터미널을 엽니다.
2. AWS Transfer Family RSA-, ECDSA- 및 ED25519 형식의 키를 허용합니다. 생성하려는 키 쌍의 타입에 따라 적절한 명령을 선택합니다.

> **Note**
>
> 다음 예에서는 패스프레이즈를 지정하지 않았는데, 이 경우 도구에서 패스프레이즈를 입력한 다음 반복하여 확인하도록 요청합니다. 패스프레이즈를 만들면 프라이빗 키를 더 잘 보호할 수 있고 전반적인 시스템 보안도 향상될 수 있습니다. 패스프레이즈는 복구할 수 없습니다: 암호를 잊은 경우 새 키를 생성해야 합니다.
> 그러나 Transfer Family 서버는 시작 시 패스프레이즈를 요청할 수 없으므로 서버 호스트 키를 생성하는 경우 명령에서 `-N ""` 옵션을 지정하거나 메시지가 표시되면 **Enter**을 두 번 눌러 빈 암호를 지정해야 합니다.

- RSA 4096비트 키 쌍 생성:

```
ssh-keygen -t rsa -b 4096 -f key_name
```

- ECDSA 521비트 키 쌍을 생성하려면 (ECDSA의 비트 크기는 256, 384, 521입니다):

```
ssh-keygen -t ecdsa -b 521 -f key_name
```

- ED25519 키 쌍을 생성하려면:

```
ssh-keygen -t ed25519 -f key_name
```

> **Note**
>
> *key_name*은(는) SSH 키 쌍 파일 명칭입니다.

다음은 ssh-keygen 출력의 예입니다.

```
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**
>
> 앞의 예처럼 ssh-keygen 명령을 실행하면, 현재 디렉터리에 퍼블릭 및 프라이빗 키가 파일로 생성됩니다.

이제 SSH 키 쌍을 사용할 준비가 되었습니다. 3단계와 4단계에 따라 서비스 관리 사용자를 위한 SSH 공개 키를 저장하세요. 이러한 사용자는 Transfer Family 서버 엔드포인트에서 파일을 전송할 때 이 키를 사용합니다.

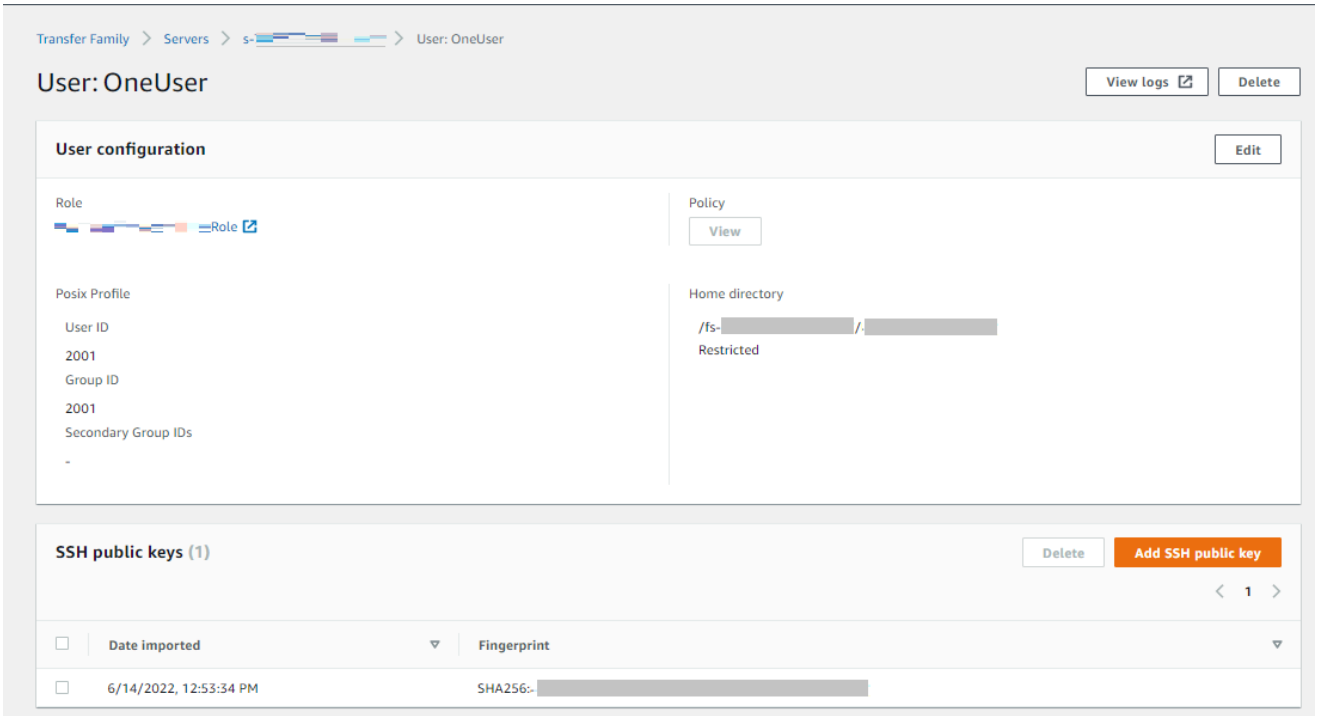
3. *key_name*.pub 파일로 이동하여 엽니다.
4. 텍스트를 복사하여 서비스 관리 사용자의 SSH 퍼블릭 키에 붙여넣습니다.

   a. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 연 다음 탐색 창에서 서버를 선택합니다.

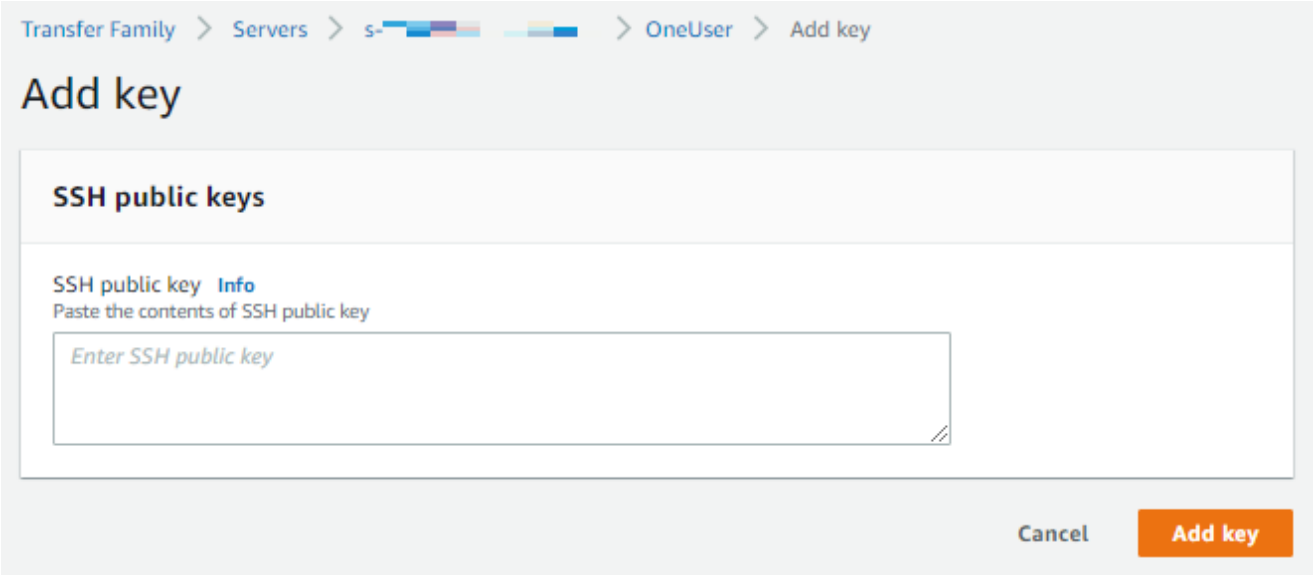
b. 서버 페이지에서 업데이트하려는 사용자가 포함된 서버의 서버 ID를 선택합니다.

c. 퍼블릭 키를 추가할 사용자를 선택합니다.

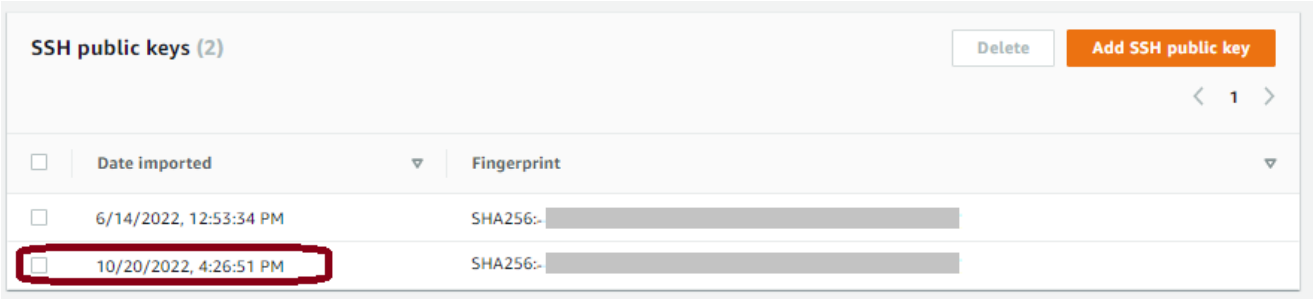
d. SSH 퍼블릭 키 창에서 SSH 퍼블릭 키 추가를 선택합니다.



e. 생성한 퍼블릭 키 텍스트를 SSH 퍼블릭 키 텍스트 상자에 붙여넣은 다음 키 추가를 선택합니다.



새 키는 SSH 퍼블릭 키 창에 나열됩니다.



Microsoft Windows에서 SSH 키 생성

Windows는 약간 다른 SSH 키 쌍 형식을 사용합니다. 퍼블릭 키는 PUB 형식이어야 하고, 프라이빗 키는 PPK 형식이어야 합니다. Windows에서는 PuTTYgen을 이용해 적절한 형식의 SSH 키 쌍을 만들 수 있습니다. 또 PuTTYgen을 사용하면 `ssh-keygen`으로 생성한 프라이빗 키를 `.ppk` 파일로 변환할 수도 있습니다.

> **Note**
>
> WinSCP를 `.ppk` 형식이 아닌 프라이빗 키 파일과 함께 제공하면, 클라이언트는 키를 `.ppk` 형식으로 변환할 기회를 제공합니다.

Windows에서 PuTTYgen을 이용해 SSH 키를 생성하는 자습서를 확인하고 싶다면, SSH.com 웹사이트를 참조하세요.

SSH2 퍼블릭 키를 PEM 형식으로 변환

AWS Transfer Family PEM 형식의 공개 키만 허용합니다. SSH2 퍼블릭 키가 있는 경우 이를 변환해야 합니다. SSH2 퍼블릭 키의 형식은 다음과 같습니다:

```
---- BEGIN SSH2 PUBLIC KEY ----
Comment: "rsa-key-20160402"
AAAAB3NzaC1yc2EAAAABJQAAAgEAiL0jjDdFqK/kYThqKt7THrjABTPWvXmB3URI
:
:
---- END SSH2 PUBLIC KEY ----
```

PEM 퍼블릭 키의 형식은 다음과 같습니다:

```
ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAA...
```

다음 명령을 실행하여 SSH2 형식의 퍼블릭 키를 PEM 형식의 퍼블릭 키로 변환합니다. *ssh2-key*를 SSH2 키 명칭으로 바꾸고 *PEM-key*를 PEM 키 명칭으로 바꾸세요.

```
ssh-keygen -i -f ssh2-key.pub > PEM-key.pub
```

SSH 키 교체

보안상 SSH 키 교체는 보안 모범 사례로 권장됩니다. 일반적으로 이러한 교체는 보안 정책의 일부로 지정되며, 자동화된 방식으로 구현됩니다. 보안 수준에 따라, 대단히 민감한 통신의 경우 SSH 키 쌍을 한 번만 사용할 수도 있습니다. 이렇게 하면 저장된 키 때문에 발생하는 모든 위험을 제거할 수 있습니다. 하지만 SSH 보안 인증을 일정 기간 보관하고, SFTP 사용자에게 과도한 부담을 주지 않는 주기를 설정하는 방법이 더 보편적입니다. 가장 일반적인 주기는 3개월입니다.

SSH 키 교체는 2가지 방식으로 실행합니다.

- 콘솔에서 새 SSH 퍼블릭 키를 업로드하고 기존 SSH 퍼블릭 키를 삭제할 수 있습니다.
- API를 사용하면 API를 사용하여 사용자의 SSH (Secure Shell) 공개 키를 삭제하고 DeleteSshPublicKeyAPI를 사용하여 사용자 계정에 새 SSH (Secure Shell) 공개 키를 추가하여 기존 사용자를 업데이트할 수 있습니다. ImportSshPublicKey

Console

콘솔에서 키 로테이션을 수행하려면

1. https://console.aws.amazon.com/transfer/ 에서 AWS Transfer Family 콘솔을 엽니다.
2. 서버 페이지로 이동합니다.
3. 서버 ID 열에서 식별자를 선택하여 서버 세부 정보 페이지를 표시합니다.
4. 사용자에서 SSH 퍼블릭 키를 교체하려는 사용자의 확인란을 선택한 다음 작업을 선택하고 키 추가를 선택하여 키 추가 페이지를 확인합니다.

   또는

   사용자 세부 정보 페이지를 보려면 사용자 이름을 선택한 다음 SSH 퍼블릭 키 추가를 선택하여 키 추가 페이지를 확인합니다.
5. 새 SSH 퍼블릭 키를 입력하고 키 추가를 선택합니다.

> **Important**
>
> SSH 퍼블릭 키의 형식은 생성한 키 타입에 따라 다릅니다.
>
> - RSA 키의 경우, 그 형식은 `ssh-rsa` *`string`*입니다.
> - ED25519 키의 경우, 그 형식은 `ssh-ed25519` *`string`*입니다.
> - ECDSA 키의 경우, 키는 생성한 키 크기에 따라 `ecdsa-sha2-nistp256`, `ecdsa-sha2-nistp384`, 또는 `ecdsa-sha2-nistp521`(으)로 시작합니다. 그런 다음 다른 키 타입과 마찬가지로 시작 문자열 뒤에 *`string`*이(가) 옵니다.

사용자 세부 정보 페이지로 돌아오면 방금 입력한 새 SSH 퍼블릭 키가 SSH 퍼블릭 키 섹션에 표시됩니다.

6. 삭제할 보안 인증 옆의 확인란을 선택하고 삭제를 선택합니다.
7. 단어 `delete`를 입력하여 삭제 작업을 확인한 다음 삭제를 선택합니다.

API

API를 사용하여 키 로테이션을 수행하려면

1. macOS, Linux 또는 Unix 작동 시스템에서는 명령 터미널을 엽니다.
2. 다음 명령을 입력하여 삭제하려는 SSH 키를 검색합니다. 이 명령을 사용하려면, *`serverID`*를 Transfer Family 서버의 서버 ID로 바꾸고 *`username`*를 사용자 이름으로 바꾸세요.

```
aws transfer describe-user --server-id='serverID' --user-name='username'
```

이 명령은 사용자에 대한 세부 정보를 반환합니다. `"SshPublicKeyId":` 필드 내용을 복사합니다. 이 값은 이 절차의 뒷부분에서 입력해야 합니다.

```
"SshPublicKeys": [ { "SshPublicKeyBody": "public-key", "SshPublicKeyId":
 "keyID",
   "DateImported": 1621969331.072 } ],
```

3. 다음으로, 사용자의 새 SSH 키를 가져옵니다. 프롬프트에 다음 명령을 입력합니다. 이 명령을 사용하려면 *`serverID`*를 Transfer Family 서버의 서버 ID로 바꾸고 *`username`*를 사용자 이름으로 바꾸고 *`public-key`*를 새 퍼블릭 키의 지문으로 바꾸세요.

```
aws transfer import-ssh-public-key --server-id='serverID' --user-name='username'
   --ssh-public-key-body='public-key'
```

이 명령이 성공하면 출력이 반환되지 않습니다.

4. 마지막으로 다음 명령을 실행하여 이전 키를 삭제합니다. 이 명령을 사용하려면 *serverID*를 Transfer Family 서버의 서버 ID로 바꾸고 *username*를 사용자 이름으로 바꾸고 *keyID-from-step-2*를 이 절차의 2단계에서 복사한 키 ID 값으로 바꾸세요.

```
aws transfer delete-ssh-public-key --server-id='serverID' --user-name='username'
   --ssh-public-key-id='keyID-from-step-2'
```

5. (옵션) 이전 키가 더 이상 존재하지 않는지 확인하려면 2단계를 반복합니다.

PGP 키 생성 및 관리

Transfer Family가 워크플로를 통해 처리하는 파일에 프리티 굿 프라이버시(PGP) 복호화을 사용할 수 있습니다. 워크플로 단계에서 암호 해독을 사용하려면 PGP 키를 제공하십시오.

AWS 스토리지 블로그에는 Transfer Family Managed 워크플로를 사용하여 코드를 작성하지 않고 파일을 간단히 해독하는 방법, PGP를 사용한 파일 암호화 및 암호 해독 방법을 설명하는 게시물이 있습니다. AWS Transfer Family

PGP 키 생성

PGP 키를 생성하는 데 사용하는 연산자는 운영 체제와 사용 중인 키 생성 소프트웨어 버전에 따라 다릅니다.

Linux 또는 Unix를 사용하는 경우 패키지 설치 프로그램을 사용하여 gpg를 설치하세요. Linux 배포판에 따라 다음 명령 중 하나를 사용할 수 있습니다.

```
sudo yum install gnupg
```

```
sudo apt-get install gnupg
```

Windows 또는 macOS의 경우, https://gnupg.org/download/에서 필요한 것을 다운로드할 수 있습니다.

PGP 키 생성기 소프트웨어를 설치한 후 `gpg --full-gen-key` 또는 `gpg --gen-key` 명령을 실행하여 키 쌍을 생성합니다.

> **Note**
>
> GnuPG 버전 2.3.0 이상을 사용하는 경우 `gpg --full-gen-key`를 실행해야 합니다. 생성할 키 타입을 묻는 메시지가 표시되면 RSA 또는 ECC를 선택합니다. 하지만 ECC를 선택할 경우 타원 곡선에서 NIST 또는 BrainPool을 선택해야 합니다. Curve 25519를 선택하지 마세요.

PGP 키 쌍에 지원되는 알고리즘

PGP 키 쌍에 대해 다음과 같은 알고리즘을 지원합니다.

- RSA
- Elgamal
- ECC:
  - NIST
  - BrainPool

> **Note**
>
> CCurve25519 키는 지원되지 않습니다.

유용한 **gpg** 하위 명령

다음은 gpg에 유용한 몇 가지 하위 명령입니다:

- `gpg --help` – 이 명령에는 사용 가능한 옵션이 나열되며 몇 가지 예가 포함될 수 있습니다.
- `gpg --list-keys`— 이 명령은 생성한 모든 키 페어의 세부 정보를 나열합니다.
- `gpg --fingerprint`— 이 명령은 각 키의 핑거프린트를 포함하여 모든 키 페어에 대한 세부 정보를 나열합니다.
- `gpg --export -a` *`user-name`* – 이 명령은 키가 생성될 때 사용된 *`user-name`* 키의 퍼블릭 키 부분을 내보냅니다.

PGP 키 관리

PGP 키를 관리하려면 를 사용하십시오. AWS Secrets Manager

> **Note**
>
> 비밀 명칭에는 Transfer Family 서버 ID가 포함됩니다. 즉, PGP 키 정보를 AWS Secrets Manager에 저장하기에 앞서 먼저 서버를 식별하거나 생성했어야 합니다.

모든 사용자에 대해 하나의 키와 패스프레이즈를 사용하려는 경우 PGP 키 블록 정보를 비밀 명칭 `aws/transfer/`*`server-id`*`/@pgp-default` 아래에 저장할 수 있습니다. 여기서 *`server-id`*은(는) Transfer Family 서버의 ID입니다. Transfer Family는 워크플로를 실행하는 사용자와 *`user-name`* 일치하는 키가 없는 경우 이 기본 키를 사용합니다.

특정 사용자를 위한 키를 생성할 수 있습니다. 이 경우 암호 이름의 형식은 다음과 같습니다`aws/transfer/`*`server-id`*`/`*`user-name`*. 여기서 Transfer Family 서버의 워크플로를 실행하는 사용자와 *`user-name`* 일치합니다.

> **Note**
>
> 사용자당 Transfer Family 서버당 최대 3개의 PGP 프라이빗 키를 저장할 수 있습니다.

복호화에 사용할 PGP 키를 구성하려면

1. 사용 중인 GPG 버전에 따라 다음 명령 중 하나를 실행하여 Curve 25519 암호화 알고리즘을 사용하지 않는 PGP 키 쌍을 생성하십시오.

   - **GnuPG** 버전 2.3.0 이상을 사용하는 경우, 다음 명령을 실행합니다.

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

     RSA를 선택할 수도 있고, **ECC**를 선택할 경우 **NIST** 또는 **BrainPool** 하나를 타원 곡선으로 선택할 수 있습니다. `gpg --gen-key` 대신 실행하는 경우 ECC Curve 25519 암호화 알고리즘을 사용하는 키 쌍을 생성합니다. 이 알고리즘은 현재 PGP 키를 지원하지 않습니다.

   - RSA가 기본 암호화 타입이므로 2.3.0 이전의 **GnuPG** 버전의 경우, 다음 명령을 사용할 수 있습니다.

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

> **Important**
>
> 키 생성 프로세스 중에 패스프레이즈와 이메일 주소를 제공해야 합니다. 이러한 값을 기록해 두세요. 이 절차의 AWS Secrets Manager 뒷부분에서 키의 세부 정보를 입력할 때 패스프레이즈를 제공해야 합니다. 그리고 다음 단계에서 프라이빗 키를 내보내려면 동일한 이메일 주소를 제공해야 합니다.

2. 프라이빗 키를 내보내려면 다음 명령을 실행합니다. 이 명령을 사용하려면 *private.pgp*를 프라이빗 키 블록을 저장할 파일 명칭으로, *marymajor@example.com*를 키 쌍을 생성할 때 사용한 이메일 주소로 바꾸세요.

```
gpg --output private.pgp --armor --export-secret-key marymajor@example.com
```

3. PGP 키를 저장하는 AWS Secrets Manager 데 사용합니다.

   a. https://console.aws.amazon.com/secretsmanager/ 에서 AWS Management Console 로그인하고 AWS Secrets Manager 콘솔을 엽니다.

   b. 왼쪽 탐색 창에서 암호를 선택합니다.

   c. 암호 페이지에서 새 암호 저장을 선택합니다.

   d. 암호 타입 선택 페이지의 암호 타입에 대해 다른 타입의 암호를 선택합니다.

   e. 키/값 쌍 섹션에서 키/값 탭을 선택합니다.

   - 키 – **PGPPrivateKey**를 입력합니다.

     > **Note**
     >
     > **PGPPrivateKey** 문자열을 정확히 입력해야 합니다: 문자 앞이나 사이에 공백을 추가하지 마세요.

   - 값 — 프라이빗 키의 텍스트를 값 필드에 붙여넣습니다. 이 절차의 앞부분에서 키를 내보낼 때 지정한 파일 (예: `private.pgp`) 에서 프라이빗 키의 텍스트를 찾을 수 있습니다. 키는 `-----BEGIN PGP PRIVATE KEY BLOCK-----`(으)로 시작하고 `-----END PGP PRIVATE KEY BLOCK-----`(으)로 끝납니다.

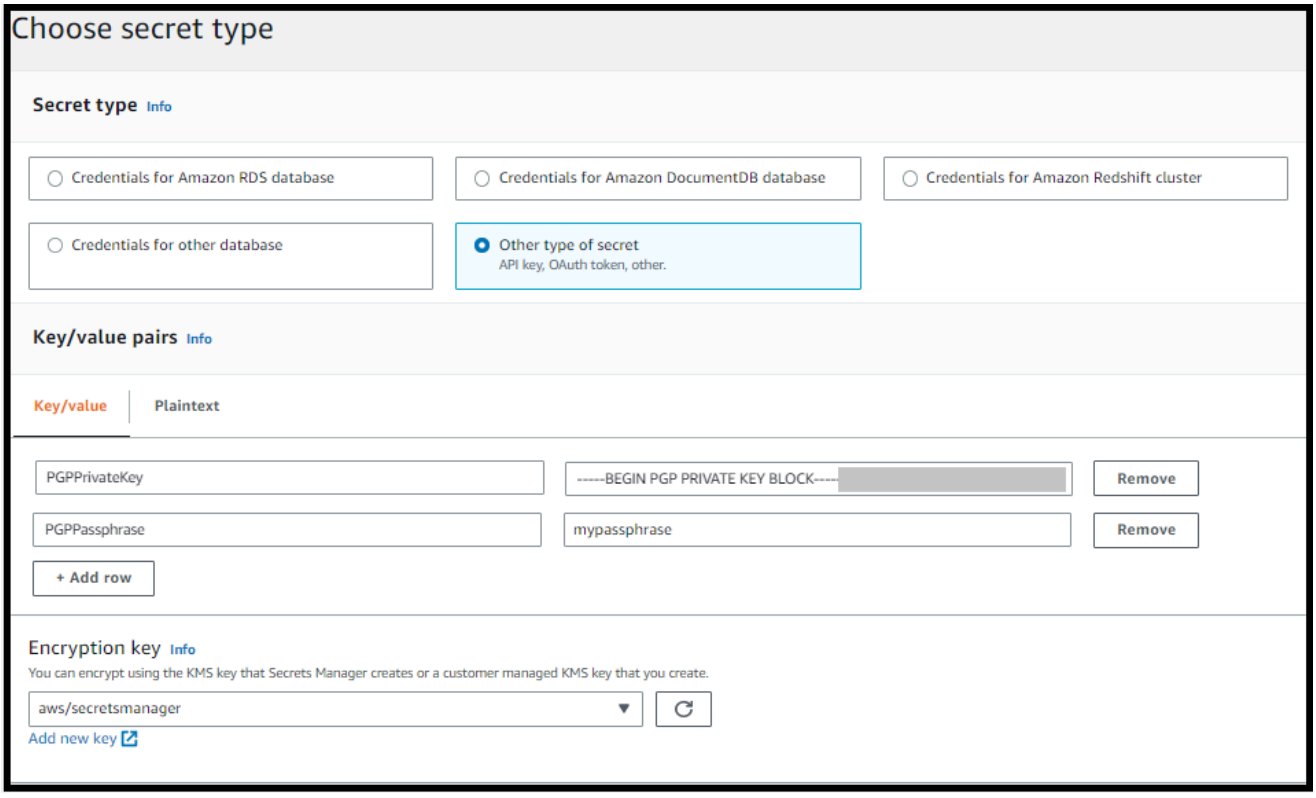
> **Note**
>
> 텍스트 블록에는 프라이빗 키만 포함되고 퍼블릭 키는 포함되지 않는지 확인하세요.

f. 행 추가를 선택하고 키/값 쌍 섹션에서 키/값 탭을 선택합니다.

- 키 – **PGPPassphrase**를 입력합니다.

> **Note**
>
> **PGPPassphrase** 문자열을 정확히 입력해야 합니다: 문자 앞이나 사이에 공백을 추가하지 마세요.

- 값 – PGP 키 쌍을 생성할 때 사용한 패스프레이즈를 입력합니다.



> **Note**
>
> 최대 3세트의 키와 패스프레이즈를 추가할 수 있습니다. 두 번째 세트를 추가하려면 새 행 두 개를 추가하고 키에 **PGPPrivateKey2**와(과) **PGPPassphrase2**를 입력한 다음 다른 프라이빗 키와 패스프레이즈를 붙여넣습니다. 세 번째 세트를 추가하려면 키 값이 **PGPPrivateKey3** 및 **PGPPassphrase3**여야 합니다.

g. 다음을 선택합니다.

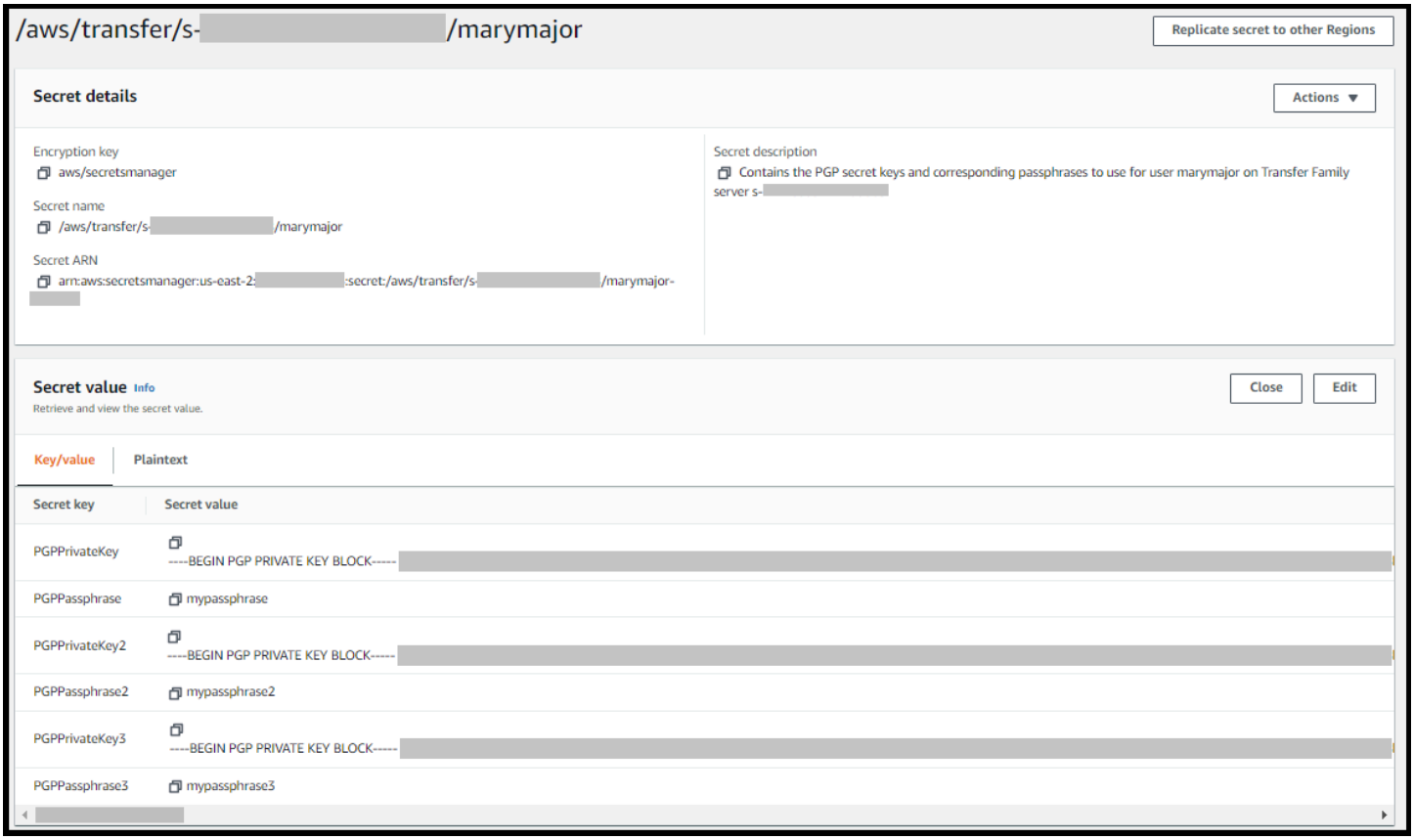
h. 암호 구성 페이지에 암호를 위한 명칭과 설명을 입력합니다.

- 기본 키, 즉 모든 Transfer Family 사용자가 사용할 수 있는 키를 만들려면 다음을 입력하세요**aws/transfer/*server-id*/@pgp-default**. *server-id*를 복호화 단계가 있는 워크플로가 포함된 서버의 ID로 바꾸세요.
- 특정 Transfer Family 사용자가 사용할 키를 만들려면 **aws/transfer/*server-id*/*user-name***를 입력하세요. *server-id*를 복호화 단계가 있는 워크플로가 포함된 서버의 ID로 바꾸고 *user-name*를 워크플로를 실행하는 사용자 이름으로 바꾸세요. *user-name*은(는) Transfer Family 서버가 사용하는 ID 제공자에 저장됩니다.

i. 다음을 선택하고 순환 구성 페이지의 기본값을 그대로 사용합니다. 그 다음 다음을 선택합니다.

j. 검토 페이지에서 저장을 선택하여 암호를 만들고 저장합니다.

다음 스크린샷은 특정 Transfer Family 서버의 사용자 **marymajor**의 세부 정보를 보여줍니다. 이 예에서는 세 개의 키와 해당 패스프레이즈를 보여줍니다.

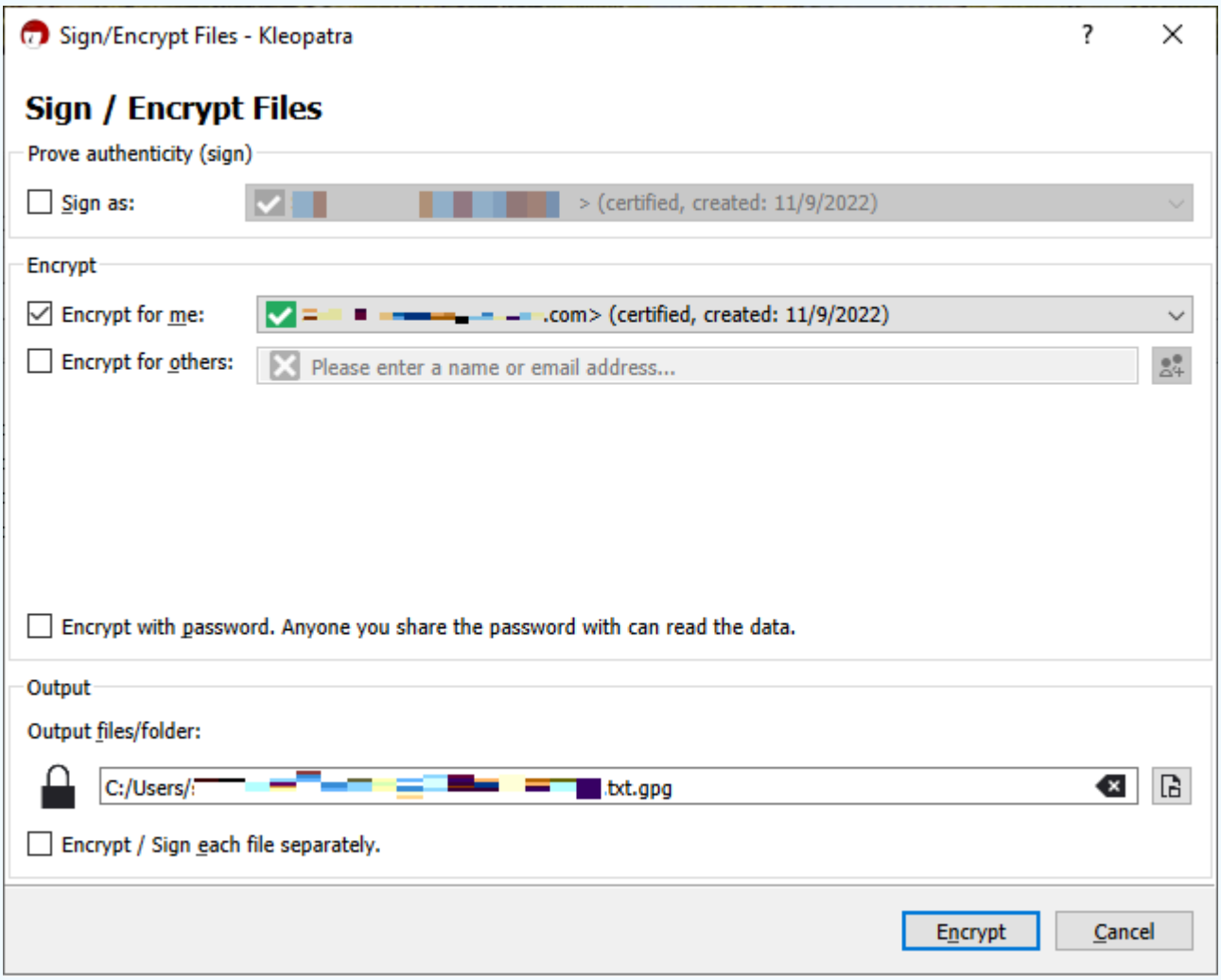


지원되는 PGP 클라이언트

다음 클라이언트는 Transfer Family에서 테스트되었으며 PGP 키를 생성하고 워크플로를 통해 해독하려는 파일을 암호화하는 데 사용할 수 있습니다.

- Gpg4win + Kleopatra.

> **Note**
>
> 파일 서명/암호화를 선택할 때는 서명 주체 선택을 취소해야 합니다: 현재 암호화된 파일에 대한 서명은 지원하지 않습니다.



암호화된 파일에 서명하고 암호 해독 워크플로를 사용하여 Transfer Family 서버에 업로드하려고 하면 다음과 같은 오류 메시지가 나타납니다.

```
Encrypted file with signed message unsupported
```

- 주요 GnuPG 버전: 2.4, 2.3, 2.2, 2.0, 1.4.

다른 PGP 클라이언트도 작동할 수 있지만 여기에 언급된 클라이언트만 Transfer Family로 테스트되었습니다.

에 대한 ID 및 액세스 관리 AWS Transfer Family

AWS Identity and Access Management IAM (IAM) 은 관리자가 AWS 서비스 리소스에 대한 액세스를 안전하게 제어할 수 있도록 AWS 도와줍니다. IAM 관리자는 리소스를 사용할 수 있는 인증 (로그인) 및

권한 부여 (권한 보유) 를 받을 수 있는 사용자를 제어합니다. AWS Transfer Family IAM은 추가 AWS 서비스 비용 없이 사용할 수 있습니다.

주제

고객

사용하는 방식 AWS Identity and Access Management (IAM) 은 수행하는 작업에 따라 다릅니다. AWS Transfer Family

서비스 사용자 - AWS Transfer Family 서비스를 사용하여 작업을 수행하는 경우 관리자가 필요한 자격 증명과 권한을 제공합니다. 더 많은 AWS Transfer Family 기능을 사용하여 작업을 수행함에 따라 추가 권한이 필요할 수 있습니다. 액세스 권한 관리 방식을 이해하면 적절한 권한을 관리자에게 요청할 수 있습니다. AWS Transfer Family의 기능에 액세스할 수 없는 경우 AWS Transfer Family ID 및 액세스 문제 해결을 참조하세요.

서비스 관리자 — 회사에서 AWS Transfer Family 리소스를 담당하는 경우 전체 액세스 권한이 있을 수 AWS Transfer Family있습니다. 서비스 사용자가 액세스해야 하는 AWS Transfer Family 기능과 리소스를 결정하는 것은 여러분의 몫입니다. 그런 다음, IAM 관리자에게 요청을 제출하여 서비스 사용자의 권한을 변경해야 합니다. 이 페이지의 정보를 검토하여 IAM의 기본 개념을 이해하십시오. 회사에서 IAM을 어떻게 사용할 수 있는지 자세히 AWS Transfer Family알아보려면 을 참조하십시오IAM의 AWS Transfer Family 작동 방식.

IAM 관리자 - IAM 관리자라면 AWS Transfer Family에 대한 액세스 권한 관리 정책 작성 방법을 자세히 알고 싶을 것입니다. IAM에서 사용할 수 있는 AWS Transfer Family ID 기반 정책의 예를 보려면 을 참조하십시오. AWS Transfer Family ID 기반 정책 예제

ID를 통한 인증

인증은 자격 증명 자격 증명을 AWS 사용하여 로그인하는 방법입니다. IAM 사용자로 인증 (로그인 AWS) 하거나 IAM 역할을 맡아 인증 (로그인) 해야 합니다. AWS 계정 루트 사용자

ID 소스를 통해 제공된 자격 증명을 사용하여 페더레이션 ID로 로그인할 수 있습니다. AWS AWS IAM Identity Center (IAM ID 센터) 사용자, 회사의 싱글 사인온 인증, Google 또는 Facebook 자격 증명이 페더레이션 ID의 예입니다. 연동 자격 증명으로 로그인할 때 관리자가 이전에 IAM 역할을 사용하여 ID 페더레이션을 설정했습니다. 페더레이션을 사용하여 액세스하는 경우 AWS 간접적으로 역할을 맡게 됩니다.

사용자 유형에 따라 AWS Management Console 또는 AWS 액세스 포털에 로그인할 수 있습니다. 로그인에 대한 자세한 내용은 AWS 로그인 사용 설명서의 내 로그인 방법을 참조하십시오. AWS AWS 계정

AWS 프로그래밍 방식으로 액세스하는 경우 자격 증명을 사용하여 요청에 암호화 방식으로 서명할 수 있는 소프트웨어 개발 키트 (SDK) 와 명령줄 인터페이스 (CLI) 를 AWS 제공합니다. AWS 도구를 사용하지 않는 경우 요청에 직접 서명해야 합니다. 권장 방법을 사용하여 직접 요청에 서명하는 방법에 대한 자세한 내용은 IAM 사용 설명서의 AWS API 요청 서명을 참조하십시오.

사용하는 인증 방법에 상관없이 추가 보안 정보를 제공해야 할 수도 있습니다. 예를 들어, AWS 계정의 보안을 강화하기 위해 다단계 인증 (MFA) 을 사용할 것을 권장합니다. 자세한 내용은 AWS IAM Identity Center 사용 설명서의 다중 인증 및 IAM 사용 설명서의 AWS에서 다중 인증(MFA) 사용을 참조하세요.

AWS 계정 루트 사용자

계정을 만들 때는 먼저 AWS 계정계정의 모든 AWS 서비스 리소스와 모든 리소스에 완전히 액세스할 수 있는 하나의 로그인 ID로 시작합니다. 이 ID를 AWS 계정 루트 사용자라고 하며, 계정을 만들 때 사용한 이메일 주소와 비밀번호로 로그인하여 액세스할 수 있습니다. 일상적인 태스크에 루트 사용자를 사용하지 않을 것을 강력히 권장합니다. 루트 사용자 보안 인증 정보를 보호하고 루트 사용자만 수행할 수 있는 태스크를 수행하는 데 사용하세요. 루트 사용자로 로그인해야 하는 태스크의 전체 목록은 IAM 사용자 안내서의 루트 사용자 보안 인증이 필요한 태스크를 참조하세요.

연동 자격 증명

가장 좋은 방법은 관리자 액세스가 필요한 사용자를 비롯한 수동 AWS 서비스 사용자가 ID 공급자와의 페더레이션을 사용하여 임시 자격 증명을 사용하여 액세스하도록 하는 것입니다.

페더레이션 ID는 기업 사용자 디렉토리, 웹 ID 공급자, Identity Center 디렉터리의 사용자 또는 ID 소스를 통해 제공된 자격 증명을 사용하여 액세스하는 AWS 서비스 모든 사용자를 말합니다. AWS

Directory Service페더레이션 ID에 AWS 계정액세스하면 이들이 역할을 맡고 역할은 임시 자격 증명을 제공합니다.

중앙 집중식 액세스 관리를 위해 AWS IAM Identity Center(을)를 사용하는 것이 좋습니다. IAM Identity Center에서 사용자 및 그룹을 생성하거나 자체 ID 소스의 사용자 및 그룹 집합에 연결하고 동기화하여 모든 사용자 및 애플리케이션에서 사용할 수 있습니다. AWS 계정 IAM Identity Center에 대한 자세한 내용은AWS IAM Identity Center 사용 설명서에서 IAM Identity Center란 무엇입니까?를 참조하세요.

IAM 사용자 및 그룹

IAM 사용자는 단일 사용자 또는 애플리케이션에 대한 특정 권한을 AWS 계정 가진 사용자 내 자격 증명입니다. 가능하면 암호 및 액세스 키와 같은 장기 자격 증명이 있는 IAM 사용자를 생성하는 대신 임시 자격 증명을 사용하는 것이 좋습니다. 하지만 IAM 사용자의 장기 자격 증명이 필요한 특정 사용 사례가 있는 경우 액세스 키를 교체하는 것이 좋습니다. 자세한 내용은 IAM 사용 설명서의 장기 보안 인증이 필요한 사용 사례의 경우 정기적으로 액세스 키 교체를 참조하세요.

IAM 그룹은 IAM 사용자 컬렉션을 지정하는 자격 증명입니다. 사용자는 그룹으로 로그인할 수 없습니다. 그룹을 사용하여 여러 사용자의 권한을 한 번에 지정할 수 있습니다. 그룹을 사용하면 대규모 사용자 집합의 권한을 더 쉽게 관리할 수 있습니다. 예를 들어, IAMAdmins라는 그룹이 있고 이 그룹에 IAM 리소스를 관리할 권한을 부여할 수 있습니다.

사용자는 역할과 다릅니다. 사용자는 한 사람 또는 애플리케이션과 고유하게 연결되지만, 역할은 해당 역할이 필요한 사람이라면 누구나 수임할 수 있습니다. 사용자는 영구적인 장기 보안 인증을 가지고 있지만, 역할은 임시 보안 인증만 제공합니다. 자세한 정보는 IAM 사용 설명서의 IAM 사용자를 만들어야 하는 경우(역할이 아님)를 참조하세요.

IAM 역할

IAM 역할은 특정 권한을 가진 사용자 AWS 계정 내의 자격 증명입니다. IAM 사용자와 유사하지만, 특정 개인과 연결되지 않습니다. 역할을 AWS Management Console 전환하여 에서 일시적으로 IAM 역할을 맡을 수 있습니다. AWS CLI 또는 AWS API 작업을 호출하거나 사용자 지정 URL을 사용하여 역할을 수임할 수 있습니다. 역할 사용 방법에 대한 자세한 정보는 IAM 사용 설명서의 IAM 역할 사용을 참조하세요.

임시 보안 인증이 있는 IAM 역할은 다음과 같은 상황에서 유용합니다.

- 페더레이션 사용자 액세스 - 연동 자격 증명에 권한을 부여하려면 역할을 생성하고 해당 역할의 권한을 정의합니다. 연동 자격 증명이 인증되면 역할이 연결되고 역할에 정의된 권한이 부여됩니다. 페더레이션 역할에 대한 자세한 내용은 IAM 사용 설명서의 타사 자격 증명 공급자의 역할 만들기를 참

조하세요. IAM Identity Center를 사용하는 경우 권한 세트를 구성합니다. 인증 후 아이덴티티가 액세스할 수 있는 항목을 제어하기 위해 IAM Identity Center는 권한 세트를 IAM의 역할과 연관 짓습니다. 권한 세트에 대한 자세한 내용은 AWS IAM Identity Center 사용 설명서의 권한 세트를 참조하세요.

- 임시 IAM 사용자 권한 - IAM 사용자 또는 역할은 IAM 역할을 수임하여 특정 태스크에 대한 다양한 권한을 임시로 받을 수 있습니다.
- 크로스 계정 액세스 - IAM 역할을 사용하여 다른 계정의 사용자(신뢰할 수 있는 보안 주체)가 내 계정의 리소스에 액세스하도록 허용할 수 있습니다. 역할은 계정 간 액세스를 부여하는 기본적인 방법입니다. 그러나 일부 AWS 서비스경우에는 역할을 프록시로 사용하는 대신 정책을 리소스에 직접 연결할 수 있습니다. 크로스 계정 액세스를 위한 역할과 리소스 기반 정책의 차이점을 알아보려면 IAM 사용 설명서의 IAM 역할과 리소스 기반 정책의 차이를 참조하세요.
- 서비스 간 액세스 — 일부는 다른 AWS 서비스서비스의 기능을 AWS 서비스 사용합니다. 예컨대, 어떤 서비스에서 호출을 수행하면 일반적으로 해당 서비스는 Amazon EC2에서 애플리케이션을 실행하거나 Amazon S3에 객체를 저장합니다. 서비스는 호출하는 보안 주체의 권한을 사용하거나, 서비스 역할을 사용하거나, 또는 서비스 연결 역할을 사용하여 이 작업을 수행할 수 있습니다.
  - 순방향 액세스 세션 (FAS) — IAM 사용자 또는 역할을 사용하여 작업을 수행하는 경우 보안 AWS 주체로 간주됩니다. 일부 서비스를 사용하는 경우 다른 서비스에서 다른 작업을 시작하는 작업을 수행할 수 있습니다. FAS는 전화를 거는 주체의 권한을 다운스트림 AWS 서비스서비스에 AWS 서비스 요청하기 위한 요청과 결합하여 사용합니다. FAS 요청은 다른 서비스 AWS 서비스 또는 리소스와의 상호 작용이 필요한 요청을 서비스가 수신한 경우에만 이루어집니다. 이 경우 두 작업을 모두 수행할 수 있는 권한이 있어야 합니다. FAS 요청 시 정책 세부 정보는 전달 액세스 세션을 참조하세요.
  - 서비스 역할 - 서비스 역할은 서비스가 사용자를 대신하여 태스크를 수행하기 위해 맡는 IAM 역할입니다. IAM 관리자는 IAM 내에서 서비스 역할을 생성, 수정 및 삭제할 수 있습니다. 자세한 정보는 IAM 사용 설명서의 AWS 서비스에 대한 권한을 위임할 역할 생성을 참조하세요.
  - 서비스 연결 역할 — 서비스 연결 역할은 에 연결된 서비스 역할의 한 유형입니다. AWS 서비스서비스는 사용자를 대신하여 작업을 수행하기 위해 역할을 수임할 수 있습니다. 서비스 연결 역할은 사용자에게 AWS 계정 표시되며 해당 서비스가 소유합니다. IAM 관리자는 서비스 링크 역할의 권한을 볼 수 있지만 편집은 할 수 없습니다.
- Amazon EC2에서 실행되는 애플리케이션 — IAM 역할을 사용하여 EC2 인스턴스에서 실행되고 API 요청을 AWS CLI 하는 애플리케이션의 임시 자격 증명을 관리할 수 있습니다. AWS 이는 EC2 인스턴스 내에 액세스 키를 저장할 때 권장되는 방법입니다. EC2 인스턴스에 AWS 역할을 할당하고 모든 애플리케이션에서 사용할 수 있게 하려면 인스턴스에 연결된 인스턴스 프로필을 생성합니다. 인스턴스 프로파일에는 역할이 포함되어 있으며 EC2 인스턴스에서 실행되는 프로그램이 임시 보안 인증을 얻을 수 있습니다. 자세한 정보는 IAM 사용 설명서의 IAM 역할을 사용하여 Amazon EC2 인스턴스에서 실행되는 애플리케이션에 권한 부여를 참조하세요.

IAM 역할을 사용할지 또는 IAM 사용자를 사용할지를 알아보려면 IAM 사용 설명서의 IAM 역할(사용자 대신)을 생성하는 경우를 참조하세요.

정책을 사용한 액세스 관리

정책을 생성하고 이를 AWS ID 또는 리소스에 AWS 연결하여 액세스를 제어할 수 있습니다. 정책은 ID 또는 리소스와 연결될 때 AWS 해당 권한을 정의하는 객체입니다. AWS 주도자 (사용자, 루트 사용자 또는 역할 세션) 가 요청할 때 이러한 정책을 평가합니다. 정책에서 권한은 요청이 허용되거나 거부되는 지를 결정합니다. 대부분의 정책은 JSON 문서로 AWS 저장됩니다. JSON 정책 문서의 구조와 콘텐츠에 대한 자세한 정보는 IAM 사용 설명서의 JSON 정책 개요를 참조하세요.

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

기본적으로, 사용자와 역할에는 어떠한 권한도 없습니다. 사용자에게 사용자가 필요한 리소스에서 작업을 수행할 권한을 부여하려면 IAM 관리자가 IAM 정책을 생성하면 됩니다. 그런 다음 관리자가 IAM 정책을 역할에 추가하고, 사용자가 역할을 수임할 수 있습니다.

IAM 정책은 작업을 수행하기 위해 사용하는 방법과 상관없이 작업에 대한 권한을 정의합니다. 예를 들어, `iam:GetRole`태스크를 허용하는 정책이 있다고 가정합니다. 해당 정책을 사용하는 사용자는 AWS Management Console, AWS CLI, 또는 AWS API에서 역할 정보를 가져올 수 있습니다.

ID 기반 정책

ID 기반 정책은 IAM 사용자, 사용자 그룹 또는 역할과 같은 자격 증명에 연결할 수 있는 JSON 권한 정책 문서입니다. 이러한 정책은 사용자와 역할이 어떤 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는 지를 제어합니다. 자격 증명 기반 정책을 생성하는 방법을 알아보려면 IAM 사용 설명서의 IAM 정책 생성을 참조하세요.

자격 증명 기반 정책은 인라인 정책 또는 관리형 정책으로 한층 더 분류할 수 있습니다. 인라인 정책은 단일 사용자, 그룹 또는 역할에 직접 포함됩니다. 관리형 정책은 내 여러 사용자, 그룹 및 역할에 연결할 수 있는 독립형 정책입니다. AWS 계정관리형 정책에는 AWS 관리형 정책과 고객 관리형 정책이 포함됩니다. 관리형 정책 또는 인라인 정책을 선택하는 방법을 알아보려면 IAM 사용 설명서의 관리형 정책과 인라인 정책의 선택을 참조하세요.

리소스 기반 정책

리소스 기반 정책은 리소스에 연결하는 JSON 정책 설명서입니다. 리소스 기반 정책의 예는 IAM 역할 신뢰 정책과 Amazon S3 버킷 정책입니다. 리소스 기반 정책을 지원하는 서비스에서 서비스 관리자는

이러한 정책을 사용하여 특정 리소스에 대한 액세스를 통제할 수 있습니다. 정책이 연결된 리소스의 경우 정책은 지정된 보안 주체가 해당 리소스와 어떤 조건에서 어떤 작업을 수행할 수 있는 지를 정의합니다. 리소스 기반 정책에서 보안 주체를 지정해야 합니다. 보안 주체에는 계정, 사용자, 역할, 연동 사용자 등이 포함될 수 있습니다. AWS 서비스

리소스 기반 정책은 해당 서비스에 있는 인라인 정책입니다. IAM의 AWS 관리형 정책은 리소스 기반 정책에 사용할 수 없습니다.

액세스 제어 목록(ACLs)

액세스 제어 목록(ACL)은 어떤 보안 주체(계정 멤버, 사용자 또는 역할)가 리소스에 액세스할 수 있는 권한을 가지고 있는지를 제어합니다. ACLs는 JSON 정책 문서 형식을 사용하지 않지만 리소스 기반 정책과 유사합니다.

ACL을 지원하는 서비스의 예로는 아마존 S3와 아마존 VPC가 있습니다. AWS WAF ACL에 대해 자세히 알아보려면 Amazon Simple Storage Service 개발자 안내서의 액세스 제어 목록(ACL) 개요를 참조하세요.

기타 정책 타입

AWS 일반적이지 않은 추가 정책 유형을 지원합니다. 이러한 정책 타입은 더 일반적인 정책 타입에 따라 사용자에게 부여되는 최대 권한을 설정할 수 있습니다.

- 권한 경계 – 권한 경계는 보안 인증 기반 정책에 따라 IAM 엔터티(IAM 사용자 또는 역할)에 부여할 수 있는 최대 권한을 설정하는 고급 기능입니다. 개체에 대한 권한 경계를 설정할 수 있습니다. 그 결과로 얻는 권한은 엔터티의 자격 증명 기반 정책과 그 권한 경계의 교집합입니다. `Principal` 필드에서 사용자나 역할을 보안 주체로 지정하는 리소스 기반 정책은 권한 경계를 통해 제한되지 않습니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 권한 경계에 대한 자세한 정보는 IAM 사용 설명서의 IAM 엔터티에 대한 권한 경계를 참조하세요.
- 서비스 제어 정책 (SCP) - SCP는 조직 또는 조직 단위 (OU) 에 대한 최대 권한을 지정하는 JSON 정책입니다. AWS Organizations AWS Organizations 사업체가 소유한 여러 AWS 계정 개를 그룹화하고 중앙에서 관리하는 서비스입니다. 조직에서 모든 기능을 활성화할 경우 서비스 제어 정책(SCP)을 임의의 또는 모든 계정에 적용할 수 있습니다. SCP는 구성원 계정의 엔티티 (각 엔티티 포함) 에 대한 권한을 제한합니다. AWS 계정 루트 사용자조직 및 SCP에 대한 자세한 정보는 AWS Organizations 사용 설명서의 SCP 작동 방식을 참조하세요.
- 세션 정책 – 세션 정책은 역할 또는 페더레이션 사용자에 대해 임시 세션을 프로그래밍 방식으로 생성할 때 파라미터로 전달하는 고급 정책입니다. 결과적으로 얻는 세션의 권한은 사용자 또는 역할 자격 증명 기반 정책의 교차 및 세션 정책입니다. 또한 권한을 리소스 기반 정책에서 가져올 수도 있습

니다. 이러한 정책 중 하나에 포함된 명시적 거부는 허용을 재정의합니다. 자세한 정보는 IAM 사용 설명서의 세션 정책을 참조하세요.

여러 정책 타입

여러 정책 타입이 요청에 적용되는 경우 결과 권한은 이해하기가 더 복잡합니다. 여러 정책 유형이 관련되어 있을 때 요청을 허용할지 여부를 AWS 결정하는 방법을 알아보려면 IAM 사용 설명서의 정책 평가 로직을 참조하십시오.

IAM의 AWS Transfer Family 작동 방식

AWS Identity and Access Management (IAM) 을 사용하여 액세스를 관리하려면 먼저 어떤 IAM 기능과 함께 사용할 수 있는지 이해해야 합니다. AWS Transfer Family AWS Transfer Family기타 AWS 서비스가 AWS Transfer Family IAM과 연동되는 방식을 자세히 알아보려면 IAM 사용 설명서에서 IAM과 연동되는AWS 서비스를 참조하십시오.

주제

- AWS Transfer Family 보안 인증 기반 정책
- AWS Transfer Family 리소스 기반 정책
- AWS Transfer Family 태그 기반 인증
- AWS Transfer Family IAM 역할

AWS Transfer Family 보안 인증 기반 정책

IAM 보안 인증 기반 정책을 사용하면 허용되거나 거부되는 작업과 리소스 및 작업이 허용되거나 거부되는 조건을 지정할 수 있습니다. AWS Transfer Family 은(는) 특정 작업, 리소스 및 조건 키를 지원합니다. JSON 정책에서 사용하는 모든 요소에 대해 알아보려면 AWS Identity and Access Management 사용자 가이드의 IAM JSON 정책 요소 참조를 참조하세요.

작업

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지를 지정할 수 있습니다.

JSON 정책의 `Action`요소는 정책에서 액세스를 허용하거나 거부하는 데 사용할 수 있는 태스크를 설명합니다. 정책 작업은 일반적으로 관련 AWS API 작업과 이름이 같습니다. 일치하는 API 작업이 없는 권한 전용 작업 같은 몇 가지 예외도 있습니다. 정책에서 여러 작업이 필요한 몇 가지 작업도 있습니다. 이러한 추가 작업을 일컬어 종속 작업이라고 합니다.

연결된 작업을 수행할 수 있는 권한을 부여하기 위한 정책에 작업을 포함하십시오.

정책 조치는 조치 앞에 다음 접두사를 AWS Transfer Family 사용합니다. `transfer:` 예를 들어 Transfer Family `CreateServer` API 작업으로 서버를 생성할 수 있는 권한을 부여하려면 해당 정책에 `transfer:CreateServer` 작업을 포함합니다. 정책 명령문에는 `Action` 또는 `NotAction` 요소가 포함되어야 합니다. AWS Transfer Family 는 이 서비스로 수행할 수 있는 작업을 설명하는 고유한 작업 집합을 정의합니다.

명령문 하나에 여러 작업을 지정하려면 다음과 같이 쉼표로 구분합니다.

```
"Action": [
      "transfer:action1",
      "transfer:action2"
```

와일드카드(*)를 사용하여 여러 작업을 지정할 수 있습니다. 예를 들어, `Describe`라는 단어로 시작하는 모든 작업을 지정하려면 다음 작업을 포함합니다.

```
"Action": "transfer:Describe*"
```

AWS Transfer Family 작업 목록을 보려면 서비스 권한 부여 AWS Transfer Family참조에 정의된 작업을 참조하십시오.

리소스

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지 지정할 수 있습니다.

`Resource` JSON 정책 요소는 작업이 적용되는 하나 이상의 개체를 지정합니다. 문장에는 `Resource`또는 `NotResource`요소가 반드시 추가되어야 합니다. 모범 사례에 따라 Amazon 리소스 이름(ARN)을 사용하여 리소스를 지정합니다. 리소스 수준 권한이라고 하는 특정 리소스 타입을 지원하는 작업에 대해 이 작업을 수행할 수 있습니다.

작업 나열과 같이 리소스 수준 권한을 지원하지 않는 작업의 경우, 와일드카드(*)를 사용하여 해당 문이 모든 리소스에 적용됨을 나타냅니다.

```
"Resource": "*"
```

Transfer Family 서버 리소스에는 다음 ARN이 있습니다.

```
arn:aws:transfer:${Region}:${Account}:server/${ServerId}
```

예를 들어, 명령문에 s-01234567890abcdef Transfer Family 서버를 지정하려면 다음 ARN을 사용합니다.

```
"Resource": "arn:aws:transfer:us-east-1:123456789012:server/s-01234567890abcdef"
```

ARN 형식에 대한 자세한 설명은 서비스 승인 참조에서 Amazon 리소스 이름(ARN) 또는 IAM 사용자 가이드에서 IAM ARN을 참조하세요.

특정 계정에 속하는 모든 인스턴스를 지정하려면 와일드카드(*)를 사용합니다.

```
"Resource": "arn:aws:transfer:us-east-1:123456789012:server/*"
```

일부 AWS Transfer Family 작업은 IAM 정책에 사용되는 것과 같은 여러 리소스에서 수행됩니다. 이러한 경우, 와일드카드(*)를 사용해야 합니다.

```
"Resource": "arn:aws:transfer:*:123456789012:server/*"
```

Transfer Family 서버 및 사용자에 대한 액세스를 허용하는 정책을 만드는 경우와 같이 둘 이상의 리소스 타입을 지정해야 하는 경우도 있습니다. 단일 명령문에서 여러 리소스를 지정하려면 ARN을 쉼표로 구분합니다.

```
"Resource": [
      "resource1",
      "resource2"
            ]
```

AWS Transfer Family 리소스 목록을 보려면 서비스 권한 부여 AWS Transfer Family참조에 정의된 리소스 유형을 참조하십시오.

조건 키

관리자는 AWS JSON 정책을 사용하여 누가 무엇에 액세스할 수 있는지 지정할 수 있습니다. 즉, 어떤 보안 주체가 어떤 리소스와 어떤 조건에서 작업을 수행할 수 있는지 지정할 수 있습니다.

Condition 요소(또는 Condition 블록)를 사용하면 정책이 발효되는 조건을 지정할 수 있습니다. Condition 요소는 옵션입니다. 같거나 작음과 같은 조건 연산자를 사용하여 정책의 조건을 요청의 값과 일치시키는 조건식을 생성할 수 있습니다.

한 문에서 여러 Condition요소를 지정하거나 단일 Condition요소에서 여러 키를 지정하는 경우 AWS 는 논리적 AND태스크를 사용하여 평가합니다. 단일 조건 키에 여러 값을 지정하는 경우 는 논리

적 OR 연산을 사용하여 조건을 AWS 평가합니다. 명문의 권한을 부여하기 전에 모든 조건을 충족해야 합니다.

조건을 지정할 때 자리 표시자 변수를 사용할 수도 있습니다. 예를 들어, IAM 사용자에게 IAM 사용자 이름으로 태그가 지정된 경우에만 리소스에 액세스할 수 있는 권한을 부여할 수 있습니다. 자세한 정보는 IAM 사용 설명서의 IAM 정책 요소: 변수 및 태그를 참조하세요.

AWS 글로벌 조건 키 및 서비스별 조건 키를 지원합니다. 모든 AWS 글로벌 조건 키를 보려면 IAM 사용 AWS 설명서의 글로벌 조건 컨텍스트 키를 참조하십시오.

AWS Transfer Family 자체 조건 키 세트를 정의하며 일부 글로벌 조건 키 사용도 지원합니다. AWS Transfer Family 조건 키 목록을 보려면 서비스 권한 부여 AWS Transfer Family참조의 조건 키를 참조하십시오.

예제

AWS Transfer Family ID 기반 정책의 예를 보려면 을 참조하십시오. AWS Transfer Family ID 기반 정책 예제

AWS Transfer Family 리소스 기반 정책

리소스 기반 정책은 지정된 보안 주체가 리소스에서 수행할 수 있는 작업과 조건을 지정하는 JSON 정책 문서입니다. AWS Transfer Family Amazon S3은 Amazon S3 ##에 대한 리소스 기반 권한 정책을 지원합니다. 리소스 기반 정책을 사용하여 리소스별로 다른 계정에 사용 권한을 부여할 수 있습니다. *## ### ## ### #### AWS #### Amazon S3 ### ###### ### # ####.*

크로스 계정 액세스를 활성화하려는 경우, 전체 계정이나 다른 계정의 IAM 엔터티를 리소스 기반 정책의 보안 주체로 지정할 수 있습니다. 리소스 기반 정책에 크로스 계정 보안 주체를 추가하는 것은 트러스트 관계 설정의 절반밖에 되지 않는다는 것을 유념하세요. 보안 주체와 리소스가 서로 다른 AWS 계정에 있는 경우 보안 주체에 리소스에 액세스할 수 있는 권한도 부여해야 합니다. 엔터티에 보안 인증 기반 정책을 연결하여 권한을 부여합니다. 하지만 리소스 기반 정책이 동일 계정의 보안 주체에 액세스를 부여하는 경우 추가 보안 인증 기반 정책이 필요하지 않습니다. 자세한 내용은 AWS Identity and Access Management 사용자 가이드의 IAM 역할과 리소스 기반 정책의 차이를 참조하세요.

Amazon ECR 서비스는 리소스 기반 정책 중 한 가지 타입만 지원하는데, 이 정책 타입은 ## 정책이라고 하며 ##에 연결되어 있습니다. 이 정책은 객체에서 작업을 수행할 수 있는 보안 주체 엔터티(계정, 사용자, 역할 및 연동 사용자)를 정의합니다.

예제

AWS Transfer Family 리소스 기반 정책의 예를 보려면 을 참조하십시오. AWS Transfer Family 태그 기반 정책 예제

AWS Transfer Family 태그 기반 인증

AWS Transfer Family 리소스에 태그를 첨부하거나 요청에 태그를 전달할 수 있습니다. AWS Transfer Family태그를 기반으로 액세스를 제어하려면 `transfer:ResourceTag/key-name`, `aws:RequestTag/key-name` 또는 `aws:TagKeys` 조건 키를 사용하여 정책의 조건 요소에 태그 정보를 제공합니다. 태그를 사용하여 AWS Transfer Family 리소스에 대한 액세스를 제어하는 방법에 대한 자세한 내용은 을 참조하십시오AWS Transfer Family 태그 기반 정책 예제.

AWS Transfer Family IAM 역할

IAM 역할은 AWS 계정 내에서 특정 권한을 가진 엔티티입니다.

임시 자격 증명 사용: AWS Transfer Family

임시 보안 인증을 사용하여 페더레이션을 통해 로그인하거나, IAM 역할을 맡거나, 교차 계정 역할을 맡을 수 있습니다. AssumeRole또는 GetFederationToken과 같은 AWS STS API 작업을 호출하여 임시 보안 자격 증명을 얻습니다.

AWS Transfer Family 임시 자격 증명 사용을 지원합니다.

AWS Transfer Family ID 기반 정책 예제

기본적으로 IAM 사용자 및 역할은 AWS Transfer Family 리소스를 생성하거나 수정할 수 있는 권한이 없습니다. 또한 AWS Management Console AWS CLI, 또는 AWS API를 사용하여 작업을 수행할 수 없습니다. IAM 관리자는 지정된 리소스에서 특정 API 작업을 수행할 수 있는 권한을 사용자와 역할에게 부여하는 IAM 정책을 생성해야 합니다. 그런 다음 관리자는 해당 권한이 필요한 IAM 사용자 또는 그룹에 이러한 정책을 연결해야 합니다.

이러한 예 JSON 정책 설명서를 사용하여 IAM 보안 인증 기반 정책을 생성하는 방법을 알아보려면 AWS Identity and Access Management 사용자 가이드의 JSON 탭에서 정책 생성을 참조하세요.

주제

- 정책 모범 사례
- AWS Transfer Family 콘솔 사용
- 사용자가 자신이 권한을 볼 수 있도록 허용

정책 모범 사례

ID 기반 정책은 누군가가 계정에서 AWS Transfer Family 리소스를 생성, 액세스 또는 삭제할 수 있는지 여부를 결정합니다. 이 작업으로 인해 AWS 계정에 비용이 발생할 수 있습니다. 자격 증명 기반 정책을 생성하거나 편집할 때는 다음 지침과 권장 사항을 따르십시오.

- AWS 관리형 정책으로 시작하여 최소 권한 권한으로 이동 — 사용자와 워크로드에 권한을 부여하려면 여러 일반적인 사용 사례에 권한을 부여하는 AWS 관리형 정책을 사용하세요. 해당 내용은 에서 사용할 수 있습니다. AWS 계정사용 사례에 맞는 AWS 고객 관리형 정책을 정의하여 권한을 더 줄이는 것이 좋습니다. 자세한 정보는 IAM 사용 설명서의 AWS 관리형 정책 또는 AWS 직무에 대한 관리형 정책을 참조하세요.
- 최소 권한 적용 – IAM 정책을 사용하여 권한을 설정하는 경우 태스크를 수행하는 데 필요한 권한만 부여합니다. 이렇게 하려면 최소 권한으로 알려진 특정 조건에서 특정 리소스에 대해 수행할 수 있는 작업을 정의합니다. IAM을 사용하여 권한을 적용하는 방법에 대한 자세한 정보는 IAM 사용 설명서에 있는 IAM의 정책 및 권한을 참조하세요.
- IAM 정책의 조건을 사용하여 액세스 추가 제한 – 정책에 조건을 추가하여 작업 및 리소스에 대한 액세스를 제한할 수 있습니다. 예를 들어 SSL을 사용하여 모든 요청을 전송해야 한다고 지정하는 정책 조건을 작성할 수 있습니다. 예를 AWS 서비스들어 특정 작업을 통해 서비스 작업을 사용하는 경우 조건을 사용하여 서비스 작업에 대한 액세스 권한을 부여할 수도 AWS CloudFormation있습니다. 자세한 정보는 IAM 사용 설명서의 IAM JSON 정책 요소: 조건을 참조하세요.
- IAM Access Analyzer를 통해 IAM 정책을 검증하여 안전하고 기능적인 권한 보장 – IAM Access Analyzer에서는 IAM 정책 언어(JSON)와 모범 사례가 정책에서 준수되도록 신규 및 기존 정책을 검증합니다. IAM Access Analyzer는 100개 이상의 정책 확인 항목과 실행 가능한 추천을 제공하여 안전하고 기능적인 정책을 작성하도록 돕습니다. 자세한 정보는 IAM 사용 설명서의 IAM Access Analyzer 정책 검증을 참조하세요.
- 멀티 팩터 인증 (MFA) 필요 - IAM 사용자 또는 루트 사용자가 필요한 시나리오가 있는 경우 추가 보안을 위해 AWS 계정 MFA를 활성화하십시오. API 작업을 직접 호출할 때 MFA가 필요하면 정책에 MFA 조건을 추가합니다. 자세한 정보는 IAM 사용 설명서의 MFA 보호 API 액세스 구성을 참조하세요.

IAM의 모범 사례에 대한 자세한 내용은 IAM 사용 설명서의 IAM의 보안 모범 사례를 참조하세요.

AWS Transfer Family 콘솔 사용

AWS Transfer Family 콘솔에 액세스하려면 최소한의 권한이 있어야 합니다. 이러한 권한을 통해 AWS 계정의 AWS Transfer Family 리소스에 대한 세부 정보를 나열하고 볼 수 있어야 합니다. 최소 필수 권한보다 더 제한적인 보안 인증 기반 정책을 만들면 콘솔이 해당 정책에 연결된 개체(IAM 사용자 또는

역할)에 대해 의도대로 작동하지 않습니다. 자세한 내용은 AWS Identity and Access Management 사용자 가이드의 사용자에게 권한 추가를 참조하세요.

AWS CLI 또는 AWS API만 호출하는 사용자에게 최소 콘솔 권한을 허용할 필요는 없습니다. 그 대신, 수행하려는 API 작업과 일치하는 작업에만 액세스할 수 있도록 합니다.

사용자가 자신이 권한을 볼 수 있도록 허용

이 예시는 IAM 사용자가 자신의 사용자 자격 증명에 연결된 인라인 및 관리형 정책을 볼 수 있도록 허용하는 정책을 생성하는 방법을 보여줍니다. 이 정책에는 콘솔에서 AWS CLI 또는 AWS API를 사용하여 프로그래밍 방식으로 이 작업을 완료할 수 있는 권한이 포함됩니다.

```
{
    "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 태그 기반 정책 예제

다음은 태그를 기반으로 AWS Transfer Family 리소스에 대한 액세스를 제어하는 방법의 예입니다.

태그를 사용하여 AWS Transfer Family 리소스에 대한 액세스 통제

IAM 정책의 조건은 AWS Transfer Family 리소스에 대한 권한을 지정하는 데 사용하는 구문의 일부입니다. AWS Transfer Family 리소스의 태그를 기반으로 리소스 (예: 사용자, 서버, 역할 및 기타 엔티티)에 대한 액세스를 제어할 수 있습니다. 태그는 키/값 쌍입니다. 리소스에 태그를 지정하는 방법에 대한 자세한 내용은 의 AWS 리소스 태그 지정을 참조하십시오. AWS 일반 참조

AWS Transfer Family에서는 리소스에 태그가 있을 수 있으며 일부 작업에는 태그가 포함될 수 있습니다. IAM 정책을 생성하면 태그 조건 키를 사용하여 다음을 통제할 수 있습니다.

- 리소스에 있는 태그를 기반으로 AWS Transfer Family 리소스에서 작업을 수행할 수 있는 사용자
- 어떤 태그가 작업의 요청에서 전달될 수 있는지 통제합니다.
- 요청에서 특정 키를 사용할 수 있는지 여부를 통제합니다.

태그 기반 액세스 통제를 사용하면 API 수준에서보다 더 세밀한 통제를 적용할 수 있습니다. 또한 리소스 기반 액세스 통제를 사용하는 것보다 더 동적인 통제를 적용할 수 있습니다. 요청에서 제공된 태그(요청 태그)를 기반으로 작업을 허용하거나 거부하는 IAM 정책을 생성할 수 있습니다. 작업 중인 리소스에 대한 태그(리소스 태그)를 기반으로 IAM 정책을 생성할 수 있습니다. 일반적으로 리소스 태그는 리소스에 이미 있는 태그용이고, 요청 태그는 리소스에 태그를 추가하거나 리소스에서 태그를 제거할 때 사용됩니다.

태그 조건 키의 전체 구문 및 의미는 IAM 사용자 가이드의 AWS 리소스 태그를 사용한 액세스 통제를 참조하세요. API Gateway로 IAM 정책을 지정하는 방법에 대한 자세한 설명은 API Gateway 개발자 가이드의 IAM 권한으로 API에 대한 액세스 통제를 참조하세요.

예 1: 리소스 태그 기반 작업 거부

태그를 기반으로 리소스에서 수행할 작업을 거부할 수 있습니다. 다음 예 정책은 사용자 또는 서버 리소스에 키 `stage`와(과) 값 `prod`이(가) 태그 지정된 경우 `TagResource`, `UntagResource`, `StartServer`, `StopServer`, `DescribeServer`, `DescribeUser` 작업을 거부합니다.

```
{
    "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"
                }
            }
        }
    ]
}
```

예 2: 리소스 태그 기반 작업 허용

태그를 기반으로 리소스에서 작업을 수행하도록 허용할 수 있습니다. 다음 예 정책은 사용자 또는 서버 리소스에 키 `stage`와(과) 값 `prod`이(가) 태그 지정된 경우 `TagResource`, `UntagResource`, `StartServer`, `StopServer`, `DescribeServer`, `DescribeUser` 작업을 거부합니다.

```
{
    "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"
                }
            }
        }
    ]
}
```

예 3: 요청 태그에 따른 사용자 또는 서버 생성 거부

다음의 정책 예에는 명령문이 2개입니다. 첫 번째 명령문은 태그의 코스트 센터 키에 값이 없는 경우 모든 리소스에 대한 `CreateServer` 작업을 거부합니다.

두 번째 명령문은 태그의 코스트 센터 키에 1, 2 또는 3을 제외한 다른 값이 포함된 경우 `CreateServer` 작업을 거부합니다.

> **Note**
>
> 이 정책은 1, 2, 또는 3 라는 `costcenter` 키와 값을 포함하는 리소스를 만들거나 삭제할 수 있도록 허용합니다.

```
{
    "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"
                    ]
                }
            }
        }
    ]
}
```

AWS Transfer Family ID 및 액세스 문제 해결

다음 정보를 사용하면 및 IAM을 사용할 때 발생할 수 있는 일반적인 문제를 AWS Transfer Family 진단하고 해결하는 데 도움이 됩니다.

주제

- 저는 다음과 같은 작업을 수행할 권한이 없습니다. AWS Transfer Family
- 저는 IAM을 수행할 권한이 없습니다. PassRole
- 내 AWS 계정 외부의 사용자가 내 AWS Transfer Family 리소스에 액세스할 수 있도록 허용하고 싶습니다.

저는 다음과 같은 작업을 수행할 권한이 없습니다. AWS Transfer Family

작업을 수행할 권한이 없다는 AWS Management Console 메시지가 표시되면 관리자에게 도움을 요청해야 합니다. 관리자는 로그인 보안 인증 정보를 제공한 사람입니다.

다음 예 오류는 `mateojackson` IAM 사용자가 콘솔을 사용하여 *##*에 대한 세부 정보를 보려고 하지만 `transfer:`*`GetWidget`* 권한이 없는 경우에 발생합니다.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 transfer:GetWidget on resource: my-example-widget
```

이 경우 Mateo는 *`my-example-widget`* 태스크를 사용하여 `transfer;:`*`GetWidget`* 리소스에 액세스하도록 허용하는 정책을 업데이트하라고 관리자에게 요청합니다.

저는 IAM을 수행할 권한이 없습니다. PassRole

`iam:PassRole` 작업을 수행할 수 있는 권한이 없다는 오류가 수신되면 AWS Transfer Family에 역할을 전달할 수 있도록 정책을 업데이트해야 합니다.

새 서비스 역할 또는 서비스 연결 역할을 만드는 대신 기존 역할을 해당 서비스에 전달할 AWS 서비스 수 있는 기능도 있습니다. 이렇게 하려면 사용자가 서비스에 역할을 전달할 수 있는 권한을 가지고 있어야 합니다.

다음 예 오류는 `marymajor`라는 IAM 사용자가 콘솔을 사용하여 AWS Transfer Family에서 작업을 수행하려고 하는 경우에 발생합니다. 하지만 작업을 수행하려면 서비스 역할이 부여한 권한이 서비스에 있어야 합니다. Mary는 서비스에 역할을 전달할 수 있는 권한을 가지고 있지 않습니다.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

이 경우 Mary가 `iam:PassRole`작업을 수행할 수 있도록 Mary의 정책을 업데이트해야 합니다.

도움이 필요하면 관리자에게 문의하세요. AWS 관리자는 로그인 자격 증명을 제공한 사람입니다.

다음 예 정책에는 역할을 AWS Transfer Family에 전달할 권한이 포함되어 있습니다.

```
{
   "Version": "2012-10-17",
   "Statement": [
      { "Action": "iam:PassRole",
        "Resource": "arn:aws::iam::123456789012:role/*",
        "Effect": "Allow"
      }
   ]
}
```

내 AWS 계정 외부의 사용자가 내 AWS Transfer Family 리소스에 액세스할 수 있도록 허용하고 싶습니다.

다른 계정의 사용자 또는 조직 외부의 사람이 리소스에 액세스할 때 사용할 수 있는 역할을 생성할 수 있습니다. 역할을 수임할 신뢰할 수 있는 사람을 지정할 수 있습니다. 리소스 기반 정책 또는 액세스 제어 목록(ACL)을 지원하는 서비스의 경우 이러한 정책을 사용하여 다른 사람에게 리소스에 대한 액세스 권한을 부여할 수 있습니다.

자세히 알아보려면 다음을 참조하세요.

- 이러한 기능의 AWS Transfer Family 지원 여부를 알아보려면 을 참조하십시오IAM의 AWS Transfer Family 작동 방식.
- 소유한 리소스에 대한 액세스 권한을 AWS 계정 부여하는 방법을 알아보려면 IAM 사용 설명서에서 자신이 소유한 다른 AWS 계정 IAM 사용자에게 액세스 권한 제공을 참조하십시오.
- 제3자에게 리소스에 대한 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 타사 AWS 계정AWS 계정 소유에 대한 액세스 제공을 참조하십시오.
- ID 페더레이션을 통해 액세스 권한을 제공하는 방법을 알아보려면 IAM 사용 설명서의 외부에서 인증된 사용자에게 액세스 권한 제공(ID 페더레이션)을 참조하세요.
- 크로스 계정 액세스를 위한 역할과 리소스 기반 정책 사용의 차이점을 알아보려면 IAM 사용 설명서의 IAM 역할과 리소스 기반 정책의 차이를 참조하세요.

규정 준수 검증: AWS Transfer Family

제3자 감사자는 여러 규정 AWS 준수 프로그램의 AWS Transfer Family 일환으로 보안 및 규정 준수를 평가합니다. 여기에는 SOC, PCI, HIPAA 등이 포함됩니다. 전체 목록은 규정 준수 프로그램별 범위 내 AWS 서비스를 참조하십시오.

특정 규정 준수 프로그램 범위 내 AWS 서비스 목록은 규정 준수 프로그램별 범위 내AWS 서비스를 참조하십시오. 일반 정보는 AWS 규정 준수 프로그램을 참조하세요.

를 사용하여 타사 감사 보고서를 다운로드할 수 AWS Artifact있습니다. 자세한 내용은 에서 보고서 다운로드를 참조하십시오 AWS Artifact.

사용 시 규정 준수 AWS Transfer Family 책임은 데이터의 민감도, 회사의 규정 준수 목표, 관련 법률 및 규정에 따라 결정됩니다. AWS 규정 준수에 도움이 되는 다음 리소스를 제공합니다.

- 보안 및 규정 준수 퀵 스타트 가이드 - 이 배포 가이드에서는 아키텍처 고려 사항을 설명하고 보안 및 규정 준수에 중점을 둔 기본 환경을 배포하기 위한 단계를 제공합니다. AWS
- HIPAA 보안 및 규정 준수를 위한 설계 백서 — 이 백서는 기업이 HIPAA 준수 애플리케이션을 개발하는 데 사용할 수 있는 방법을 설명합니다. AWS
- AWS 규정 준수 리소스 – 사용자의 업계와 위치에 해당할 수 있는 워크북 및 가이드 모음입니다.
- AWS Config— 이 AWS 서비스는 리소스 구성이 내부 관행, 업계 지침 및 규정을 얼마나 잘 준수하는지 평가합니다.
- AWS Security Hub— 이 AWS 서비스는 보안 업계 표준 및 모범 사례를 준수하는지 확인하는 데 도움이 AWS 되는 보안 상태를 종합적으로 보여줍니다.

레질리언스: AWS Transfer Family

AWS 글로벌 인프라는 AWS 지역 및 가용 영역을 중심으로 구축됩니다. AWS 지역은 물리적으로 분리되고 격리된 여러 가용 영역을 제공하며, 이러한 가용 영역은 지연 시간이 짧고 처리량이 높으며 중복성이 높은 네트워킹으로 연결됩니다. 가용 영역을 사용하면 중단 없이 가용 영역 간에 자동으로 장애 조치가 이루어지는 애플리케이션 및 데이터베이스를 설계하고 운영할 수 있습니다. 가용 영역은 기존의 단일 또는 복수 데이터 센터 인프라보다 가용성, 내결함성, 확장성이 뛰어납니다.

AWS Transfer Family 최대 3개의 가용 영역을 지원하며 연결 및 전송 요청을 위한 Auto Scaling, 중복 플릿으로 뒷받침됩니다.

유의할 사항:

- 퍼블릭 엔드포인트의 경우:
  - 가용 영역 수준의 이중화가 서비스에 내장되어 있습니다.
  - 각 AZ에는 중복 플릿이 있습니다.
  - 이 중복성은 자동으로 제공됩니다.
- Virtual Private Cloud(VPC) 엔드포인트 지원에 대한 내용은 Virtual Private Cloud(VPC)에 서버 생성를 참조하세요.

참고 항목

- 가용 영역에 대한 AWS 리전 자세한 내용은 AWS 글로벌 인프라를 참조하십시오.
- 지연 시간 기반 라우팅을 사용하여 중복성을 높이고 네트워크 지연 시간을 최소화하는 방법에 대한 예는 블로그 게시물 서버의 네트워크 지연 시간 최소화를 참조하십시오. AWS Transfer Family

인프라 보안: AWS Transfer Family

관리형 서비스로서 AWS 글로벌 네트워크 보안으로 AWS Transfer Family 보호됩니다. AWS 보안 서비스 및 인프라 AWS 보호 방법에 대한 자세한 내용은 AWS 클라우드 보안을 참조하십시오. 인프라 보안 모범 사례를 사용하여 AWS 환경을 설계하려면 Security Pillar AWS Well-Architected Framework의 인프라 보호를 참조하십시오.

AWS 게시된 API 호출을 사용하여 네트워크를 통해 액세스할 AWS Transfer Family 수 있습니다. 고객은 다음을 지원해야 합니다.

- 전송 계층 보안(TLS) TLS 1.2는 필수이며 TLS 1.3을 권장합니다.

- DHE(Ephemeral Diffie-Hellman) 또는 ECDHE(Elliptic Curve Ephemeral Diffie-Hellman)와 같은 완전 전송 보안(PFS)이 포함된 암호 제품군 Java 7 이상의 최신 시스템은 대부분 이러한 모드를 지원합니다.

또한 요청은 액세스 키 ID 및 IAM 주체와 관련된 비밀 액세스 키를 사용하여 서명해야 합니다. 또는 AWS Security Token Service(AWS STS)를 사용하여 임시 보안 보안 인증을 생성하여 요청에 서명할 수 있습니다.

웹 애플리케이션 방화벽 추가

AWS WAF 웹 애플리케이션 및 API를 공격으로부터 보호하는 데 도움이 되는 웹 애플리케이션 방화벽입니다. 사용자가 정의한 맞춤형 웹 보안 규칙 및 조건에 따라 웹 요청을 허용, 차단 또는 계산하는 규칙 집합(웹 액세스 통제 목록 또는 웹 ACL)을 구성할 수 있습니다. 자세한 내용은 API 보호를 AWS WAF 위한 사용을 참조하십시오.

추가하려면 AWS WAF

1. https://console.aws.amazon.com/apigateway/에서 Amazon API Gateway 콘솔을 엽니다.
2. API 탐색 창에서 맞춤 ID 제공자 템플릿을 선택합니다.
3. 단계를 선택합니다.
4. 단계(Stages) 창에서 단계 명칭을 선택합니다.
5. Stage Editor(단계 편집기) 창에서 설정 탭을 선택합니다.
6. 다음 중 하나를 수행하십시오.

   - 웹 애플리케이션 방화벽(WAF)에서 이 단계와 연계할 지역 웹 ACL을 선택합니다.
   - 필요한 웹 ACL이 없는 경우 다음과 같이 웹 ACL을 생성해야 합니다:
     1. Create web ACL(웹 ACL 생성)을 선택합니다.
     2. AWS WAF 서비스 홈페이지에서 웹 ACL 생성을 선택합니다.
     3. 웹 ACL 세부 정보에서 명칭에 웹 ACL의 명칭을 입력합니다.
     4. 규칙에서 규칙 추가를 선택한 다음 나만의 규칙 및 규칙 그룹 추가를 선택합니다.
     5. 규칙 타입에서 IP 세트를 선택하여 특정 IP 주소 목록을 식별합니다.
     6. 규칙에는 규칙 명칭을 입력합니다.
     7. IP 세트의 경우 기존 IP 세트를 선택합니다. IP 집합을 생성하려면 IP 세트 생성을 참조하세요.
     8. 기원 주소로 사용할 IP 주소의 경우 헤더의 IP 주소를 선택합니다.

9. 헤더 필드 명칭에 SourceIP을 입력합니다.
10. 헤더 내 위치의 경우 첫 번째 IP 주소를 선택합니다.
11. 누락된 IP 주소를 위한 대처를 위해, 헤더에서 유효하지 않거나 누락된 IP 주소를 처리하려는 방법에 따라 일치 또는 불일치를 선택합니다.
12. 조치에서 IP 세트 조치를 선택합니다.
13. 어떤 규칙과도 일치하지 않는 요청에 대한 기본 웹 ACL 동작의 경우 허용 또는 차단을 선택한 후 다음을 클릭합니다.
14. 4단계와 5단계에서 다음을 선택합니다.
15. 검토 및 생성에서 옵션을 검토한 다음 웹 ACL 생성을 선택합니다.

7. 변경 사항 저장(Save Changes)을 선택합니다.
8. 리소스를 선택합니다.
9. 작업 및 API 배포를 선택합니다.

AWS 웹 애플리케이션 방화벽을 AWS Transfer Family 통한 보안 방법에 대한 자세한 내용은 AWS 스토리지 블로그의 AWS 애플리케이션 방화벽 및 Amazon API Gateway를 AWS Transfer Family 통한 보안을 참조하십시오.

교차 서비스 혼동된 대리인 방지

혼동된 대리자 문제는 작업을 수행할 권한이 없는 엔터티가 권한이 더 많은 엔터티에게 작업을 수행하도록 강요할 수 있는 보안 문제입니다. 에서 AWS서비스 간 사칭으로 인해 대리인 문제가 혼동될 수 있습니다. 교차 서비스 가장은 한 서비스(직접 호출하는 서비스)가 다른 서비스(직접 호출되는 서비스)를 직접 호출할 때 발생할 수 있습니다. 호출하는 서비스는 다른 고객의 리소스에 대해 액세스 권한이 없는 방식으로 작동하게 권한을 사용하도록 조작될 수 있습니다. 이를 방지하기 위해 AWS 에서는 계정의 리소스에 대한 액세스 권한이 부여된 서비스 보안 주체를 사용하여 모든 서비스에 대한 데이터를 보호하는 데 도움이 되는 도구를 제공합니다. 이 문제에 대한 자세한 설명은 IAM 사용자 가이드의 혼동되는 대리 문제를 참조하세요.

리소스 정책에서 aws:SourceArn및 aws:SourceAccount글로벌 조건 컨텍스트 키를 사용하여 AWS Transfer Family가 해당 리소스에 대해 갖는 권한을 제한하는 것이 좋습니다. 두 글로벌 조건 컨텍스트 키를 모두 사용하는 경우 aws:SourceAccount 값과 aws:SourceArn 값의 계정은 동일한 정책 명령문에서 사용할 경우 동일한 계정 ID를 사용해야 합니다.

혼동된 대리자 문제로부터 보호하는 가장 효과적인 방법은 허용하려는 리소스의 정확한 전체 Amazon 리소스 이름(ARN)을 사용하는 것입니다. 리소스의 전체 ARN을 모르거나 여러 리소스를 지정하는 경

우, ARN의 알 수 없는 부분에 대해 와일드카드 문자(*)를 포함한 `aws:SourceArn` 글로벌 조건 컨텍스트 키를 사용합니다. 예를 들어 `arn:aws:transfer::region::account-id:server/*`입니다.

AWS Transfer Family는 다음과 같은 유형의 역할을 사용합니다.

- 사용자 역할 — 서비스 관리 사용자가 필요한 Transfer Family 리소스에 액세스할 수 있습니다. AWS Transfer Family는 Transfer Family 사용자 ARN의 컨텍스트에서 이 역할을 맡습니다.
- 액세스 역할 – 전송 중인 Amazon S3 파일에만 액세스할 수 있습니다. 인바운드 AS2 전송의 경우 액세스 역할은 계약의 Amazon 리소스 이름(ARN)을 사용합니다. 아웃바운드 AS2 전송의 경우 액세스 역할은 커넥터의 ARN을 사용합니다.
- 간접 호출 역할 - Amazon API Gateway와 함께 서버의 사용자 지정 자격 증명 공급자로 사용합니다. Transfer Family는 Transfer Family 서버 ARN의 컨텍스트에서 이 역할을 담당합니다.
- 로깅 역할 — Amazon에 항목을 기록하는 데 사용됩니다 CloudWatch. Transfer Family는 이 역할을 사용하여 File Transfer에 대한 정보와 함께 성공 및 실패 세부 정보를 기록합니다. Transfer Family는 Transfer Family 서버 ARN의 컨텍스트에서 이 역할을 담당합니다. 아웃바운드 AS2 전송의 경우, 로깅 역할은 커넥터 ARN을 사용합니다.
- 실행 역할 – Transfer Family 사용자가 전화를 걸어 워크플로를 시작할 수 있습니다. Transfer Family는 Transfer Family 워크플로 ARN의 컨텍스트에서 이 역할을 담당합니다.

자세한 내용은 IAM 사용자 가이드에서 IAM의 정책 및 권한을 참조하세요.

> **Note**
>
> 다음 예에서는 자신의 정보로 각각의 `### ## ## ###`를 바꿉니다.

> **Note**
>
> 이 예에서는 `ArnLike` 및 `ArnEquals` 모두를 사용합니다. 기능적으로 동일하므로 정책을 구성할 때 둘 중 하나를 사용할 수 있습니다. Transfer Family 설명서에서는 조건에 와일드카드 문자가 포함된 경우에는 `ArnLike`를 사용하고, 정확한 일치 조건을 나타내기 위해서는 `ArnEquals`를 사용합니다.

AWS Transfer Family 사용자 역할 교차 서비스 혼동 방지

다음 예 정책은 계정에 있는 모든 서버의 모든 사용자가 역할을 수임하도록 허용합니다.

```
{
    "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/*"
                }
            }
        }
    ]
}
```

다음 예 정책은 특정 서버의 모든 사용자가 역할을 수임하도록 허용합니다.

```
{
    "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/*"
                }
            }
```

```
        }
    ]
}
```

다음 예 정책은 특정 서버의 특정 사용자가 역할을 수임하도록 허용합니다.

```
{
    "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 워크플로우 역할 서비스 간 혼란 방지

다음 예 정책은 계정의 모든 워크플로가 역할을 수임하도록 허용합니다.

```
{
    "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/*"
                }
            }
        }
    ]
}
```

다음 예 정책은 특정 워크플로가 역할을 가정하도록 허용합니다.

```
{
    "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 로깅 및 서비스 간 호출 역할 혼동 방지

> **Note**
>
> 다음 예는 로깅 및 호출 역할 모두에 사용할 수 있습니다.
> 이 예시에서는 서버에 연결된 워크플로가 없는 경우 워크플로의 ARN 세부 정보를 제거할 수 있습니다.

다음 예제 로깅/호출 정책은 계정의 모든 서버 (및 워크플로) 가 역할을 맡을 수 있도록 허용합니다.

```
{
    "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/*"
                   ]
                }
            }
        }
    ]
}
```

다음 예제 로깅/호출 정책은 특정 서버 (및 워크플로) 가 역할을 맡도록 허용합니다.

```
{
    "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"
                ]
            }
        }
    }
  ]
}
```

AWSAWS Transfer Family에 대한 관리형 정책

사용자, 그룹 및 역할에 권한을 추가하려면 정책을 직접 작성하는 것보다 AWS 관리형 정책을 사용하는 것이 더 쉽습니다. 팀에 필요한 권한만 제공하는 AWS Identity and Access Management (IAM) 고객 관리형 정책을 만들려면 시간과 전문 지식이 필요합니다. 빠르게 시작하려면 AWS 관리형 정책을 사용할 수 있습니다. 이 정책은 일반적인 사용 사례를 다루며 사용자의 AWS 계정에서 사용할 수 있습니다. AWS 관리형 정책에 대한 자세한 내용은 IAM 사용 설명서에서 AWS 관리형 정책을 참조하세요. 모든 AWS 관리형 정책의 자세한 목록은 AWS 관리형 정책 참조 가이드를 참조하십시오.

AWS 서비스는 AWS 관리형 정책을 유지 관리하고 업데이트합니다. AWS 관리형 정책에서는 권한을 변경할 수 없습니다. 서비스에서 때때로 추가 권한을 AWS 관리형 정책에 추가하여 새로운 기능을 지원합니다. 이 타입의 업데이트는 정책이 연결된 모든 보안 인증(사용자, 그룹 및 역할)에 적용됩니다. 서비스는 새로운 기능이 시작되거나 새 작업을 사용할 수 있을 때 AWS 관리형 정책에 업데이트됩니다. 서비스는 AWS 관리형 정책에서 권한을 제거하지 않으므로 정책 업데이트로 인해 기존 권한이 손상되지 않습니다.

또한 여러 서비스에 걸친 작업 기능에 대한 관리형 정책을 AWS 지원합니다. 예를 들어 `ReadOnlyAccess` AWS 관리형 정책은 모든 AWS 서비스와 리소스에 대한 읽기 전용 액세스를 제공합니다. 서비스가 새 기능을 시작하면 새 작업 및 리소스에 대한 읽기 전용 권한이 AWS 추가됩니다. 직무 정책의 목록과 설명은 IAM 사용 설명서의 직무에 관한AWS 관리형 정책을 참조하세요.

AWS 관리형 정책: AWSTransferConsoleFullAccess

이 `AWSTransferConsoleFullAccess` 정책은 AWS 관리 콘솔을 통해 Transfer Family에 대한 모든 권한을 제공합니다.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다:

- `acm:ListCertificates` – 인증서 Amazon 리소스 이름(ARN)의 목록과 각 ARN의 도메인 명칭을 검색할 수 있는 권한을 부여합니다.
- `ec2:DescribeAddresses` – 하나 이상의 탄력적 IP 주소를 설명할 수 있는 권한을 부여합니다.
- `ec2:DescribeAvailabilityZones` – 사용 가능한 하나 이상의 가용 영역을 설명할 수 있는 권한을 부여합니다.
- `ec2:DescribeNetworkInterfaces` – 하나 이상의 네트워크 인터페이스를 설명할 수 있는 권한을 부여합니다.
- `ec2:DescribeSecurityGroups` – 하나 이상의 보안 그룹을 설명할 수 있는 권한을 부여합니다.
- `ec2:DescribeSubnets` – 하나 이상의 서브넷을 설명할 수 있는 권한을 부여합니다.
- `ec2:DescribeVpcs` – 하나 이상의 가상 프라이빗 게이트웨이를 설명할 수 있는 권한을 부여합니다.
- `ec2:DescribeVpcEndpoints` – 하나 이상의 VPC 엔드포인트를 설명할 수 있는 권한을 부여합니다.
- `health:DescribeEventAggregates` – 각 이벤트 타입(문제, 예약된 변경 및 계정 알림)의 이벤트 수를 반환합니다.
- `iam:GetPolicyVersion` – 정책 설명서를 포함하여 지정된 관리형 정책의 버전에 대한 정보를 검색할 권한을 부여합니다.
- `iam:ListPolicies` – 모든 관리형 정책을 나열할 권한을 부여합니다.
- `iam:ListRoles` – 지정된 경로 접두사가 있는 IAM 역할을 나열할 권한을 부여합니다.
- `iam:PassRole` – Transfer Family에 IAM 역할을 전달할 수 있는 권한을 부여합니다. 자세한 내용은 사용자에게 역할을 전달할 수 있는 권한 부여를 AWS 서비스 참조하십시오.
- `route53:ListHostedZones` – 현재 AWS 계정와(과) 연계된 퍼블릭 및 프라이빗 호스팅 영역의 목록을 가져올 수 있는 권한을 부여합니다.
- `s3:ListAllMyBuckets` – 요청의 인증된 발신자가 소유한 모든 버킷을 나열할 수 있는 권한을 부여합니다.
- `transfer:*` – Transfer Family 리소스에 대한 액세스 권한을 부여합니다. 별표(*)는 모든 리소스에 대한 액세스 권한을 부여합니다.

```
{
    "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 관리형 정책: AWSTransferFullAccess

또한 `AWSTransferFullAccess` 정책은 Transfer Family 서비스에 대한 전체 액세스 권한을 제공합니다.

권한 세부 정보

이 정책에는 다음 권한이 포함되어 있습니다.

- `transfer:*` – Transfer Family 리소스에 대한 액세스 권한을 부여합니다. 별표(*)는 모든 리소스에 대한 액세스 권한을 부여합니다.

- `iam:PassRole` – Transfer Family에 IAM 역할을 전달할 수 있는 권한을 부여합니다. 자세한 내용은 사용자에게 역할을 전달할 수 있는 권한 부여를 AWS 서비스 참조하십시오.
- `ec2:DescribeAddresses` – 하나 이상의 탄력적 IP 주소를 설명할 수 있는 권한을 부여합니다.
- `ec2:DescribeNetworkInterfaces` – 하나 이상의 네트워크 인터페이스를 설명할 수 있는 권한을 부여합니다.
- `ec2:DescribeVpcEndpoints` – 하나 이상의 VPC 엔드포인트를 설명할 수 있는 권한을 부여합니다.

```
{
    "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 관리형 정책: AWSTransferLoggingAccess

이 AWSTransferLoggingAccess 정책은 AWS Transfer Family에 로그 스트림과 그룹을 생성하고 계정에 로그 이벤트를 추가할 수 있는 모든 권한을 부여합니다.

권한 세부 정보

이 정책에는 에 대한 다음과 같은 권한이 포함됩니다 Amazon CloudWatch Logs.

- CreateLogStream – 보안 주체에게 로그 스트림을 생성할 수 있는 권한을 부여합니다.
- DescribeLogStreams – 주도자에게 로그 그룹의 로그 스트림을 나열할 수 있는 권한을 부여합니다.
- CreateLogGroup – 보안 주체에게 로그 스트림을 생성할 수 있는 권한을 부여합니다.
- PutLogEvents – 보안 주체가 로그 이벤트의 배치를 로그 스트림에 업로드할 수 있는 권한을 부여합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:DescribeLogStreams",
                "logs:CreateLogGroup",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS 관리형 정책: AWSTransferReadOnlyAccess

AWSTransferReadOnlyAccess 정책은 Transfer Family 서비스에 대한 읽기 전용 액세스를 제공합니다.

권한 세부 정보

이 정책에는 Transfer Family에 대한 다음 권한이 포함되어 있습니다.

- `DescribeUser` – 주도자에게 사용자의 설명을 볼 수 있는 권한을 부여합니다.
- `DescribeServer` – 주도자에게 서버에 대한 설명을 볼 수 있는 권한을 부여합니다.
- `ListUsers` – 주도자에게 서버의 사용자를 나열할 수 있는 권한을 부여합니다.
- `ListServers` – 주도자에게 계정의 서버를 나열할 수 있는 권한을 부여합니다.
- `TestIdentityProvider` – 보안 주체에게 구성된 ID 제공자가 올바르게 설정되었는지 테스트할 수 있는 권한을 부여합니다.
- `ListTagsForResource` – 주도자에게 리소스에 대한 태그를 나열할 수 있는 권한을 부여합니다.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "transfer:DescribeUser",
                "transfer:DescribeServer",
                "transfer:ListUsers",
                "transfer:ListServers",
                "transfer:TestIdentityProvider",
                "transfer:ListTagsForResource"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS Family 업데이트를 AWS 관리형 정책으로 이전

Transfer Family에서 이러한 변경 사항을 추적하기 시작한 이후 업데이트된 AWS Transfer Family의 AWS 관리형 정책에 대한 세부 정보를 확인하세요. 이 페이지의 변경 사항에 대한 자동 알림을 받아보려면 에 대한 문서 기록 AWS Transfer Family 페이지에서 RSS 피드를 구독하십시오.

변경 사항	설명	날짜
설명서 업데이트	Transfer Family 관리 정책 각각에 대한 섹션이 추가되었습니다.	2022년 1월 27일
AWSTransferReadOnlyAccess-기존 정책 업데이트	AWS Transfer Family는 정책을 읽을 수 있는 새 권한을 추가했습니다 AWS Managed Microsoft AD.	2021년 9월 30일
AWS Transfer Family가 변경 사항을 추적하기 시작했습니다.	AWS Transfer Family는 AWS 관리형 정책의 변경 사항을 추적하기 시작했습니다.	2021년 6월 15일

문제 해결 AWS Transfer Family

다음 정보를 사용하면 작업할 때 발생할 수 있는 일반적인 문제를 진단하고 해결하는 데 도움이 AWS Transfer Family됩니다.

Transfer Family의 IAM 관련 문제는 AWS Transfer Family ID 및 액세스 문제 해결를 참조하세요.

주제

서비스 관리 사용자 문제 해결

이 섹션에서는 다음 문제에 대한 가능한 해결 방법을 설명합니다.

주제

Amazon EFS 서비스 관리 사용자 문제 해결

설명

`sftp` 명령을 실행하면 프롬프트가 나타나지 않고 대신 다음 메시지가 표시됩니다:

```
Couldn't canonicalize: Permission denied
           Need cwd
```

원인

AWS Identity and Access Management (IAM) 사용자의 역할에는 Amazon Elastic File System (Amazon EFS) 에 액세스할 수 있는 권한이 없습니다.

솔루션

사용자 역할에 대한 정책 권한을 늘리세요. 와 같은 AWS `AmazonElasticFileSystemClientFullAccess` 관리형 정책을 추가할 수 있습니다.

퍼블릭 키 본문이 너무 긴 문제 해결

설명

서비스 관리 사용자를 생성하려고 하면 다음 오류가 발생합니다:

```
Failed to create user (1 validation error detected:
'sshPublicKeyBody' failed to satisfy constraint: Member must have length less than or
 equal to 2048)
```

원인

공개 키 본문에는 PGP 키를 입력할 수 있지만 서비스 관리 사용자의 경우 PGP 키를 AWS Transfer Family 지원하지 않을 수 있습니다.

솔루션

PGP 키가 RSA 기반인 경우, 귀하는 그것을 PEM 형식으로 변환할 수 있습니다. 예를 들어, Ubuntu는 다음과 같은 변환 도구를 제공합니다: https://manpages.ubuntu.com/manpages/xenial/man1/openpgp2ssh.1.html

문제 해결에서 SSH 퍼블릭 키를 추가하지 못했습니다.

설명

서비스 관리 사용자에 대한 퍼블릭 키를 추가하려고 하면 다음 오류가 발생합니다:

```
Failed to add SSH public key (Unsupported or invalid SSH public key format)
```

원인

SSH2 형식의 공개 키를 가져오려고 할 수 있지만 서비스 관리 사용자를 위한 SSH2 형식의 공개 키는 지원되지 않습니다. AWS Transfer Family

솔루션

키를 OpenSSH 형식으로 변환해야 합니다. 이 과정은 SSH2 퍼블릭 키를 PEM 형식으로 변환에 설명되어 있습니다.

Amazon API Gateway 문제 해결

이 섹션에서는 다음과 같은 API Gateway 문제에 대한 가능한 해결 방법을 설명합니다.

주제

- 인증 오류가 너무 많음
- 연결 종료

인증 오류가 너무 많음

설명

SSH(Secure Shell) 파일 전송 프로토콜(SFTP)을 사용하여 서버에 연결하려고 하면 다음 오류가 발생합니다.

```
Received disconnect from 3.15.127.197 port 22:2: Too many authentication failures
Authentication failed.
Couldn't read packet: Connection reset by peer
```

원인

잘못된 데이터베이스 사용자 암호를 입력했을 수 있습니다. 올바른 암호를 다시 입력해 보세요.

암호가 올바르면 유효하지 않은 Amazon 리소스 이름(ARN) 역할이 문제의 원인일 수 있습니다. 이것이 문제인지 확인하려면 서버의 ID 공급자를 테스트하세요. 다음과 비슷한 응답이 표시되는 경우 역할 ARN은 자리 표시자일 뿐이며 역할 ID 값이 모두 0인 것으로 표시됩니다:

```
{
    "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"
}
```

솔루션

자리 표시자 역할 ARN을 서버 액세스 권한이 있는 실제 역할로 교체하세요.

역할을 업데이트하려면

1. https://console.aws.amazon.com/cloudformation 에서 AWS CloudFormation 콘솔을 엽니다.
2. 왼쪽 탐색 창에서 스택을 선택합니다.
3. 스택 목록에서 스택을 선택한 다음 파라미터 탭을 선택합니다.
4. 업데이트를 선택합니다. 업데이트 스택 페이지에서 현재 템플릿 사용을 선택하고 다음을 선택합니다.
5. Transfer Family 서버에 액세스할 수 있는 충분한 권한이 있는 역할 UserRoleArnARN으로 바꾸십시오.

   > **Note**
   >
   > 필요한 권한을 부여하려면 `AmazonAPIGatewayAdministrator` 및 `AmazonS3FullAccess` 관리형 정책을 역할에 추가할 수 있습니다.

6. 다음을 선택하고 다시 다음을 선택합니다. ## 검토 페이지에서 IAM 리소스를 생성할 AWS CloudFormation 수 있음을 인정함을 선택한 다음 Update stack (스택 업데이트) 을 선택합니다.

연결 종료

설명

SSH(Secure Shell) 파일 전송 프로토콜(SFTP)을 사용하여 서버에 연결하려고 하면 다음 오류가 발생합니다.

```
Connection closed
```

원인

이 문제의 한 가지 가능한 원인은 Amazon CloudWatch 로깅 역할이 Transfer Family와 신뢰 관계를 맺고 있지 않기 때문입니다.

솔루션

서버의 로깅 역할이 Transfer Family와 신뢰 관계를 맺고 있는지 확인하세요. 자세한 내용은 신뢰 관계를 구축하기 위해를 참조하세요.

암호화된 Amazon S3 버킷에 대한 정책 문제 해결

설명

Transfer Family 서버의 스토리지로 사용하고 있는 암호화된 Amazon S3 버킷이 있습니다. 서버에 파일을 업로드하려고 하면 오류 `Couldn't close file: Permission denied`가 발생합니다.

그리고 서버 로그를 보면 다음과 같은 오류가 표시됩니다:

```
ERROR Message="Access denied" Operation=CLOSE Path=/bucket/user/test.txt BytesIn=13
ERROR Message="Access denied"
```

원인

IAM 사용자에 대한 정책에는 암호화된 버킷에 액세스할 권한이 없습니다.

솔루션

요구되는 AWS Key Management Service (AWS KMS) 권한을 부여하려면 정책에 추가 권한을 지정해야 합니다. 자세한 내용은 Amazon S3의 데이터 암호화를 참조하세요.

인증 문제 해결

이 섹션에서는 다음 인증 문제에 대한 가능한 해결 방법을 설명합니다.

주제

인증 실패—SSH/SFTP

설명

SSH(Secure Shell) 파일 전송 프로토콜(SFTP)을 사용하여 서버에 연결하려고 하면 다음과 비슷한 메시지가 나타납니다:

```
Received disconnect from 3.130.115.105 port 22:2: Too many authentication failures
  Authentication failed.
```

> **Note**
>
> API Gateway를 사용 중이고 이 오류가 발생하는 경우 인증 오류가 너무 많음를 참조하세요.

원인

사용자에 대한 RSA 키 쌍을 추가하지 않았으므로 대신 암호를 사용하여 인증해야 합니다.

솔루션

`sftp` 명령은 `-o PubkeyAuthentication=no` 옵션을 지정할 때 실행됩니다. 이 옵션을 선택하면 시스템에서 강제로 암호를 요청합니다. 예:

```
sftp -o PubkeyAuthentication=no sftp-user@server-id.server.transfer.region-
id.amazonaws.com
```

관리형 AD 불일치 영역 문제

설명

사용자의 영역과 그룹 영역이 일치해야 합니다. 둘 다 기본 영역에 있거나 둘 다 신뢰할 수 있는 영역에 있어야 합니다.

원인

사용자와 그룹이 일치하지 않는 경우 Transfer Family에서 사용자를 인증할 수 없습니다. 사용자의 ID 공급자를 테스트하면 사용자 그룹에 연결된 액세스 권한을 찾을 수 없음이라는 오류 메시지가 표시됩니다.

솔루션

사용자 영역에서 그룹 영역(기본값 또는 신뢰할 수 있음)과 일치하는 그룹을 참조합니다.

기타 인증 문제

설명

인증 오류가 발생했지만 다른 문제 해결은 작동하지 않습니다.

원인

선행 또는 후행 슬래시(/) 가 포함된 논리 디렉터리의 대상을 지정했을 수 있습니다.

솔루션

논리적 디렉터리 대상을 업데이트하여 슬래시로 시작하고 후행 슬래시를 포함하지 않도록 하세요. 예를 들어 `/DOC-EXAMPLE-BUCKET/images` 는 허용되지만 `DOC-EXAMPLE-BUCKET/images` 허용되지는 않습니다. `/DOC-EXAMPLE-BUCKET/images/`

관리형 워크플로 문제 해결

이 섹션에서는 다음과 같은 워크플로우 문제에 대한 가능한 해결 방법을 설명합니다.

주제

- Amazon을 사용하여 워크플로 관련 오류 문제 해결 CloudWatch
- 워크플로 복사 오류 문제 해결

Amazon을 사용하여 워크플로 관련 오류 문제 해결 CloudWatch

설명

워크플로에 문제가 있는 경우 Amazon을 사용하여 원인을 CloudWatch 조사할 수 있습니다.

원인

원인은 여러 가지가 있을 수 있습니다. Amazon CloudWatch Logs를 사용하여 조사하십시오.

솔루션

Transfer Family는 워크플로우 실행 상태를 CloudWatch 로그로 내보냅니다. CloudWatch 로그에는 다음과 같은 유형의 워크플로 오류가 나타날 수 있습니다.

- `"type": "StepErrored"`
- `"type": "ExecutionErrored"`
- `"type": "ExecutionThrottled"`
- `"Service failure on starting workflow"`

다양한 필터 및 패턴 구문을 사용하여 워크플로의 실행 로그를 필터링할 수 있습니다. 예를 들어, 로그에 로그 필터를 생성하여 ExecutionErrored메시지가 포함된 워크플로 실행 로그를 캡처할 수 있습니다. CloudWatch 자세한 내용은 Amazon CloudWatch Logs 사용 설명서의 구독을 통한 로그 데이터의 실시간 처리 및 필터 및 패턴 구문을 참조하십시오.

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"}
```

여기서 `StepErrored`는 워크플로 내의 한 단계에서 오류가 발생했음을 나타냅니다. 단일 워크플로에서 여러 단계를 구성할 수 있습니다. 이 오류는 오류가 발생한 단계를 알려주고 오류 메시지를 제공합니다. 이 특정 예에서는 파일에 태그를 지정하도록 단계를 구성했지만 Amazon EFS 파일 시스템에서 파일에 태그를 지정하는 것은 지원되지 않으므로 단계에서 오류가 발생했습니다.

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"}}
```

워크플로에서 어떤 단계도 실행할 수 없는 경우 ExecutionErrored 메시지가 생성됩니다. 예를 들어, 특정 워크플로에서 단일 단계를 구성했는데 해당 단계를 실행할 수 없는 경우 전체 워크플로가 실패합니다.

Executionthrottled

시스템에서 지원할 수 있는 속도보다 빠른 속도로 워크플로가 트리거되면 실행이 제한됩니다. 이 로그 메시지는 워크플로의 실행 속도를 늦춰야 함을 나타냅니다. 워크플로 실행 속도를 줄일 수 없는 경우 Contact로 문의하십시오. AWS SupportAWS

워크플로 시작 시 서비스 실패

서버에서 워크플로를 제거하고 새 워크플로로 바꾸거나 워크플로의 실행 역할에 영향을 미치는 서버 구성을 업데이트할 때마다 새 워크플로를 실행하기 전에 약 10분을 기다려야 합니다. Transfer Family 서버는 워크플로 세부 정보를 캐시하며 서버가 캐시를 새로 고치는 데 10분이 걸립니다.

또한 활성 SFTP 세션에서 로그아웃한 다음 10분 대기 시간이 지난 후 다시 로그인해야 변경 사항을 확인할 수 있습니다.

워크플로 복사 오류 문제 해결

설명

업로드된 파일을 복사하는 단계가 포함된 워크플로를 실행하는 경우 다음 오류가 발생할 수 있습니다:

```
{
  "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"
  }
}
```

원인

원본 파일은 대상 버킷과 AWS 리전 다른 Amazon S3 버킷에 있습니다.

솔루션

복사 단계가 포함된 워크플로를 실행하는 경우, 소스 버킷과 목적지 버킷이 동일한 AWS 리전안에 있는지 확인하세요.

워크플로 복호화 문제 해결

이 섹션에서는 암호화된 워크플로와 관련된 다음과 같은 문제에 대한 가능한 해결책을 설명합니다.

주제

- 서명된 암호화 파일의 오류 문제 해결
- FIPS 알고리즘의 오류 문제 해결

서명된 암호화 파일의 오류 문제 해결

설명

암호 해독 워크플로가 실패하고 다음과 같은 오류 메시지가 나타납니다.

```
"Encrypted file with signed message unsupported"
```

원인

Transfer Family는 현재 암호화된 파일에 대한 서명을 지원하지 않습니다.

솔루션

PGP 클라이언트에서 암호화된 파일에 서명하는 옵션이 있는 경우 선택을 취소해야 합니다. Transfer Family는 현재 암호화된 파일에 대한 서명을 지원하지 않기 때문입니다.

FIPS 알고리즘의 오류 문제 해결

설명

복호화 워크플로가 실패하고 로그 메시지가 다음과 비슷합니다:

```
{
   "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"
   }
}
```

원인

Transfer Family 서버에는 FIPS 모드가 활성화되어 있으며 관련 복호화 워크플로 단계가 있습니다. Transfer Family 서버에 업로드하기 전에 파일을 암호화하는 경우 암호화 클라이언트는 FIPS에서 승인하지 않은 대칭 암호화 알고리즘을 사용하는 암호화된 파일을 생성할 수 있습니다. 이러한 시나리오에서는 워크플로에서 파일을 해독할 수 없습니다. 다음 예에서 GnuPG 버전 2.4.0은 OCB (비 FIPS 블록 암호 모드) 를 사용하여 파일을 암호화합니다: 이로 인해 워크플로가 실패합니다.

솔루션

파일을 암호화하는 데 사용한 GPG 키를 편집한 다음 다시 암호화해야 합니다. 다음 절차는 취해야 하는 단계를 설명합니다.

PGP 키를 편집하려면

1. `gpg --list-keys`를 실행하여 편집해야 하는 키를 식별합니다.

   그러면 키 목록이 반환됩니다. 각 키에는 다음과 유사한 세부 정보가 있습니다.

   ```
   pub   ed25519 2022-07-07 [SC]
         wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
   uid           [ultimate] Mary Major <marymajor@example.com>
   sub   cv25519 2022-07-07 [E]
   ```

2. 편집할 키를 식별합니다. 이전 단계에 표시된 예에서 ID는 `wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`입니다.
3. `gpg --edit-key wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`를 실행합니다.

시스템은 GnuPG 프로그램 및 지정된 키에 대한 세부 정보로 응답합니다.

4. gpg> 프롬프트에서 showpref를 입력합니다. 다음과 같은 세부 정보가 반환됩니다:

```
[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
```

   참고로, 키에 저장된 선호 알고리즘이 나열되어 있습니다.

5. OCB를 제외한 모든 알고리즘을 유지하도록 키를 편집하려고 합니다. setpref 명령을 실행하여 유지할 모든 알고리즘을 지정합니다:

```
gpg> setpref AES256, AES192,AES,3DES,SHA512, SHA384, SHA256, SHA224, SHA1,ZLIB,
 BZIP2, ZIP, Uncompressed
```

   그러면 다음 세부 사항이 반환됩니다:

```
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. y를 입력하여 업데이트한 다음, 변경을 확인하라는 메시지가 표시되면 암호를 입력합니다.
7. 변경 사항을 저장합니다.

```
gpg> save
```

복호화 워크플로를 다시 실행하기 전에 편집한 키를 사용하여 파일을 다시 암호화해야 합니다.

Amazon SES 문제 해결

이 섹션에서는 다음과 같은 Amazon EFS 문제에 대한 가능한 해결 방법을 설명합니다.

주제

누락된 POSIX 프로필 문제 해결

설명

서버에 Amazon EFS 스토리지를 사용하고 사용자 지정 ID 공급자를 사용하는 경우 AWS Lambda 함수에 POSIX 프로필을 제공해야 합니다.

원인

한 가지 가능한 원인은 AWS Lambda지원 Amazon API Gateway 메서드를 생성하기 위해 제공하는 템플릿에 현재 POSIX 정보가 포함되어 있지 않기 때문입니다.

POSIX 정보를 제공한 경우 POSIX 정보를 제공하는 데 사용한 형식이 Transfer Family에서 올바르게 파싱되지 않을 수 있습니다.

솔루션

Transfer Family에 `PosixProfile` 파라미터에 대한 JSON 요소를 제공하고 있는지 확인하세요.

예를 들어, Python을 사용하는 경우 `PosixProfile` 파라미터를 파싱하는 위치에 다음 줄을 추가할 수 있습니다.

```
if PosixProfile:
        response_data["PosixProfile"] = json.loads(PosixProfile)
```

또는 에서 JavaScript 다음 줄을 추가할 수 있습니다. 여기서 *`uid-value`* 및 *`gid-value`* 는 각각 사용자 ID (UID) 와 그룹 ID (GID) 를 나타내는 0 이상의 정수입니다.

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

이 코드 예는 `PosixProfile` 파라미터를 문자열이 아닌 JSON 객체로 Transfer Family에 전송합니다.

또한 다음과 AWS Secrets Manager같이 내에 `PosixProfile` 매개변수를 저장해야 합니다. *`your-uid`*와(과) *`your-gid`*를 GID 및 UID의 실제 값으로 바꾸세요.

```
{"Uid": your-uid, "Gid": your-gid, "SecondaryGids": []}
```

Amazon EFS로 논리적 디렉터리 문제 해결

설명

사용자의 홈 디렉터리가 존재하지 않고 사용자가 `ls` 명령을 실행하면 시스템은 다음과 같이 응답합니다:

```
sftp> ls
remote readdir ("/"): No such file or directory
```

원인

Transfer Family 서버에서 Amazon EFS를 사용하는 경우, 사용자가 논리적 홈 디렉터리에서 작업할 수 있으려면 먼저 읽기 및 쓰기 권한이 있는 사용자의 홈 디렉터리를 생성해야 합니다. 사용자는 논리적 홈 디렉터리 상의 `mkdir`에 대한 권한이 없기 때문에 이 디렉터리를 직접 만들 수는 없습니다.

솔루션

상위 디렉터리에 대한 관리 액세스 권한이 있는 사용자는 사용자의 논리적 홈 디렉터리를 생성해야 합니다.

ID 제공자 테스트 문제 해결

설명

콘솔 또는 `TestIdentityProvider` API 호출을 사용하여 ID 공급자를 테스트하는 경우 `Response` 필드는 비어 있습니다. 예:

```
{
    "Response": "{}",
    "StatusCode": 200,
    "Message": ""
}
```

원인

가장 가능성이 높은 원인은 잘못된 사용자 이름이나 암호로 인해 인증이 실패했기 때문입니다.

솔루션

사용자에 대한 올바른 자격 증명을 사용하고 있는지 확인하고 필요한 경우 사용자 이름 또는 암호를 업데이트하세요.

SFTP 커넥터의 신뢰할 수 있는 호스트 키 추가 문제 해결

설명

SFTP 커넥터를 만들거나 편집하고 신뢰할 수 있는 호스트 키를 추가하면 다음 오류 메시지가 나타납니다: `Failed to edit connector details (Invalid host key format.)`

원인

올바른 퍼블릭 키를 붙여넣으면 키의 `comment` 일부가 포함된 것이 문제일 수 있습니다. AWS Transfer Family 현재 키의 설명 부분을 수락하지 않습니다.

솔루션

키를 텍스트 필드에 붙여넣을 때 키의 설명 부분을 삭제하세요. 예를 들어, 키는 다음과 비슷할 것입니다:

```
ssh-rsa AAAA...== marymajor@dev-dsk-marymajor-1d-c1234567.us-east-1.amazon.com
```

== 문자 뒤에 오는 텍스트를 제거하고, 키에서 == 문자 앞까지 오는 부분만 붙여넣습니다.

```
ssh-rsa AAAA...==
```

파일 업로드 문제 해결

이 섹션에서는 다음과 같은 파일 업로드 문제에 대한 가능한 해결 방법을 설명합니다.

주제

- Amazon S3 파일 업로드 오류 문제 해결
- 읽을 수 없는 파일 명칭 문제 해결

Amazon S3 파일 업로드 오류 문제 해결

설명

Transfer Family를 사용하여 Amazon S3 스토리지에 파일을 업로드하려고 하면 다음과 같은 오류 메시지가 나타납니다: AWS Transfer는 S3 객체에 대한 임의 액세스 쓰기를 지원하지 않음.

원인

Amazon S3를 서버 스토리지로 사용하는 경우 Transfer Family는 단일 전송에 대한 복수 연결을 지원하지 않습니다.

솔루션

Transfer Family 서버가 Amazon S3를 스토리지로 사용하는 경우 단일 전송에 복수 연결을 사용한다고 언급하는 클라이언트 소프트웨어 옵션을 비활성화하세요.

읽을 수 없는 파일 명칭 문제 해결

설명

업로드한 파일 중 일부에 손상된 파일 명칭이 있습니다. FTP 및 SFTP 전송 시 움라우트, 악센트 부호가 있는 글자 또는 중국어나 아랍어와 같은 특정 스크립트 등 파일 명칭의 특정 문자가 제대로 표시되지 않는 문제가 발생하는 경우가 있습니다.

원인

FTP 및 SFTP 프로토콜을 사용하면 클라이언트가 파일 명칭의 문자 인코딩을 협상할 수 있지만 Amazon S3와 Amazon EFS는 그렇지 않습니다. 대신 UTF-8 문자 인코딩이 필요합니다. 따라서 특정 문자가 제대로 렌더링되지 않습니다.

솔루션

이 문제를 해결하려면 클라이언트 응용 프로그램에서 파일 명칭 문자 인코딩을 검토하고 해당 인코딩이 UTF-8 로 설정되어 있는지 확인하세요.

`ResourceNotFound` 예외 문제 해결

설명

리소스를 찾을 수 없는 오류가 발생합니다. 예를 들어 `UpdateServer`를 실행하면 다음과 같은 오류가 발생할 수 있습니다:

```
An error occurred (ResourceNotFoundException) when calling the UpdateServer operation:
 Unknown server
```

원인

ResourceNotFoundException메시지를 받는 데에는 여러 가지 이유가 있습니다. 대부분의 경우 API 명령에 지정한 리소스가 존재하지 않습니다. 기존 리소스를 지정한 경우 가장 가능성이 높은 원인은 기본 지역이 리소스의 지역과 다르기 때문입니다. 예를 들어 기본 지역이 us-east-1이고 Transfer Family 서버가 us-east-2에 있는 경우 알 수 없는 리소스 예외가 발생합니다.

기본 지역 설정에 대한 자세한 설명은 aws configure를 통한 빠른 구성을 참조하세요.

솔루션

API 명령에 지역 파라미터를 추가하여 특정 리소스를 찾을 위치를 명시적으로 지정합니다.

```
aws transfer -describe-server --server-id server-id --region us-east-2
```

SFTP 커넥터 문제 해결

이 섹션에서는 다음 SFTP 커넥터 문제의 가능한 해결 방법을 설명합니다.

주제

- 키 협상이 실패했습니다.
- 기타 SFTP 커넥터 문제

키 협상이 실패했습니다.

설명

키 교환 협상이 실패하는 오류가 발생합니다. 예:

```
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]
```

원인

이 오류는 서버에서 지원하는 호스트 키 알고리즘과 커넥터에서 지원하는 호스트 키 알고리즘이 겹치지 않기 때문입니다.

솔루션

원격 서버가 오류 메시지에 나열된 클라이언트 호스트 키 알고리즘 중 하나 이상을 지원하는지 확인하세요. 지원되는 키 알고리즘 목록은 SFTP AWS Transfer Family 커넥터에 대한 보안 정책 섹션을 참조하세요.

기타 SFTP 커넥터 문제

설명

실행 후 오류가 `StartFileTransfer` 발생하지만 문제의 원인을 알 수 없으며 API 호출 후 커넥터 ID만 반환됩니다.

원인

이 오류에는 여러 가지 원인이 있을 수 있습니다. 문제를 해결하려면 커넥터를 테스트하고 CloudWatch 로그를 검색하는 것이 좋습니다.

솔루션

- 커넥터 테스트: 을 참조하십시오SFTP 커넥터를 테스트합니다.. 테스트가 실패하면 시스템은 테스트 실패 이유를 기반으로 오류 메시지를 제공합니다. 이 섹션에서는 콘솔에서 또는 TestConnectionAPI 명령을 사용하여 커넥터를 테스트하는 방법을 설명합니다.
- 커넥터의 CloudWatch 로그 보기: 을 참조하십시오SFTP 커넥터의 예제 로그 항목. 이 항목에서는 SFTP 커넥터 로그 항목의 예와 적절한 로그를 찾는 데 도움이 되는 명명 규칙을 제공합니다.

AS2 문제 해결

적용성 보고서 2(AS2) 지원 서버의 오류 메시지 및 문제 해결 팁은 다음과 같습니다: AS2 오류 코드.

API 참조

다음 섹션에서는 AWS Transfer Family API 서비스 호출, 데이터 유형, 매개변수 및 오류를 문서화합니다.

주제

AWS Transfer Family API에 오신 것을 환영합니다.

AWS Transfer Family 다음 프로토콜을 통해 Amazon Simple Storage Service (Amazon S3) 스토리지로 파일을 주고받는 데 사용할 수 있는 안전한 전송 서비스입니다.

- Secure Shell (SSH) File Transfer 프로토콜 (SFTP)
- File Transfer 프로토콜 보안(FTPS)
- File Transfer 프로토콜(FTP)
- 적용성 보고서 2(AS2)

파일 전송 프로토콜은 금융 서비스, 의료, 광고, 소매 등 다양한 업계의 데이터 교환 워크플로에 사용됩니다. AWS Transfer Family 파일 전송 워크플로를 로 마이그레이션하는 AWS작업을 간소화합니다.

AWS Transfer Family 서비스를 사용하려면 선택한 AWS 지역의 서버를 인스턴스화하면 됩니다. 서버를 생성하고, 가용 서버를 나열하고, 서버를 업데이트하고 삭제할 수 있습니다. 서버는 파일 작업을 요청하는 엔티티입니다. AWS Transfer Family서버에는 수많은 중요 속성이 존재합니다. 서버는 시스템이 할당한 `ServerId` 식별자로 확인하는 이름이 지정된 인스턴스입니다. 옵션으로, 서버에 호스트 이름을 할당할 수 있습니다. 사용자 지정 호스트 이름도 가능합니다. 이 서비스는 인스턴스화한 서버(심지어 `OFFLINE` 서버 포함)와 전송한 데이터 양에 대한 요금을 청구합니다.

사용자는 파일 작업을 요청하는 서버가 알고 있어야 합니다. 사용자 이름으로 식별한 사용자가 서버에 할당됩니다. 사용자 이름은 요청을 인증할 때 사용합니다. 서버는 인증 메서드로

AWS_DIRECTORY_SERVICE, SERVICE_MANAGED, AWS_LAMBDA, 또는 API_GATEWAY 중 오직 하나만 사용할 수 있습니다.

다음과 같은 ID 제공자 타입을 사용하여 사용자를 인증할 수 있습니다.

- SERVICE_MANAGED의 경우 SSH 퍼블릭 키는 서버의 사용자 속성에 저장됩니다. 사용자는 SERVICE_MANAGED 인증 메서드에 대해 하나 이상의 SSH 퍼블릭 키를 보유할 수 있습니다. 클라이언트가 SERVICE_MANAGED 메서드에 대한 파일 작업을 요청하면, 클라이언트는 사용자 이름과 SSH 프라이빗 키를 제공하며, 키가 인증되면 액세스할 수 있게 됩니다.
- AWS_DIRECTORY_SERVICE 인증 방법을 선택하여 Microsoft Active Directory 그룹을 사용하여 사용자 인증 및 액세스를 관리할 수 있습니다.
- 를 사용하여 사용자 지정 ID 공급자에 연결할 수 AWS Lambda있습니다. AWS_LAMBDA 인증 방법을 선택합니다.
- 사용자 인증과 액세스를 모두 제공하는 사용자 지정 인증 메서드를 이용해 사용자 요청을 인증할 수도 있습니다. 이 메서드는 Amazon API Gateway에 의존하여, 자격 증명 공급자가 제공한 API 직접 호출을 이용해 사용자 요청을 확인합니다. 이 메서드는 API 호출에서는 API_GATEWAY라고 하며, 콘솔에서는 Custom(사용자 지정) 이라고 합니다. 이 사용자 지정 메서드를 이용해 디렉터리 서비스, 데이터베이스 이름과 암호 쌍 또는 기타 메커니즘에 대해 사용자를 인증할 수도 있습니다.

사용자에는 사용자 자신과 Amazon S3 버킷 간의 신뢰 관계를 제공하는 정책이 할당됩니다. 사용자는 버킷 전체 또는 일부에 액세스할 수 있습니다. 서버가 사용자를 대신해 작동하려면, 서버는 사용자로부터 신뢰 관계를 상속받아야 합니다. 신뢰 관계를 포함하는 AWS Identity and Access Management (IAM) 역할이 생성되며, 해당 역할에는 AssumeRole 작업이 할당됩니다. 그런다음 서버는 마치 사용자인 것처럼 파일 작업을 수행할 수 있습니다.

home 디렉터리 속성을 설정한 사용자는 해당 디렉터리(또는 폴더)가 파일 작업의 대상 및 원본으로 작동하게 해야 합니다. home 디렉터리가 설정되지 않았다면, 버킷의 root 디렉터리가 랜딩 디렉터리가 됩니다.

서버, 사용자, 역할은 모두 Amazon 리소스 이름(ARN)으로 식별합니다. ARN을 사용하여 개체에 태그, 키 값 쌍을 할당할 수 있습니다. 태그는 이러한 개체(entity)를 그룹화하고 검색하는 데 사용할 수 있는 메타데이터입니다. 태그를 유용하게 사용할 수 있는 대표적인 분야는 회계입니다.

AWS Transfer Family ID 형식에서는 다음과 같은 규칙을 준수합니다.

- s-01234567890abcdef 양식에서 얻은 ServerId 값
- key-01234567890abcdef 양식에서 얻은 SshPublicKeyId 값

Amazon 리소스 이름(ARN) 양식은 다음과 같은 형식을 취합니다.

- 사용자의 경우, ARN은 `arn:aws:transfer:region:account-id:server/server-id` 형식을 취합니다.

  서버 ARN의 예: `arn:aws:transfer:us-east-1:123456789012:server/s-01234567890abcdef`
- 사용자의 경우, ARN은 `arn:aws:transfer:region:account-id:user/server-id/username` 형식을 취합니다.

  예를 들면, `arn:aws:transfer:us-east-1:123456789012:user/s-01234567890abcdef/user1`입니다.

사용 중인 DNS 항목(엔드포인트)은 다음과 같습니다.

- `transfer.region.amazonaws.com` 형식을 취하는 API 엔드포인트
- `server.transfer.region.amazonaws.com` 형식을 취하는 서버 엔드포인트

AWS 지역별 Transfer Family 엔드포인트 목록은 의 AWS Transfer Family 엔드포인트 및 할당량을 참조하십시오. AWS 일반 참조

이 API 인터페이스 참조에는 관리에 사용할 수 있는 프로그래밍 인터페이스에 대한 설명서가 AWS Transfer Family 포함되어 있습니다. AWS Transfer Family참조 구조는 다음과 같습니다.

- 작업의 영문자순 목록은 Actions를 참조하세요.
- 데이터 형식의 알파벳순 목록은 Data Types을 참조하세요.
- 공통 쿼리 파라미터 목록은 공통 파라미터를 참조하세요.
- 오류 코드에 대한 설명은 공통 오류를 참조하세요.

> **Tip**
>
> 명령을 실제로 실행하는 대신 모든 API 직접 호출에 `--generate-cli-skeleton` 파라미터를 사용하여 파라미터 템플릿을 생성하고 표시할 수 있습니다. 그런 다음 생성된 템플릿을 사용하여 사용자 정의하고 이후 명령의 입력으로 사용할 수 있습니다. 자세한 내용은 파라미터 스켈레톤 파일 생성 및 사용을 참조하세요.

작업

다음 작업이 지원됩니다.

- 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
- StartDirectoryListing
- StartFileTransfer
- StartServer
- StopServer
- TagResource
- TestConnection
- TestIdentityProvider
- UntagResource
- UpdateAccess
- UpdateAgreement
- UpdateCertificate
- UpdateConnector
- UpdateHostKey

- UpdateProfile
- UpdateServer
- UpdateUser

CreateAccess

관리자가 사용하는 활성화된 프로토콜을 통해 파일을 업로드하고 다운로드할 수 있는 액세스 권한을 가질 디렉터리의 그룹을 선택하는 데 사용합니다 AWS Transfer Family. 예를 들어, Microsoft Active Directory에는 50,000명의 사용자가 있을 수 있지만 서버에 파일을 전송할 수 있는 기능은 극히 일부만 필요할 수 있습니다. 관리자는 `CreateAccess`를 필요로 하는 올바른 사용자 집합으로 액세스를 제한하는 데 사용할 수 있습니다.

구문 요청

```
{
   "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"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

ExternalId

디렉터리 내의 특정 그룹을 식별하는 데 필요한 고유 식별자입니다. 연결하는 그룹의 사용자는 사용하는 활성화된 프로토콜을 통해 Amazon S3 또는 Amazon EFS 리소스에 액세스할 수 AWS

Transfer Family있습니다. 그룹 이름을 아는 경우 PowerShell Windows를 사용하여 다음 명령을 실행하여 SID 값을 볼 수 있습니다.

```
Get-ADGroup -Filter {samAccountName -like "YourGroupName*"} -Properties
* | Select SamAccountName,ObjectSid
```

이 명령에서 Active Directory 그룹의 YourGroupName이름으로 바꾸십시오.

이 파라미터를 확인하는 데 사용되는 정규 표현식은 공백 없이 대문자 및 소문자 영숫자로 구성된 문자의 문자열입니다. 밑줄이나 또한 =,.@:/- 문자도 포함할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: `S-1-[\d-]+`

필수 여부: 예

HomeDirectory

사용자가 클라이언트를 사용하여 서버에 로그인하는 경우 사용자를 위한 랜딩 디렉터리(폴더).

`HomeDirectory`의 예: `/bucket_name/home/mydirectory`

> **Note**
>
> `HomeDirectory` 파라미터는 `HomeDirectoryType`이(가) `PATH`(으)로 설정된 경우에만 사용합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 1024입니다.

패턴: `(|/.*)`

필수 여부: 아니요

HomeDirectoryMappings

사용자에게 표시할 Amazon S3 또는 Amazon EFS 경로 및 키와 이러한 경로 및 키를 표시할 방법을 지정하는 논리적 디렉터리 매핑입니다. `Entry` 및 `Target` 쌍을 지정해야 합니다. 여기서

Entry는 경로가 표시되는 방식을 보여주고 Target는 실제 Amazon S3 경로입니다. 대상만 지정하는 경우, 그대로 표시됩니다. 또한 AWS Identity and Access Management (IAM) 역할이 경로에 대한 액세스를 제공하는지 확인해야 합니다. Target 이 값은 LOGICAL로 설정된 경우에만 HomeDirectoryType를 설정할 수 있습니다.

다음은 Entry 및 Target 쌍의 예입니다.

```
[ { "Entry": "/directory1", "Target": "/bucket_name/home/
mydirectory" } ]
```

대부분의 경우 세션 정책 대신 이 값을 사용하여 사용자를 지정된 홈 디렉터리("chroot")로 제한할 수 있습니다. 이렇게 하려면 Entry을 /로 설정하고, Target을 HomeDirectory 파라미터 값으로 설정하면 됩니다.

다음은 chroot에 대한 Entry 및 Target 쌍의 예입니다.

```
[ { "Entry": "/", "Target": "/bucket_name/home/mydirectory" } ]
```

타입: HomeDirectoryMapEntry 객체 배열

어레이 멤버: 최소 항목 수 1개. 최대 항목 수는 50000개입니다.

필수 여부: 아니요

HomeDirectoryType

사용자가 서버에 로그인하는 경우 홈 디렉터리가 될 랜딩 디렉터리(폴더) 타입입니다. PATH로 설정하면, 사용자는 파일 전송 프로토콜 클라이언트에서와 같이 절대 Amazon S3 버킷 또는 EFS 경로를 볼 수 있습니다. LOGICAL로 설정하면, HomeDirectoryMappings에서 Amazon S3 또는 Amazon EFS 경로를 사용자에게 표시할 방법에 대한 매핑을 제공해야 합니다.

> **Note**
>
> HomeDirectoryType이 LOGICAL인 경우, HomeDirectoryMappings 파라미터를 사용하여 매핑을 제공해야 합니다. 반면에 HomeDirectoryType이 PATH인 경우, HomeDirectory 파라미터를 사용하여 절대 경로를 제공하세요. 템플릿에 HomeDirectory 및 HomeDirectoryMappings를 모두 포함할 수는 없습니다.

타입: 문자열

유효 값: `PATH | LOGICAL`

필수 여부: 아니요

Policy

여러 사용자가 동일한 AWS Identity and Access Management (IAM) 역할을 사용할 수 있도록 하기 위한 사용자 세션 정책. 이 정책은 Amazon S3 버킷의 부분에 대한 사용자의 액세스 범위를 축소합니다. 이 정책 내에서 사용할 수 있는 변수는 `${Transfer:UserName}`, `${Transfer:HomeDirectory}` 및 `${Transfer:HomeBucket}`입니다.

> **Note**
>
> 이 정책은 `ServerId`의 도메인이 Amazon S3인 경우에만 적용됩니다. Amazon EFS는 세션 정책을 사용하지 않습니다.
> 세션 정책의 경우 정책을 정책의 Amazon 리소스 이름 (ARN) 대신 JSON Blob으로 AWS Transfer Family 저장합니다. 정책을 JSON Blob으로 저장하고 `Policy` 인수로 전달합니다.
> 세션 정책의 예는 세션 정책 예를 참조하세요.
> 자세한 내용은 API AssumeRole참조를 AWS Security Token Service 참조하십시오.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 2048입니다.

필수 여부: 아니요

PosixProfile

Amazon EFS 파일 시스템에 대한 사용자의 액세스를 통제하는 사용자 ID(`Uid`), 그룹 ID(`Gid`) 및 보조 그룹 ID(`SecondaryGids`)를 포함한 전체 POSIX 자격 증명입니다. 파일 시스템의 파일 및 디렉터리에 설정된 POSIX 권한에 따라 Amazon EFS 파일 시스템에서 파일을 송수신할 때 사용자에게 제공되는 액세스 수준이 결정됩니다.

타입: PosixProfile 객체

필수 항목 여부: 아니요

Role

Amazon S3 버킷 또는 Amazon EFS 파일 시스템에 대한 사용자의 액세스를 제어하는 AWS Identity and Access Management (IAM) 역할의 Amazon 리소스 이름 (ARN). 이 역할에 연결된 정

책은 Amazon S3 버킷 또는 Amazon EFS 파일 시스템에 대한 파일 송수신 시 사용자에게 제공할 액세스의 수준을 결정합니다. 또한 IAM 역할에는 사용자의 전송 요청을 처리할 때 서버가 해당 리소스에 액세스할 수 있도록 허용하는 신뢰 관계가 포함되어야 합니다.

타입: 문자열

길이 제약: 최소 길이는 20. 최대 길이는 2,048.

패턴: `arn:.*role/\S+`

필수 여부: 예

ServerId

서버 인스턴스에 대해 시스템에서 할당한 고유 식별자입니다. 이 항목은 사용자를 추가한 특정 서버입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

필수 항목 여부: 예

응답 구문

```
{
   "ExternalId": "string",
   "ServerId": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

ExternalId

를 사용하여 활성화된 프로토콜을 통해 Amazon S3 또는 Amazon EFS 리소스에 액세스할 수 있는 그룹의 외부 식별자입니다 AWS Transfer Family.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: `S-1-[\d-]+`

ServerId

사용자가 연결된 서버의 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceExistsException

요청된 리소스가 존재하지 않거나 명령에 지정된 리전이 아닌 다른 리전에 있습니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS 파이썬용 SDK
- AWS 루비 V3용 SDK

CreateAgreement

계약을 생성합니다. 계약은 AWS Transfer Family 서버와 AS2 프로세스 간의 양자 거래 파트너 계약 또는 파트너십입니다. 계약은 서버와 AS2 프로세스 간의 파일 및 메시지 전송 관계를 정의합니다. 계약을 정의하기 위해 Transfer Family는 서버, 로컬 프로필, 파트너 프로필, 인증서 및 기타 속성을 결합합니다.

파트너는 `PartnerProfileId`로 식별되고 AS2 프로세스는 `LocalProfileId`로 식별됩니다.

구문 요청

```
{
   "AccessRole": "string",
   "BaseDirectory": "string",
   "Description": "string",
   "LocalProfileId": "string",
   "PartnerProfileId": "string",
   "ServerId": "string",
   "Status": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

AccessRole

커넥터는 AS2 또는 SFTP 프로토콜을 사용하여 파일을 전송하는 데 사용됩니다. 액세스 역할에는 사용할 AWS Identity and Access Management 역할의 Amazon 리소스 이름 (ARN) 을 제공합니다.

AS2 커넥터의 경우

AS2를 사용하여 `StartFileTransfer`를 호출하고 요청 파라미터인 `SendFilePaths`에 파일 경로를 지정하여 파일을 전송할 수 있습니다. 파일의 상위 디렉터리(예: `--send-file-paths /`

bucket/dir/file.txt의 경우 상위 디렉터리는 /bucket/dir/임)를 사용하여 처리된 AS2 메시지 파일을 임시로 저장하고, 파트너로부터 수신 시 MDN을 저장하고, 전송의 관련 메타데이터를 포함하는 최종 JSON 파일을 작성합니다. 따라서 AccessRole은(는) StartFileTransfer 요청에 사용된 파일 위치의 상위 디렉터리에 대한 읽기 및 쓰기 액세스를 제공해야 합니다. 또한 StartFileTransfer와(과) 함께 전송하려는 파일의 상위 디렉터리에 대한 읽기 및 쓰기 액세스 권한을 제공해야 합니다.

AS2 커넥터에 기본 인증을 사용하는 경우 액세스 역할에는 암호에 대한 secretsmanager:GetSecretValue 권한이 필요합니다. Secrets Manager에서 관리 키 대신 고객 AWS 관리 키를 사용하여 암호를 암호화하는 경우 역할에도 해당 키에 대한 kms:Decrypt 권한이 필요합니다.

SFTP 커넥터의 경우

액세스 역할이 StartFileTransfer 요청에서 사용된 파일 위치의 상위 디렉터리에 대한 읽기 및 쓰기 액세스 권한을 제공하는지 확인하세요. 또한 역할이 secretsmanager:GetSecretValue 권한을 제공하는지 확인하십시오. AWS Secrets Manager

타입: 문자열

길이 제약: 최소 길이는 20. 최대 길이는 2,048.

패턴: arn:.*role/\S+

필수 여부: 예

BaseDirectory

AS2 프로토콜을 사용하여 전송되는 파일의 랜딩 디렉터리(폴더)입니다.

BaseDirectory의 예: /DOC-EXAMPLE-BUCKET/home/mydirectory

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 1024입니다.

패턴: (|/.*)

필수 여부: 예

Description

계약을 식별하는 데 사용되는 명칭 또는 간단한 설명입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 200입니다.

패턴: `[\p{Graph}]+`

필수 여부: 아니요

LocalProfileId

AS2 로컬 프로필의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `p-([0-9a-f]{17})`

필수 여부: 예

PartnerProfileId

계약에 사용되는 파트너 프로필의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `p-([0-9a-f]{17})`

필수 여부: 예

ServerId

서버 인스턴스에 대해 시스템에서 할당한 고유 식별자입니다. 이는 계약에서 사용하는 특정 서버를 나타냅니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

필수 여부: 예

Status

구성 요소의 상태입니다. 계약은 `ACTIVE` 또는 `INACTIVE` 둘 중 하나일 수 있습니다.

타입: 문자열

유효 값: `ACTIVE | INACTIVE`

필수 여부: 아니요

Tags

계약을 그룹화하고 검색하는 데 사용할 수 있는 키-값 쌍입니다.

타입: Tag 객체 배열

배열 멤버: 최소 항목 수는 1개입니다. 최대 항목 수는 50개입니다.

필수 여부: 아니요

응답 구문

```
{
   "AgreementId": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

AgreementId

계약의 고유 식별자입니다. 이 ID를 사용하여 계약을 삭제하거나 업데이트할 수 있을 뿐만 아니라 계약 ID를 지정해야 하는 기타 API 호출에도 사용할 수 있습니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `a-([0-9a-f]{17})`

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

AWS Transfer Family 서비스에 오류가 발생하면 이 예외가 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceExistsException

요청된 리소스가 존재하지 않거나 명령에 지정된 리전이 아닌 다른 리전에 있습니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

ThrottlingException

요청 제한 때문에 요청이 거부되었습니다.

HTTP 상태 코드: 400

예

예

다음 예에서는 계약을 생성하고 계약 ID를 반환합니다.

```
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
```

샘플 응답

API 호출은 새 계약의 계약 ID를 반환합니다.

```
{
    "AgreementId": "a-11112222333344444"
}
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

CreateConnector

AS2 또는 SFTP 프로토콜의 아웃바운드 연결에 대한 파라미터를 캡처하는 커넥터를 생성합니다. AS2에 대해 커넥터는 외부에서 호스팅되는 AS2 서버로 파일을 전송하는 데 필요합니다. SFTP의 경우 SFTP 서버로 파일을 보내거나 SFTP 서버에서 파일을 받을 때 커넥터가 필요합니다. 커넥터에 대한 자세한 내용은 AS2 커넥터 구성 및 SFTP 커넥터 생성을 참조하십시오.

> **Note**
>
> AS2 (`As2Config`) 또는 SFTP (`SftpConfig`) 에 대해 정확히 하나의 구성 객체를 지정해야 합니다.

구문 요청

```
{
   "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"
```

```
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

AccessRole

커넥터는 AS2 또는 SFTP 프로토콜을 사용하여 파일을 전송하는 데 사용됩니다. 액세스 역할에는 사용할 AWS Identity and Access Management 역할의 Amazon 리소스 이름 (ARN) 을 제공합니다.

AS2 커넥터의 경우

AS2를 사용하여 `StartFileTransfer`를 호출하고 요청 파라미터인 `SendFilePaths`에 파일 경로를 지정하여 파일을 전송할 수 있습니다. 파일의 상위 디렉터리(예: `--send-file-paths /bucket/dir/file.txt`의 경우 상위 디렉터리는 `/bucket/dir/`임)를 사용하여 처리된 AS2 메시지 파일을 임시로 저장하고, 파트너로부터 수신 시 MDN을 저장하고, 전송의 관련 메타데이터를 포함하는 최종 JSON 파일을 작성합니다. 따라서 `AccessRole`은(는) `StartFileTransfer` 요청에 사용된 파일 위치의 상위 디렉터리에 대한 읽기 및 쓰기 액세스를 제공해야 합니다. 또한 `StartFileTransfer`와(과) 함께 전송하려는 파일의 상위 디렉터리에 대한 읽기 및 쓰기 액세스 권한을 제공해야 합니다.

AS2 커넥터에 기본 인증을 사용하는 경우 액세스 역할에는 암호에 대한 `secretsmanager:GetSecretValue` 권한이 필요합니다. Secrets Manager에서 관리 키 대신 고객 AWS 관리 키를 사용하여 암호를 암호화하는 경우 역할에도 해당 키에 대한 `kms:Decrypt` 권한이 필요합니다.

SFTP 커넥터의 경우

액세스 역할이 `StartFileTransfer` 요청에서 사용된 파일 위치의 상위 디렉터리에 대한 읽기 및 쓰기 액세스 권한을 제공하는지 확인하세요. 또한 역할이 `secretsmanager:GetSecretValue` 권한을 제공하는지 확인하십시오. AWS Secrets Manager

타입: 문자열

길이 제약: 최소 길이는 20. 최대 길이는 2,048.

패턴: `arn:.*role/\S+`

필수 여부: 예

As2Config

AS2 커넥터 객체의 파라미터를 포함하는 구조입니다.

타입: As2ConnectorConfig 객체

필수 항목 여부: 아니요

LoggingRole

커넥터가 Amazon S3 이벤트에 대한 CloudWatch 로깅을 활성화할 수 있도록 하는 AWS Identity and Access Management (IAM) 역할의 Amazon 리소스 이름 (ARN) 입니다. 설정하면 로그에서 커넥터 활동을 볼 수 있습니다. CloudWatch

타입: 문자열

길이 제약: 최소 길이는 20. 최대 길이는 2,048.

패턴: `arn:.*role/\S+`

Required: No

SecurityPolicyName

커넥터의 보안 정책 이름을 지정합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 100입니다.

패턴: `TransferSFTPConnectorSecurityPolicy-[A-Za-z0-9-]+`

필수 여부: 아니요

SftpConfig

SFTP 커넥터 객체의 파라미터를 포함하는 구조입니다.

타입: SftpConnectorConfig 객체

필수 여부: 아니요

Tags

커넥터를 그룹화하고 검색하는 데 사용할 수 있는 키-값 쌍입니다. 태그는 어떠한 목적으로 사용자에게 연결되는 메타데이터입니다.

타입: Tag 객체 배열

배열 멤버: 최소 항목 수는 1개입니다. 최대 항목 수는 50개입니다.

필수 여부: 아니요

Url

파트너의 AS2 또는 SFTP 엔드포인트의 URL입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 255입니다.

필수 여부: 예

응답 구문

```
{
   "ConnectorId": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

ConnectorId

커넥터의 고유 식별자로, API 호출 성공 후 반환됩니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `c-([0-9a-f]{17})`

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

AWS Transfer Family 서비스에 오류가 발생하면 이 예외가 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceExistsException

요청된 리소스가 존재하지 않거나 명령에 지정된 리전이 아닌 다른 리전에 있습니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

ThrottlingException

요청 제한 때문에 요청이 거부되었습니다.

HTTP 상태 코드: 400

예

예

다음 예에서는 AS2 커넥터를 생성합니다. 명령에서 다음과 같이 항목을 바꿉니다.

- `url`: 거래 파트너의 AS2 서버의 URL을 제공합니다.
- `your-IAM-role-for-bucket-access`: 파일을 저장하는 데 사용하는 Amazon S3 버킷에 액세스할 수 있는 IAM 역할.

- ID가 포함된 로깅 역할에는 ARN을 사용하십시오. AWS 계정
- AS2 커넥터 구성 파라미터가 포함된 파일의 경로를 제공하세요. AS2 커넥터 구성 개체는 As2에 설명되어 있습니다. ConnectorConfig

```
// 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
```

예

다음 예에서는 SFTP 커넥터를 생성합니다. 명령에서 다음과 같이 항목을 바꿉니다.

- `sftp-server-url`: 파일을 교환하는 SFTP 서버의 URL을 제공합니다.
- `your-IAM-role-for-bucket-access`: 파일을 저장하는 데 사용하는 Amazon S3 버킷에 액세스할 수 있는 IAM 역할.
- ID가 포함된 로깅 역할에는 ARN을 사용하십시오. AWS 계정
- SFTP 커넥터 구성 파라미터가 포함된 파일의 경로를 제공하세요. SFTP 커넥터 구성 개체는 SftpConnectorConfig에 설명되어 있습니다.

```
// 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
```

예

API 호출은 새 커넥터의 커넥터 ID를 반환합니다.

샘플 응답

```
{
    "ConnectorId": "a-11112222333344444"
}
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

CreateProfile

AS2 전송에 사용할 로컬 또는 파트너 프로필을 생성합니다.

구문 요청

```
{
   "As2Id": "string",
   "CertificateIds": [ "string" ],
   "ProfileType": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

As2Id

`As2Id`은(는) RFC 4130에 정의된 AS2-name입니다. 인바운드 전송의 경우 파트너로부터 전송된 AS2 메시지의 `AS2-From` 헤더입니다. 아웃바운드 커넥터의 경우 `StartFileTransfer` API 작업을 사용하여 파트너에게 전송된 AS2 메시지의 `AS2-To` 헤더입니다. 이 ID는 공백을 포함할 수 없습니다.

타입: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: `[\p{Print}\s]*`

필수 여부: 예

CertificateIds

가져온 인증서의 식별자 배열입니다. 프로필 및 파트너 프로필 작업에 이 식별자를 사용합니다.

타입: 문자열 배열

길이 제약 조건: 고정 길이는 22입니다.

패턴: `cert-([0-9a-f]{17})`

필수 여부: 아니요

ProfileType

생성할 프로필 타입을 결정합니다.

- 로컬 프로필을 LOCAL 생성하도록 지정합니다. 로컬 프로필은 AS2 지원 Transfer Family 서버 조직 또는 당사자를 나타냅니다.
- 파트너 프로필을 생성하도록 PARTNER을 지정하세요. 파트너 프로필은 Transfer Family 외부의 원격 조직을 나타냅니다.

타입: 문자열

유효 값: `LOCAL | PARTNER`

필수 여부: 예

Tags

AS2 프로필을 그룹화하고 검색하는 데 사용할 수 있는 키-값 쌍입니다.

타입: Tag 객체 배열

배열 멤버: 최소 항목 수는 1개입니다. 최대 항목 수는 50개입니다.

필수 여부: 아니요

응답 구문

```
{
   "ProfileId": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

ProfileId

AS2 프로필의 고유 식별자로, API 호출 성공 후 반환됩니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `p-([0-9a-f]{17})`

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

ThrottlingException

요청 제한 때문에 요청이 거부되었습니다.

HTTP 상태 코드: 400

예

예

다음 예에서는 프로필을 생성하고 프로필 ID를 반환합니다.

인증서 ID는 `import-certificate`을 실행할 때 생성되는데, 하나는 서명 인증서용이고 다른 하나는 암호화 인증서용입니다.

```
aws transfer create-profile --as2-id MYCORP --certificate-ids c-abcdefg123456hijk
               c-987654aaaa321bbbb
```

샘플 응답

API 호출은 새 프로필의 프로필 ID를 반환합니다.

```
{
    "ProfileId": "p-11112222333344444"
}
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

CreateServer

AWS에서 선택한 파일 전송 프로토콜을 기반으로 자동 조정 가상 서버를 인스턴스화합니다. 파일 전송 프로토콜 사용 서버를 업데이트하거나 사용자와 작업할 때 새로 생성된 서버에 할당된 서비스 생성 `ServerId` 속성을 사용합니다.

구문 요청

```
{
   "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"
         }
      ]
   }
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

Certificate

AWS Certificate Manager (ACM) 인증서의 Amazon 리소스 이름(ARN)입니다. `Protocols`이(가) FTPS(으)로 설정된 경우에 필요합니다.

새 공인 인증서를 요청하려면 AWS Certificate Manager 사용 설명서의 공개 인증서 요청을 참조하십시오.

기존 인증서를 ACM으로 가져오려면 사용 설명서의 AWS Certificate Manager ACM으로 인증서 가져오기를 참조하십시오.

사설 IP 주소를 통해 FTPS를 사용하도록 사설 인증서를 요청하려면 사용 설명서의 사설 인증서 요청을 참조하십시오. AWS Certificate Manager

다음 암호화 알고리즘 및 키 크기를 사용하는 인증서가 지원됩니다:

- 2048비트 RSA(RSA_2048)
- 4096비트 RSA(RSA_4096)
- 타원 프라임 곡선 256비트(EC_prime256v1)
- 타원 프라임 곡선 384 비트(EC_secp384r1)
- 타원 프라임 곡선 521비트(EC_secp521r1)

> **Note**
>
> 인증서는 FQDN 또는 IP 주소가 지정되고 발행자에 대한 정보가 있는 유효한 SSL/TLS X.509 버전 3 인증서여야 합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 1600입니다.

필수 여부: 아니요

Domain

파일 전송에 사용되는 스토리지 시스템의 도메인을 지정합니다. Amazon Simple Storage Service (Amazon S3)와 Amazon Elastic File System (Amazon EFS)의 두 가지 도메인을 사용할 수 있습니다. 기본값은 S3입니다.

> **Note**
>
> 서버가 생성된 후에는 도메인을 변경할 수 없습니다.

타입: 문자열

유효 값: `S3 | EFS`

필수 여부: 아니요

EndpointDetails

서버에 대해 구성된 Virtual Private Cloud(VPC) 엔드포인트 설정입니다. VPC 내에서 엔드포인트를 호스팅할 때 VPC 내의 리소스에만 액세스할 수 있도록 하거나 탄력적 IP 주소를 연결하여 인터넷을 통해 클라이언트에 액세스하도록 할 수 있습니다. VPC의 기본 보안 그룹은 엔드포인트에 자동으로 할당됩니다.

타입: EndpointDetails 객체

필수 여부: 아니요

EndpointType

서버에서 사용할 엔드포인트 타입입니다. 서버의 엔드포인트를 공개적으로 액세스(PUBLIC)하거나 VPC 내부에서 호스팅하도록 선택할 수 있습니다. VPC에서 호스팅되는 엔드포인트를 사용하면 VPC 내에서만 서버 및 리소스에 대한 액세스를 제한하거나 탄력적 IP 주소를 직접 연결하여 인터넷에 연결하도록 선택할 수 있습니다.

> **Note**
>
> 2021년 5월 19일 이후에는 2021년 5월 19일 이전에 계정을 아직 생성하지 않은 AWS 계정 경우 계정을 사용하여 `EndpointType=VPC_ENDPOINT` 서버를 생성할 수 없습니다. 2021년 5월 AWS 계정 19일 또는 그 이전에 서버를 이미 생성한 경우 영향을 받지 않습니다. `EndpointType=VPC_ENDPOINT` 이 날짜 이후에는 `EndpointType=VPC`를 사용하세요.
>
> 자세한 내용은 VPC_ENDPOINT 사용 중단를 참조하세요.
>
> VPC를 `EndpointType`(으)로 사용하는 것이 좋습니다. 이 엔드포인트 타입을 사용하면 최대 3개의 탄력적 IPv4 주소(BYO IP 포함)를 서버 엔드포인트에 직접 연결하고 VPC 보안 그룹을 사용하여 클라이언트의 퍼블릭 IP 주소로부터 트래픽을 제한할 수 있습니다. `EndpointType`를 `VPC_ENDPOINT`(으)로 설정하면 이 작업을 할 수 없습니다.

타입: 문자열

유효 값: `PUBLIC | VPC | VPC_ENDPOINT`

필수 여부: 아니요

HostKey

SFTP 지원 서버에 사용할 RSA, ECDSA 또는 ED25519 프라이빗 키입니다. 키를 교체하거나 다른 알고리즘을 사용하는 활성 키 집합이 있는 경우, 여러 호스트 키를 추가할 수 있습니다.

다음 명령을 사용하여 암호가 없는 RSA 2048비트 키 생성:

```
ssh-keygen -t rsa -b 2048 -N "" -m PEM -f my-new-server-key.
```

`-b` 옵션에는 최소값인 2048을 사용합니다. 3072 또는 4096을 사용하여 더 강력한 키를 만들 수 있습니다.

다음 명령을 사용하여 암호가 없는 ECDSA 256비트 키 생성:

```
ssh-keygen -t ecdsa -b 256 -N "" -m PEM -f my-new-server-key.
```

ECDSA에 대한 -b 옵션의 유효한 값은 256, 384, 521입니다.

다음 명령을 사용하여 암호가 없는 ED25519 키 생성:

```
ssh-keygen -t ed25519 -N "" -f my-new-server-key.
```

이 모든 명령을 원하는 my-new-server-key문자열로 바꿀 수 있습니다.

> **Important**
>
> 기존 사용자를 기존 SFTP 지원 서버에서 새 SFTP 지원 서버로 마이그레이션할 계획이 아니라면 호스트 키를 업데이트하지 마세요. 실수로 서버의 호스트 키를 변경하면 작업에 영향을 줄 수 있습니다.

자세한 내용은 사용 설명서의 SFTP 지원 서버의 호스트 키 업데이트를 참조하십시오. AWS Transfer Family

타입: 문자열

길이 제약: 최소 길이는 0입니다. 최대 길이는 4096입니다.

필수 여부: 아니요

IdentityProviderDetails

`IdentityProviderType`이(가) `AWS_DIRECTORY_SERVICE`, `AWS_LAMBDA` 또는 `API_GATEWAY`(으)로 설정된 경우 필수입니다. `AWS_DIRECTORY_SERVICE`의 디렉터리를 사용하거나 API 게이트웨이 URL을 포함하여 고객 제공 인증 API를 호출하는 데 필요한 모든 정보를 포함하는 배열을 수락합니다. `IdentityProviderType`이 `SERVICE_MANAGED`로 설정된 경우에 필요합니다.

타입: IdentityProviderDetails 객체

필수 여부: 아니요

IdentityProviderType

서버 인증 모드. 기본값은 이며`SERVICE_MANAGED`, 이를 통해 서비스 내에서 사용자 자격 증명을 저장하고 액세스할 수 있습니다. AWS Transfer Family

온-프레미스 환경에서 AWS Directory Service for Microsoft Active Directory 또는 AD Connector를 AWS 사용하여 Microsoft Active Directory 내의 Active Directory 그룹에 대한 액세스를 제공하는 데 사용합니다`AWS_DIRECTORY_SERVICE`. 또한 이 옵션을 사용하려면 `IdentityProviderDetails` 파라미터를 사용하여 디렉터리 ID를 제공해야 합니다.

`API_GATEWAY` 값을 사용하여 선택하는 자격 증명 제공자와 통합합니다. `API_GATEWAY`를 설정하려면 `IdentityProviderDetails` 파라미터를 사용하여 인증을 요구하도록 Amazon API Gateway 엔드포인트 URL을 제공해야 합니다.

`AWS_LAMBDA`값을 사용하여 AWS Lambda 함수를 ID 공급자로 직접 사용할 수 있습니다. 이 값을 선택하는 경우 `IdentityProviderDetails` 데이터 타입에 대한 `Function` 파라미터에서 lLambda 함수에 대한 ARN을 지정해야 합니다.

타입: 문자열

유효 값: `SERVICE_MANAGED | API_GATEWAY | AWS_DIRECTORY_SERVICE | AWS_LAMBDA`

필수 여부: 아니요

LoggingRole

서버가 Amazon S3 또는 Amazon EFSevents에 대한 아마존 CloudWatch 로깅을 활성화할 수 있도록 허용하는 AWS Identity and Access Management (IAM) 역할의 Amazon 리소스 이름 (ARN). 설정하면 로그에서 사용자 활동을 볼 수 있습니다. CloudWatch

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 2,048.

패턴: `(|arn:.*role/\S+)`

필수 여부: 아니요

PostAuthenticationLoginBanner

사용자가 서버에 연결할 때 표시할 문자열을 지정합니다. 이 문자열은 사용자가 인증한 후에 표시됩니다.

> **Note**
>
> SFTP 프로토콜은 인증 후 디스플레이 배너를 지원하지 않습니다.

타입: 문자열

길이 제약: 최소 길이는 0입니다. 최대 길이는 4096자입니다.

패턴: `[\x09-\x0D\x20-\x7E]*`

필수 여부: 아니요

PreAuthenticationLoginBanner

사용자가 서버에 연결할 때 표시할 문자열을 지정합니다. 이 문자열은 사용자가 인증하기 전에 표시됩니다. 예를 들어 다음 배너는 시스템 사용에 대한 세부 정보를 표시합니다.

```
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.
```

타입: 문자열

길이 제약: 최소 길이는 0입니다. 최대 길이는 4096자입니다.

패턴: `[\x09-\x0D\x20-\x7E]*`

필수 여부: 아니요

ProtocolDetails

서버에 대해 구성된 프로토콜 설정입니다.

- 수동 모드(FTP 및 FTPS 프로토콜의 경우)를 표시하려면 `PassiveIp` 파라미터를 사용합니다. 방화벽, 라우터 또는 로드 밸런서의 외부 IP 주소와 같은 점으로 분리된 단일 쿼드 IPv4 주소를 입력합니다.
- Amazon S3 버킷에 업로드하는 파일에 대해 클라이언트가 `SETSTAT` 명령을 사용하려 할 때 생성되는 오류를 무시하려면 `SetStatOption` 파라미터를 사용합니다. AWS Transfer Family 서버에서 `SETSTAT` 명령을 무시하고 SFTP 클라이언트를 변경할 필요 없이 파일을 업로드하도록 하려면 값을 로 설정합니다. `ENABLE_NO_OP` `SetStatOption`파라미터를 로 `ENABLE_NO_OP` 설정하면 Transfer Family가 Amazon Logs에 CloudWatch 로그 항목을 생성하여 고객이 `SETSTAT` 전화를 거는 시기를 확인할 수 있습니다.
- AWS Transfer Family 서버가 고유한 세션 ID를 통해 최근 협상된 세션을 재개할지 여부를 확인하려면 파라미터를 사용하십시오. `TlsSessionResumptionMode`

- `As2Transports`는 AS2 메시지의 전송 방법을 나타냅니다. 현재는 HTTP만 지원됩니다.

타입: ProtocolDetails 객체

필수 여부: 아니요

Protocols

파일 전송 프로토콜 클라이언트가 서버의 엔드포인트에 연결할 수 있는 파일 전송 프로토콜을 지정합니다. 사용 가능한 프로토콜은 다음과 같습니다:

- SFTP (Secure Shell(SSH) File Transfer Protocol):: SSH를 통한 파일 전송
- FTPS (File Transfer Protocol Secure): TLS 암호화를 사용한 파일 전송
- FTP (File Transfer Protocol): 암호화되지 않은 파일 전송
- AS2(적용 설명 2): 구조화된 데이터를 전송하는 데 사용됩니다. business-to-business

> **Note**
>
> - 선택하는 경우 클라이언트가 FTPS FTPS를 통해 서버에 연결할 때 서버를 식별하는 데 사용되는 ACM AWS Certificate Manager (저장된 인증서) 을 선택해야 합니다.
> - `Protocol`에 `FTP` 또는 `FTPS`이(가) 포함된 경우 `EndpointType`은(는) `VPC`이고 `IdentityProviderType`은(는) `AWS_DIRECTORY_SERVICE`, `AWS_LAMBDA` 또는 `API_GATEWAY`여야 합니다.
> - `Protocol`에 `FTP`이(가) 포함된 경우 `AddressAllocationIds`를 연결할 수 없습니다.
> - `Protocol`이 `SFTP`로만 설정된 경우 `EndpointType`을 `PUBLIC`으로 설정하고 `IdentityProviderType`을 지원되는 ID 타입(`SERVICE_MANAGED`, `AWS_DIRECTORY_SERVICE`, `AWS_LAMBDA`, `API_GATEWAY`) 중 하나로 설정할 수 있습니다.
> - `Protocol`에 `AS2`이(가) 포함된 경우 `EndpointType`은(는) `VPC`여야 하고, 도메인은 Amazon S3여야 합니다.

타입: 문자열 배열

배열 멤버: 최소수는 1개입니다. 최대 항목 수는 4개입니다.

유효 값: `SFTP | FTP | FTPS | AS2`

필수 여부: 아니요

S3StorageOptions

Amazon S3 디렉터리의 성능이 최적화되었는지 여부를 지정합니다. 이 옵션은 기본적으로 비활성화되어 있습니다.

기본적으로 홈 디렉터리 매핑에는 a가 있습니다. TYPE DIRECTORY 이 옵션을 활성화한 경우 매핑에 파일 대상이 포함되도록 하려면 HomeDirectoryMapEntry Type 를 FILE 명시적으로 설정해야 합니다.

유형: S3StorageOptions객체

필수 항목 여부: 아니요

SecurityPolicyName

서버의 보안 정책 이름을 지정합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 100입니다.

패턴: `Transfer[A-Za-z0-9]*SecurityPolicy-[A-Za-z0-9-]+`

필수 여부: 아니요

StructuredLogDestinations

사용자의 서버 로그가 전송될 로그 그룹을 지정합니다.

로그 그룹을 지정하려면 기존 로그 그룹의 ARN을 제공해야 합니다. 이 경우, 로그 그룹의 형식은 다음과 같습니다:

`arn:aws:logs:region-name:amazon-account-id:log-group:log-group-name:*`

예제: `arn:aws:logs:us-east-1:111122223333:log-group:mytestgroup:*`

이전에 서버의 로그 그룹을 지정한 경우, `update-server` 호출 시 이 파라미터에 빈 값을 제공하여 로그 그룹을 지우고 사실상 구조화된 로깅을 끌 수 있습니다. 예:

```
update-server --server-id s-1234567890abcdef0 --structured-log-
destinations
```

타입: 문자열 배열

배열 멤버: 최소 항목 수는 0개입니다. 최대 항목 수는 1개입니다.

길이 제약 조건: 최소 길이는 20입니다. 최대 길이는 1600입니다.

패턴: `arn:\S+`

필수 여부: 아니요

Tags

서버의 그룹화 및 검색에 사용될 수 있는 키-값 쌍입니다.

타입: Tag 객체 배열

배열 멤버: 최소 항목 수는 1개입니다. 최대 항목 수는 50개입니다.

필수 여부: 아니요

WorkflowDetails

할당할 워크플로의 워크플로 ID와 워크플로 실행에 사용되는 실행 역할을 지정합니다.

파일이 완전히 업로드될 때 실행되는 워크플로 외에도 `WorkflowDetails`에는 부분 업로드 시 실행할 워크플로의 워크플로 ID와 실행 역할이 포함될 수 있습니다. 파일이 업로드되는 동안 서버 세션 연결이 끊기면 부분 업로드가 수행됩니다.

타입: WorkflowDetails 객체

필수 항목 여부: 아니요

응답 구문

```
{
   "ServerId": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

ServerId

생성되는 서버의 서비스 할당 ID입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

Errors

모든 작업에서 공통적으로 발생하는 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

AccessDeniedException

이 작업을 수행할 수 있는 충분한 액세스 권한이 없습니다.

HTTP 상태 코드: 400

InternalServiceError

AWS Transfer Family 서비스에 오류가 발생하면 이 예외가 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceExistsException

요청된 리소스가 존재하지 않거나 명령에 지정된 리전이 아닌 다른 리전에 있습니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

ThrottlingException

요청 제한 때문에 요청이 거부되었습니다.

HTTP 상태 코드: 400

예

예

다음 예에서는 VPC_ENDPOINT를 사용하여 새 표를 만듭니다.

샘플 요청

```
{
   "EndpointType": "VPC",
   "EndpointDetails":...,
   "HostKey": "Your RSA private key",
   "IdentityProviderDetails": "IdentityProvider",
   "IdentityProviderType": "SERVICE_MANAGED",
   "LoggingRole": "CloudWatchLoggingRole",
   "Tags": [
      {
         "Key": "Name",
         "Value": "MyServer"
      }
   ]
}
```

예

다음은 이 API 직접 호출에 대한 샘플 응답입니다.

샘플 응답

```
{
   "ServerId": "s-01234567890abcdef"
}
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

CreateUser

사용자를 생성하고 기존 File Transfer 프로토콜 지원 서버와 연결합니다. `IdentityProviderType`이(가) `SERVICE_MANAGED`(으)로 설정된 서버와만 사용자를 생성하여 연결할 수 있습니다. 의 매개변수를 사용하여 사용자 이름을 지정하고, 홈 디렉터리를 설정하고, 사용자의 공개 키를 저장하고, 사용자 AWS Identity and Access Management (IAM) 역할을 할당할 수 있습니다. `CreateUser` 또한 세션 정책을 선택적으로 추가하고 사용자 그룹화 및 검색에 사용될 수 있는 태그가 포함된 메타데이터를 할당할 수 있습니다.

구문 요청

```
{
   "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"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

HomeDirectory

사용자가 클라이언트를 사용하여 서버에 로그인하는 경우 사용자를 위한 랜딩 디렉터리(폴더).

HomeDirectory의 예: /bucket_name/home/mydirectory

> **Note**
>
> HomeDirectory 파라미터는 HomeDirectoryType이(가) PATH(으)로 설정된 경우에만 사용합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 1024입니다.

패턴: (|/.*)

필수 여부: 아니요

HomeDirectoryMappings

사용자에게 표시할 Amazon S3 또는 Amazon EFS 경로 및 키와 이러한 경로 및 키를 표시할 방법을 지정하는 논리적 디렉터리 매핑입니다. Entry 및 Target 쌍을 지정해야 합니다. 여기서 Entry는 경로가 표시되는 방식을 보여주고 Target는 실제 Amazon S3 경로입니다. 대상만 지정하는 경우, 그대로 표시됩니다. 또한 AWS Identity and Access Management (IAM) 역할이 경로에 대한 액세스를 제공하는지 확인해야 합니다. Target 이 값은 LOGICAL로 설정된 경우에만 HomeDirectoryType를 설정할 수 있습니다.

다음은 Entry 및 Target 쌍의 예입니다.

```
[ { "Entry": "/directory1", "Target": "/bucket_name/home/
mydirectory" } ]
```

대부분의 경우 세션 정책 대신 이 값을 사용하여 사용자를 지정된 홈 디렉터리('chroot')로 제한할 수 있습니다. 이렇게 하려면 Entry를 /(으)로 설정하고 사용자가 로그인할 때 홈 디렉터리에 대해 확인해야 하는 값을 Target(으)로 설정할 수 있습니다.

다음은 chroot에 대한 Entry 및 Target 쌍의 예입니다.

```
[ { "Entry": "/", "Target": "/bucket_name/home/mydirectory" } ]
```

타입: HomeDirectoryMapEntry 객체 배열

어레이 멤버: 최소 항목 수 1개. 최대 항목 수는 50000개입니다.

필수 여부: 아니요

HomeDirectoryType

사용자가 서버에 로그인하는 경우 홈 디렉터리가 될 랜딩 디렉터리(폴더) 타입입니다. PATH로 설정하면, 사용자는 파일 전송 프로토콜 클라이언트에서와 같이 절대 Amazon S3 버킷 또는 EFS 경로를 볼 수 있습니다. LOGICAL로 설정하면, HomeDirectoryMappings에서 Amazon S3 또는 Amazon EFS 경로를 사용자에게 표시할 방법에 대한 매핑을 제공해야 합니다.

> **Note**
>
> HomeDirectoryType이 LOGICAL인 경우, HomeDirectoryMappings 파라미터를 사용하여 매핑을 제공해야 합니다. 반면에 HomeDirectoryType이 PATH인 경우, HomeDirectory 파라미터를 사용하여 절대 경로를 제공하세요. 템플릿에 HomeDirectory 및 HomeDirectoryMappings를 모두 포함할 수는 없습니다.

타입: 문자열

유효 값: PATH | LOGICAL

필수 여부: 아니요

Policy

여러 사용자가 동일한 AWS Identity and Access Management (IAM) 역할을 사용할 수 있도록 하기 위한 사용자 세션 정책. 이 정책은 Amazon S3 버킷의 부분에 대한 사용자의 액세스 범위를 축소합니다. 이 정책 내에서 사용할 수 있는 변수는 ${Transfer:UserName}, ${Transfer:HomeDirectory} 및 ${Transfer:HomeBucket}입니다.

> **Note**
>
> 이 정책은 ServerId의 도메인이 Amazon S3인 경우에만 적용됩니다. Amazon EFS는 세션 정책을 사용하지 않습니다.
> 세션 정책의 경우 정책을 정책의 Amazon 리소스 이름 (ARN) 대신 JSON Blob으로 AWS Transfer Family 저장합니다. 정책을 JSON Blob으로 저장하고 Policy 인수로 전달합니다.

세션 정책의 예는 세션 정책 예를 참조하세요.
자세한 내용은 AWS 보안 토큰 서비스 AssumeRoleAPI 참조를 참조하십시오.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 2048입니다.

필수 여부: 아니요

PosixProfile

Amazon EFS 파일 시스템에 대한 사용자의 액세스를 제어하는 사용자 ID(Uid), 그룹 ID(Gid) 및 보조 그룹 ID(SecondaryGids)를 포함한 전체 POSIX 자격 증명을 지정합니다. Amazon EFS의 파일 및 디렉터리에 설정된 POSIX 권한에 따라 Amazon EFS 파일 시스템에서 파일을 송수신할 때 사용자에게 제공되는 액세스 수준이 결정됩니다.

타입: PosixProfile 객체

필수 항목 여부: 아니요

Role

Amazon S3 버킷 또는 Amazon EFS 파일 시스템에 대한 사용자의 액세스를 제어하는 AWS Identity and Access Management (IAM) 역할의 Amazon 리소스 이름 (ARN). 이 역할에 연결된 정책은 Amazon S3 버킷 또는 Amazon EFS 파일 시스템에 대한 파일 송수신 시 사용자에게 제공할 액세스의 수준을 결정합니다. 또한 IAM 역할에는 사용자의 전송 요청을 처리할 때 서버가 해당 리소스에 액세스할 수 있도록 허용하는 신뢰 관계가 포함되어야 합니다.

타입: 문자열

길이 제약: 최소 길이는 20. 최대 길이는 2,048.

패턴: `arn:.*role/\S+`

필수 여부: 예

ServerId

서버 인스턴스에 대해 시스템에서 할당한 고유 식별자입니다. 이 항목은 사용자를 추가한 특정 서버입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

필수 사항 여부: Yes

SshPublicKeyBody

사용자를 서버와 인증하는 데 사용되는 Secure Shell(SSH) 키의 퍼블릭 부분입니다.

세 가지 표준 SSH 퍼블릭 키 형식 요소는 <key type>, <body base64>, <comment>(옵션)이며 각 요소 사이에 공백이 있습니다.

AWS Transfer Family RSA, ECDSA 및 ED25519 키를 수락합니다.

- RSA 키의 경우 키 타입은 `ssh-rsa`입니다.
- ED25519 키의 경우 키 타입은 `ssh-ed25519`입니다.
- ECDSA 키의 경우 키 타입은 생성한 키 크기에 따라 `ecdsa-sha2-nistp256`, `ecdsa-sha2-nistp384` 또는 `ecdsa-sha2-nistp521`입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 2048입니다.

필수 여부: 아니요

Tags

사용자의 그룹화 및 검색에 사용될 수 있는 키-값 쌍입니다. 태그는 어떠한 목적으로 사용자에게 연결되는 메타데이터입니다.

타입: Tag 객체 배열

배열 멤버: 최소 항목 수는 1개입니다. 최대 항목 수는 50개입니다.

필수 여부: 아니요

UserName

사용자를 식별하고 `ServerId`와 연결되는 고유한 문자열입니다. 이 사용자 이름은 3~100자여야 합니다. a-z, A-Z, 0-9, _(밑줄), -(하이픈), .(마침표) 및 @ 기호를 유효한 문자로 사용할 수 있습니다. 사용자 이름은 하이픈, 마침표 및 @ 기호로 시작할 수 없습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 3입니다. 최대 길이는 100입니다.

패턴: `[\w][\w@.-]{2,99}`

필수 항목 여부: 예

응답 구문

```
{
   "ServerId": "string",
   "UserName": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

ServerId

사용자가 연결된 서버의 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

UserName

Transfer Family 사용자를 식별하는 고유한 문자열입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 3입니다. 최대 길이는 100입니다.

패턴: `[\w][\w@.-]{2,99}`

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

서비스에 오류가 발생하면 이 예외가 발생합니다. AWS Transfer Family

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceExistsException

요청된 리소스가 존재하지 않거나 명령에 지정된 리전이 아닌 다른 리전에 있습니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

예

예

사용자를 생성하려면 먼저 파라미터를 JSON 파일(예: `createUserParameters`)에 저장한 다음 create-user API 명령을 실행하면 됩니다.

```
{
    "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"
```

```
}
```

샘플 요청

```
aws transfer create-user --cli-input-json file://createUserParameters
```

샘플 응답

```
{
    "ServerId": ""s-1111aaaa2222bbbb3",
    "UserName": "bobusa-API"
}
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

CreateWorkflow

파일 전송이 완료된 후 워크플로가 호출하는 지정된 단계 및 단계 세부 정보를 사용하여 워크플로를 생성할 수 있습니다. 워크플로를 생성한 후 `CreateServer` 및 `UpdateServer` 작업에서 `workflow-details` 필드를 지정하여 생성된 워크플로를 전송 서버와 연결할 수 있습니다.

구문 요청

```
{
   "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"
      }
   ]
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

Description

워크플로에 대한 텍스트 설명입니다.

타입: 문자열

길이 제약: 최소 길이는 0. 최대 길이는 256입니다.

패턴: `[\w- ]*`

Required: No

OnExceptionSteps

워크플로를 실행하는 동안 오류가 발생할 경우 수행할 단계(작업)를 지정합니다.

> **Note**
>
> 사용자 지정 단계의 경우 Lambda 함수가 콜백 API로 `FAILURE`를 전송하여 예외 단계를 시작해야 합니다. 또한 Lambda가 제한 시간이 초과되기 전에 `SUCCESS`를 전송하지 않는 경우 예외 단계가 실행됩니다.

타입: WorkflowStep 객체 배열

배열 멤버: 최소 항목 수는 0개입니다. 최대 항목 수는 8개입니다.

필수 여부: 아니요

Steps

지정된 워크플로에 있는 단계에 대한 세부 정보를 지정합니다.

`TYPE`이(가) 다음 중 이 단계에서 취해야 할 작업을 지정합니다.

- **`COPY`** - 다른 위치에 파일을 복사합니다.
- **`CUSTOM`**- AWS Lambda 함수 타겟을 사용하여 사용자 지정 단계를 수행합니다.
- **`DECRYPT`** - 업로드되기 전에 암호화된 파일을 복호화합니다.

- **DELETE** - 파일을 삭제합니다.
- **TAG** - 파일에 태그를 추가합니다.

> **Note**
>
> 현재 복사 및 태그 지정은 S3에서만 지원됩니다.

파일 위치의 경우 Amazon S3 버킷과 키, 또는 Amazon EFS 파일 시스템 ID 및 경로를 지정합니다.

타입: WorkflowStep 객체 배열

배열 멤버: 최소 항목 수는 0개입니다. 최대 항목 수는 8개입니다.

필수 여부: 예

Tags

워크플로 그룹화 및 검색에 사용될 수 있는 키-값 쌍입니다. 태그는 어떠한 목적으로 워크플로에 연결되는 메타데이터입니다.

타입: Tag 객체 배열

배열 멤버: 최소 항목 수는 1개입니다. 최대 항목 수는 50개입니다.

필수 여부: 아니요

응답 구문

```
{
   "WorkflowId": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

WorkflowId

워크플로의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `w-([a-z0-9]{17})`

Errors

모든 작업에서 공통적으로 발생하는 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

AccessDeniedException

이 작업을 수행할 수 있는 충분한 액세스 권한이 없습니다.

HTTP 상태 코드: 400

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceExistsException

요청된 리소스가 존재하지 않거나 명령에 지정된 리전이 아닌 다른 리전에 있습니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

ThrottlingException

요청 제한 때문에 요청이 거부되었습니다.

HTTP 상태 코드: 400

예

예

다음 예와 같이 워크플로 단계 정보를 텍스트 파일에 저장한 다음 해당 파일을 사용하여 워크플로를 생성할 수 있습니다. 다음 예에서는 워크플로 단계를 *example-file.json* (명령을 실행한 동일한 폴더)에 저장했으며 버지니아 북부(us-east-1) 리전에 워크플로를 생성하려고 한다고 가정합니다.

```
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}"
      }
   }
]
```

예

이 CreateWorkflow 호출은 새 워크플로의 워크플로 ID를 반환합니다.

샘플 응답

```
{
    "WorkflowId": "w-1234abcd5678efghi"
}
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

DeleteAccess

`ServerID`및 `ExternalID` 파라미터에 지정된 액세스를 삭제할 수 있습니다.

구문 요청

```
{
   "ExternalId": "string",
   "ServerId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

ExternalId

디렉터리 내의 특정 그룹을 식별하는 데 필요한 고유 식별자입니다. 연결하는 그룹의 사용자는 사용하는 활성화된 프로토콜을 통해 Amazon S3 또는 Amazon EFS 리소스에 액세스할 수 AWS Transfer Family있습니다. 그룹 이름을 아는 경우 PowerShell Windows를 사용하여 다음 명령을 실행하여 SID 값을 볼 수 있습니다.

```
Get-ADGroup -Filter {samAccountName -like "YourGroupName*"} -Properties
* | Select SamAccountName,ObjectSid
```

이 명령에서 Active Directory 그룹의 YourGroupName이름으로 바꾸십시오.

이 파라미터를 확인하는 데 사용되는 정규 표현식은 공백 없이 대문자 및 소문자 영숫자로 구성된 문자의 문자열입니다. 밑줄이나 또한 =,.@:/- 문자도 포함할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: `S-1-[\d-]+`

필수 사항 여부: Yes

ServerId

이 사용자가 할당한 서버에 대해 시스템에서 할당한 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반송합니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

AWS Transfer Family 서비스에 오류가 발생하면 이 예외가 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

DeleteAgreement

제공된 AgreementId에 지정된 계약을 삭제합니다.

구문 요청

```
{
   "AgreementId": "string",
   "ServerId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

AgreementId

계약의 고유 식별자입니다. 계약을 생성하면 이 식별자가 반환됩니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: a-([0-9a-f]{17})

필수 사항 여부: Yes

ServerId

삭제하려는 계약과 관련된 서버 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: s-([0-9a-f]{17})

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반송합니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

DeleteCertificate

`CertificateId`파라미터에서 지정된 인증서를 삭제합니다.

구문 요청

```
{
   "CertificateId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

CertificateId

삭제하려는 인증서 개체의 식별자.

타입: 문자열

길이 제약 조건: 고정 길이는 22입니다.

패턴: `cert-([0-9a-f]{17})`

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반송합니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

DeleteConnector

제공된 ConnectorId에 지정된 커넥터를 삭제합니다.

구문 요청

```
{
   "ConnectorId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

ConnectorId

커넥터의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: c-([0-9a-f]{17})

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반송합니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

DeleteHostKey

HostKeyId 파라미터에 지정된 호스트 키를 삭제합니다.

구문 요청

```
{
   "HostKeyId": "string",
   "ServerId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

HostKeyId

삭제 중인 호스트 키의 식별자.

타입: 문자열

길이 제약 조건: 고정 길이는 25입니다.

패턴: hostkey-[0-9a-f]{17}

필수 사항 여부: Yes

ServerId

삭제 중인 호스트 키를 포함한 서버의 식별자.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: s-([0-9a-f]{17})

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반송합니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

ThrottlingException

요청 제한 때문에 요청이 거부되었습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK

- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

DeleteProfile

`ProfileId` 파라미터에 지정된 프로필을 삭제합니다.

구문 요청

```
{
   "ProfileId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

ProfileId

삭제하려는 프로필의 식별자.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `p-([0-9a-f]{17})`

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반송합니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

DeleteServer

지정한 파일 전송 프로토콜 지원 서버를 삭제합니다.

이 작업에서 응답이 반환되지 않습니다.

구문 요청

```
{
   "ServerId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

ServerId

서버 인스턴스에 대해 시스템에서 할당한 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반송합니다.

Errors

모든 작업에서 공통적으로 발생하는 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

AccessDeniedException

이 작업을 수행할 수 있는 충분한 액세스 권한이 없습니다.

HTTP 상태 코드: 400

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

예

예

다음 예에서는 서버를 삭제합니다.

샘플 요청

```
{
   "ServerId": "s-01234567890abcdef"
}
```

예

성공하면 아무 것도 반환되지 않습니다.

샘플 응답

```
{
```

```
}
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

DeleteSshPublicKey

사용자의 Secure Shell(SSH) 퍼블릭 키를 삭제합니다.

구문 요청

```
{
   "ServerId": "string",
   "SshPublicKeyId": "string",
   "UserName": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

ServerId

사용자가 할당된 File Transfer 프로토콜 지원 서버 인스턴스에 대해 시스템에서 할당한 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

필수 사항 여부: Yes

SshPublicKeyId

사용자의 특정 SSH 키를 참조하는 데 사용되는 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 21입니다.

패턴: `key-[0-9a-f]{17}`

필수 사항 여부: Yes

UserName

퍼블릭 키를 삭제할 사용자를 식별하는 고유한 문자열입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 3입니다. 최대 길이는 100입니다.

패턴: `[\w][\w@.-]{2,99}`

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반송합니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

ThrottlingException

요청 제한 때문에 요청이 거부되었습니다.

HTTP 상태 코드: 400

예

예

다음 예는 사용자의 SSH 퍼블릭 키를 삭제합니다.

샘플 요청

```
{
   "ServerId": "s-01234567890abcdef",
   "SshPublicKeyId": "MyPublicKey",
   "UserName": "my_user"
}
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

DeleteUser

지정한 파일 전송 프로토콜 지원 서버에 속한 사용자를 삭제합니다.

이 작업에서 응답이 반환되지 않습니다.

> **Note**
>
> 서버에서 사용자를 삭제하면 사용자 정보가 손실됩니다.

구문 요청

```
{
   "ServerId": "string",
   "UserName": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

ServerId

사용자가 할당된 서버 인스턴스에 대해 시스템에서 할당한 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

필수 여부: 예

UserName

서버에서 삭제되는 사용자를 식별하는 고유한 문자열입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 3입니다. 최대 길이는 100입니다.

패턴: `[\w][\w@.-]{2,99}`

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반송합니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

예

예

다음 예에서는 Transfer Family 사용자를 삭제합니다.

샘플 요청

```
{
   "ServerId": "s-01234567890abcdef",
   "UserNames": "my_user"
}
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

DeleteWorkflow

지정된 워크플로를 삭제합니다.

구문 요청

```
{
   "WorkflowId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

WorkflowId

워크플로의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `w-([a-z0-9]{17})`

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반송합니다.

Errors

모든 작업에서 공통적으로 발생하는 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

AccessDeniedException

이 작업을 수행할 수 있는 충분한 액세스 권한이 없습니다.

HTTP 상태 코드: 400

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

DescribeAccess

특정 File Transfer 프로토콜 지원 서버에 할당된 액세스 권한을 해당 `ServerId` 속성과 해당 `ExternalId` 속성으로 식별하여 설명합니다.

이 직접 호출의 응답은 지정된 `ServerId` 값과 연결된 액세스 속성을 반환합니다.

구문 요청

```
{
   "ExternalId": "string",
   "ServerId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

ExternalId

디렉터리 내의 특정 그룹을 식별하는 데 필요한 고유 식별자입니다. 연결하는 그룹의 사용자는 사용하는 활성화된 프로토콜을 통해 Amazon S3 또는 Amazon EFS 리소스에 액세스할 수 AWS Transfer Family있습니다. 그룹 이름을 아는 경우 PowerShell Windows를 사용하여 다음 명령을 실행하여 SID 값을 볼 수 있습니다.

`Get-ADGroup -Filter {samAccountName -like "YourGroupName*"} -Properties * | Select SamAccountName,ObjectSid`

이 명령에서 Active Directory 그룹의 YourGroupName이름으로 바꾸십시오.

이 파라미터를 확인하는 데 사용되는 정규 표현식은 공백 없이 대문자 및 소문자 영숫자로 구성된 문자의 문자열입니다. 밑줄이나 또한 =,.@:/- 문자도 포함할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: `S-1-[\d-]+`

필수 사항 여부: Yes

ServerId

이 액세스 권한이 할당된 서버에 대해 시스템에서 할당한 고유 식별자.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

필수 항목 여부: 예

응답 구문

```
{
   "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"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

Access

액세스 권한이 연결된 서버의 외부 식별자.

타입: DescribedAccess 객체

ServerId

이 액세스 권한이 할당된 서버에 대해 시스템에서 할당한 고유 식별자.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

AWS Transfer Family 서비스에 오류가 발생하면 이 예외가 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

DescribeAgreement

AgreementId로 식별되는 계약을 설명합니다.

구문 요청

```
{
   "AgreementId": "string",
   "ServerId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

AgreementId

계약의 고유 식별자입니다. 계약을 생성하면 이 식별자가 반환됩니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: a-([0-9a-f]{17})

필수 여부: 예

ServerId

계약과 연결된 서버 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: s-([0-9a-f]{17})

필수 항목 여부: 예

응답 구문

```
{
```

```
   "Agreement": {
      "AccessRole": "string",
      "AgreementId": "string",
      "Arn": "string",
      "BaseDirectory": "string",
      "Description": "string",
      "LocalProfileId": "string",
      "PartnerProfileId": "string",
      "ServerId": "string",
      "Status": "string",
      "Tags": [
         {
            "Key": "string",
            "Value": "string"
         }
      ]
   }
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

Agreement

지정된 계약의 세부 정보로, `DescribedAgreement` 객체로 반환됩니다.

타입: DescribedAgreement 객체

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

DescribeCertificate

CertificateId로 식별되는 인증서를 설명합니다.

구문 요청

```
{
   "CertificateId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

CertificateId

가져온 인증서의 식별자 배열입니다. 프로필 및 파트너 프로필 작업에 이 식별자를 사용합니다.

타입: 문자열

길이 제약 조건: 고정 길이는 22입니다.

패턴: cert-([0-9a-f]{17})

필수 항목 여부: 예

응답 구문

```
{
   "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"
   }
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

Certificate

지정된 인증서의 세부 정보로, 객체로 반환됩니다.

타입: DescribedCertificate 객체

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

DescribeConnector

ConnectorId.로 식별되는 커넥터를 설명합니다.

구문 요청

```
{
   "ConnectorId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

ConnectorId

커넥터의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: c-([0-9a-f]{17})

필수 항목 여부: 예

응답 구문

```
{
   "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"
   }
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

Connector

커넥터의 세부 정보를 포함하는 구조입니다.

타입: DescribedConnector 객체

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

DescribeExecution

DescribeExecution를 사용하여 지정된 워크플로의 실행 세부 정보를 확인할 수 있습니다.

> **Note**
>
> 이 API 호출은 진행 중인 워크플로의 세부 정보만 반환합니다.
> 진행 중이 아닌 실행의 ID를 제공하거나 실행이 지정된 워크플로 ID와 일치하지 않는 경우 ResourceNotFound 예외가 발생합니다.

구문 요청

```
{
   "ExecutionId": "string",
   "WorkflowId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

ExecutionId

워크플로 실행의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 36입니다.

패턴: [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}

필수 여부: 예

WorkflowId

워크플로의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `w-([a-z0-9]{17})`

필수 항목 여부: 예

응답 구문

```
{
   "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"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

Execution

워크플로 예외의 세부 정보를 포함하는 구조입니다.

타입: DescribedExecution 객체

WorkflowId

워크플로의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `w-([a-z0-9]{17})`

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

DescribeHostKey

HostKeyId 및 ServerId에 의해 지정된 호스트 키의 세부 정보를 반환합니다.

구문 요청

```
{
   "HostKeyId": "string",
   "ServerId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

HostKeyId

설명하려는 호스트 키의 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 25입니다.

패턴: hostkey-[0-9a-f]{17}

필수 사항 여부: Yes

ServerId

설명하려는 호스트 키가 들어 있한 서버의 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: s-([0-9a-f]{17})

필수 항목 여부: 예

응답 구문

```
{
```

```
   "HostKey": {
      "Arn": "string",
      "DateImported": number,
      "Description": "string",
      "HostKeyFingerprint": "string",
      "HostKeyId": "string",
      "Tags": [
         {
            "Key": "string",
            "Value": "string"
         }
      ],
      "Type": "string"
   }
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

HostKey

지정된 호스트 키의 세부 정보를 반환합니다.

타입: DescribedHostKey 객체

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

DescribeProfile

ProfileId에 의해 지정된 프로필의 세부 정보를 반환합니다.

구문 요청

```
{
   "ProfileId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

ProfileId

설명하려는 프로필의 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: p-([0-9a-f]{17})

필수 항목 여부: 예

응답 구문

```
{
   "Profile": {
      "Arn": "string",
      "As2Id": "string",
      "CertificateIds": [ "string" ],
      "ProfileId": "string",
      "ProfileType": "string",
      "Tags": [
         {
            "Key": "string",
            "Value": "string"
```

```
        }
      ]
    }
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

Profile

지정된 프로필의 세부 정보로, 객체로 반환됩니다.

타입: DescribedProfile 객체

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

DescribeSecurityPolicy

서버 또는 SFTP 커넥터에 연결된 보안 정책을 설명합니다. 응답에는 보안 정책 속성에 대한 설명이 포함됩니다. 보안 정책에 대한 자세한 내용은 서버의 보안 정책 사용 또는 SFTP 커넥터의 보안 정책 사용을 참조하십시오.

구문 요청

```
{
   "SecurityPolicyName": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

SecurityPolicyName

세부 정보를 보려는 보안 정책의 텍스트 이름을 지정하십시오.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 100입니다.

패턴: `Transfer[A-Za-z0-9]*SecurityPolicy-[A-Za-z0-9-]+`

필수 항목 여부: 예

응답 구문

```
{
   "SecurityPolicy": {
      "Fips": boolean,
      "Protocols": [ "string" ],
      "SecurityPolicyName": "string",
      "SshCiphers": [ "string" ],
      "SshHostKeyAlgorithms": [ "string" ],
      "SshKexs": [ "string" ],
      "SshMacs": [ "string" ],
```

```
        "TlsCiphers": [ "string" ],
        "Type": "string"
    }
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

SecurityPolicy

보안 정책의 속성을 포함하는 배열.

타입: DescribedSecurityPolicy 객체

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

AWS Transfer Family 서비스에 오류가 발생하면 이 예외가 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

예

예

다음 예 명령은 보안 정책 이름을 인수로 사용하고 지정된 보안 정책에 대한 알고리즘을 반환합니다.

샘플 요청

```
aws transfer describe-security-policy --security-policy-name "TransferSecurityPolicy-
FIPS-2023-05"
```

샘플 응답

```
{
    "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"
        ]
    }
```

```
}
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

DescribeServer

ServerId 파라미터를 전달하여 지정하는 파일 전송 프로토콜 지원 서버를 설명합니다.

응답에는 서버 속성에 대한 설명이 포함됩니다. EndpointType(을)를 VPC로 설정하면 응답에 EndpointDetails(이)가 포함됩니다.

구문 요청

```
{
   "ServerId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

ServerId

서버에 대해 시스템에서 할당한 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: s-([0-9a-f]{17})

필수 항목 여부: 예

응답 구문

```
{
   "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"
            }
         ]
      }
   }
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

Server

지정한 ServerID(이)가 있는 서버의 속성을 포함하는 배열입니다.

타입: DescribedServer 객체

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

예

예

다음은 서버에 할당된 속성을 반환하는 예입니다.

샘플 요청

```
{
   "ServerId": "s-01234567890abcdef"
}
```

예

이 예는 의 한 가지 사용법을 보여줍니다 DescribeServer.

샘플 응답

```
{
    "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
    }
}
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS 파이썬용 SDK
- AWS 루비 V3용 SDK

DescribeUser

ServerId 속성으로 식별되는 특정 File Transfer 프로토콜 지원 서버에 할당된 사용자를 설명합니다.

이 직접적인 호출의 응답은 지정된 ServerId 값과 관련된 사용자의 속성을 반환합니다.

구문 요청

```
{
   "ServerId": "string",
   "UserName": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

ServerId

이 사용자가 할당한 서버에 대해 시스템에서 할당한 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: s-([0-9a-f]{17})

필수 사항 여부: Yes

UserName

하나 이상의 서버에 할당된 사용자 이름. 사용자 이름은 AWS Transfer Family 서비스를 사용하고 파일 전송 작업을 수행하기 위한 로그인 자격 증명의 일부입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 3입니다. 최대 길이는 100입니다.

패턴: [\w][\w@.-]{2,99}

필수 항목 여부: 예

응답 구문

```
{
   "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"
   }
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

ServerId

이 사용자가 할당한 서버에 대해 시스템에서 할당한 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

User

지정한 `ServerID` 값에 대한 Transfer Family 사용자의 속성이 들어 있는 배열입니다.

타입: DescribedUser 객체

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

서비스에 오류가 발생하면 이 예외가 AWS Transfer Family 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

예

예

다음 예에서는 기존 사용자의 세부 정보를 보여줍니다.

샘플 요청

```
aws transfer describe-user --server-id s-1111aaaa2222bbbb3 --user-name bob-test
```

샘플 응답

```
{
    "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"
    }
}
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK

- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

DescribeWorkflow

지정된 워크플로를 설명합니다.

구문 요청

```
{
   "WorkflowId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

WorkflowId

워크플로의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `w-([a-z0-9]{17})`

필수 항목 여부: 예

응답 구문

```
{
   "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"
   }
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

Workflow

워크플로의 세부 정보를 포함하는 구조입니다.

타입: DescribedWorkflow 객체

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

ImportCertificate

로컬(AS2) 프로필과 파트너 프로필을 생성하는 데 필요한 서명 및 암호화 인증서를 가져옵니다.

구문 요청

```
{
   "ActiveDate": number,
   "Certificate": "string",
   "CertificateChain": "string",
   "Description": "string",
   "InactiveDate": number,
   "PrivateKey": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ],
   "Usage": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

ActiveDate

인증서가 활성화되는 시기를 지정하는 선택적 날짜입니다.

타입: Timestamp

필수 여부: 아니요

Certificate

- CLI의 경우 인증서의 파일 경로를 URI 형식으로 제공하세요. 예를 들어 `--certificate file://encryption-cert.pem`입니다. 또는 원시 콘텐츠를 제공할 수 있습니다.
- SDK의 경우 인증서 파일의 원시 콘텐츠를 지정합니다. 예: `--certificate "`cat encryption-cert.pem`"`.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 16384입니다.

패턴: `[\u0009\u000A\u000D\u0020-\u00FF]*`

필수 사항 여부: Yes

CertificateChain

가져오는 인증서의 체인을 구성하는 선택적 인증서 목록.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 2097152입니다.

패턴: `[\u0009\u000A\u000D\u0020-\u00FF]*`

Required: No

Description

인증서를 식별하는 데 도움이 되는 간단한 설명.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 200입니다.

패턴: `[\p{Graph}]+`

필수 여부: 아니요

InactiveDate

인증서가 비활성화되는 시기를 지정하는 선택적 날짜.

타입: Timestamp

필수 여부: 아니요

PrivateKey

- CLI의 경우 프라이빗 키의 파일 경로를 URI 형식으로 제공합니다. 예: `--private-key file://encryption-key.pem`. 또는 프라이빗 키 파일의 원시 콘텐츠를 제공할 수도 있습니다.
- SDK의 경우 프라이빗 키 파일의 원시 콘텐츠를 지정합니다. 예제: `--private-key "`cat encryption-key.pem`"`

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 16384입니다.

패턴: `[\u0009\u000A\u000D\u0020-\u00FF]*`

Required: No

Tags

인증서를 그룹화하고 검색하는 데 사용할 수 있는 키-값 쌍입니다.

타입: Tag 객체 배열

배열 멤버: 최소 항목 수는 1개입니다. 최대 항목 수는 50개입니다.

필수 여부: 아니요

Usage

이 인증서를 사용하는 방법을 지정합니다. 다음과 같은 방법으로 사용할 수 있습니다.

- `SIGNING`: AS2 메시지 서명용
- `ENCRYPTION`: AS2 메시지 암호화용
- `TLS`: HTTPS를 통해 전송되는 AS2 통신을 보호하는 데 사용됩니다.

타입: 문자열

유효 값: `SIGNING | ENCRYPTION`

필수 여부: 예

응답 구문

```
{
   "CertificateId": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

CertificateId

가져온 인증서의 식별자 배열입니다. 프로필 및 파트너 프로필 작업에 이 식별자를 사용합니다.

타입: 문자열

길이 제약 조건: 고정 길이는 22입니다.

패턴: `cert-([0-9a-f]{17})`

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

서비스에 오류가 발생하면 이 예외가 발생합니다. AWS Transfer Family

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

예

예

다음 예에서는 암호화에 사용할 인증서를 가져옵니다. 첫 번째 명령에서는 인증서 및 인증서 체인 파일의 내용을 제공합니다. SDK 명령에 이 형식을 사용합니다.

```
aws transfer import-certificate --usage ENCRYPTION --certificate "`cat encryption-
cert.pem`" \
    --private-key "`cat encryption-key.pem`" --certificate-chain "`cat root-ca.pem`"
```

예

다음 예는 프라이빗 키, 인증서 및 인증서 체인 파일의 파일 위치를 제공한다는 점을 제외하면 이전 명령과 동일합니다. SDK를 사용하는 경우 이 버전의 명령은 작동하지 않습니다.

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

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS 파이썬용 SDK
- AWS 루비 V3용 SDK

ImportHostKey

`ServerId` 파라미터로 지정된 서버에 호스트 키를 추가합니다.

구문 요청

```
{
   "Description": "string",
   "HostKeyBody": "string",
   "ServerId": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

Description

이 호스트 키를 식별하는 텍스트 설명.

타입: 문자열

길이 제한: 최소 길이는 0. 최대 길이는 200입니다.

패턴: `[\p{Print}]*`

Required: No

HostKeyBody

SSH 키 쌍의 프라이빗 키 부분.

AWS Transfer Family RSA, ECDSA 및 ED25519 키를 사용할 수 있습니다.

타입: 문자열

길이 제약: 최소 길이는 0입니다. 최대 길이는 4096입니다.

필수 여부: 예

ServerId

가져오려는 호스트 키가 들어 있는 서버의 식별자.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

필수 사항 여부: Yes

Tags

그룹화 및 호스트 키 검색에 사용될 수 있는 키-값 쌍입니다.

타입: Tag 객체 배열

배열 멤버: 최소 항목 수는 1개입니다. 최대 항목 수는 50개입니다.

필수 여부: 아니요

응답 구문

```
{
   "HostKeyId": "string",
   "ServerId": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

HostKeyId

가져온 키의 호스트 키 식별자를 반환합니다.

타입: 문자열

길이 제약 조건: 고정 길이는 25입니다.

패턴: `hostkey-[0-9a-f]{17}`

ServerId

가져온 키가 포함된 서버 식별자를 반환합니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

서비스에 오류가 발생하면 이 예외가 발생합니다. AWS Transfer Family

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceExistsException

요청된 리소스가 존재하지 않거나 명령에 지정된 리전이 아닌 다른 리전에 있습니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

ThrottlingException

요청 제한 때문에 요청이 거부되었습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

ImportSshPublicKey

(ServerId로 식별되는) 특정 파일 전송 프로토콜 지원 서버에 할당된 UserName 값으로 식별되는 Transfer Family 사용자에게 SSH (Secure Shell) 퍼블릭 키를 추가합니다.

응답은 UserName 값, ServerId 값 및 SshPublicKeyId의 명칭을 반환합니다.

구문 요청

```
{
   "ServerId": "string",
   "SshPublicKeyBody": "string",
   "UserName": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

ServerId

서버에 대해 시스템에서 할당한 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: s-([0-9a-f]{17})

필수 여부: 예

SshPublicKeyBody

SSH 키 쌍의 퍼블릭 키 부분.

AWS Transfer Family RSA, ECDSA 및 ED25519 키를 사용할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 2048입니다.

필수 여부: 예

UserName

하나 이상의 서버에 할당된 Transfer Family 사용자의 명칭.

타입: 문자열

길이 제약 조건: 최소 길이는 3입니다. 최대 길이는 100입니다.

패턴: `[\w][\w@.-]{2,99}`

필수 항목 여부: 예

응답 구문

```
{
   "ServerId": "string",
   "SshPublicKeyId": "string",
   "UserName": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

ServerId

서버에 대해 시스템에서 할당한 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

SshPublicKeyId

가져온 시스템에서 퍼블릭 키에 부여한 명칭.

타입: 문자열

길이 제약 조건: 고정 길이는 21입니다.

패턴: `key-[0-9a-f]{17}`

UserName

지정한 `ServerID` 값에 할당된 사용자 이름입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 3입니다. 최대 길이는 100입니다.

패턴: `[\w][\w@.-]{2,99}`

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

서비스에 오류가 발생하면 이 예외가 발생합니다. AWS Transfer Family

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceExistsException

요청된 리소스가 존재하지 않거나 명령에 지정된 리전이 아닌 다른 리전에 있습니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

ThrottlingException

요청 제한 때문에 요청이 거부되었습니다.

HTTP 상태 코드: 400

예

예

이 명령은 `id_ecdsa.pub` 파일에 저장된 ECDSA 키를 가져옵니다.

```
aws transfer import-ssh-public-key --server-id s-021345abcdef6789 --ssh-public-key-body
 file://id_ecdsa.pub --user-name jane-doe
```

예

명령이 제대로 실행되지 않으면, 시스템에서 다음과 비슷한 정보를 반환합니다.

```
{
    "ServerId": "s-021345abcdef6789",
   "SshPublicKeyId": "key-1234567890abcdef0",
   "UserName": "jane-doe"
}
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK

- AWS 루비 V3용 SDK

ListAccesses

서버에 있는 모든 액세스 권한의 세부 정보를 나열합니다.

구문 요청

```
{
   "MaxResults": number,
   "NextToken": "string",
   "ServerId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

MaxResults

반환할 액세스 SID의 최대수를 지정합니다.

타입: 정수

유효한 범위: 최소값은 1입니다. 최대값은 1000입니다.

필수 여부: 아니요

NextToken

ListAccesses 호출에서 추가 결과를 얻을 수 있는 경우 출력에 NextToken 파라미터가 반환됩니다. 그런 다음 NextToken 파라미터에 후속 명령을 전달하여 추가 액세스를 계속 나열할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 6144입니다.

필수 여부: 아니요

ServerId

사용자가 할당된 서버에 대해 시스템에서 할당한 고유 식별자.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

필수 항목 여부: 예

응답 구문

```
{
   "Accesses": [
      {
         "ExternalId": "string",
         "HomeDirectory": "string",
         "HomeDirectoryType": "string",
         "Role": "string"
      }
   ],
   "NextToken": "string",
   "ServerId": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

Accesses

지정한 `ServerId` 값에 대한 액세스 및 해당 속성을 반환합니다.

타입: ListedAccess 객체 배열

NextToken

`ListAccesses` 호출에서 추가 결과를 얻을 수 있는 경우 출력에 `NextToken` 파라미터가 반환됩니다. 그런 다음 `NextToken` 파라미터에 후속 명령을 전달하여 추가 액세스를 계속 나열할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 6144입니다.

ServerId

사용자가 할당된 서버에 대해 시스템에서 할당한 고유 식별자.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidNextTokenException

전달된 `NextToken` 파라미터가 유효하지 않습니다.

HTTP 상태 코드: 400

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

ListAgreements

ServerId이(가) 제공한 계약으로 식별되는 서버의 계약 목록을 반환합니다. 결과를 특정 수로 제한하려면 MaxResults 파라미터 값을 제공하세요. 이전에 명령을 실행하여 NextToken에 대한 값을 받은 경우 해당 값을 제공하여 중단한 부분부터 계약을 계속 나열할 수 있습니다.

구문 요청

```
{
   "MaxResults": number,
   "NextToken": "string",
   "ServerId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

MaxResults

반환할 최대 계약 수입니다.

타입: 정수

유효한 범위: 최소값은 1입니다. 최대값은 1000입니다.

필수 여부: 아니요

NextToken

ListAgreements 호출에서 추가 결과를 얻을 수 있는 경우, 출력에 NextToken 파라미터가 반환됩니다. 그런 다음 NextToken 파라미터에 후속 명령을 전달하여 추가 계약을 계속 나열할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 6144입니다.

필수 여부: 아니요

ServerId

계약 목록을 작성하려는 서버의 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

필수 항목 여부: 예

응답 구문

```
{
   "Agreements": [
      {
         "AgreementId": "string",
         "Arn": "string",
         "Description": "string",
         "LocalProfileId": "string",
         "PartnerProfileId": "string",
         "ServerId": "string",
         "Status": "string"
      }
   ],
   "NextToken": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

Agreements

각 항목에 계약 세부 정보가 포함된 배열을 반환합니다.

타입: ListedAgreement 객체 배열

NextToken

`ListAgreements`를 다시 호출하여 추가 결과(있는 경우)를 수신하는 데 사용할 수 있는 토큰을 반환합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 6144입니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidNextTokenException

전달된 `NextToken` 파라미터가 유효하지 않습니다.

HTTP 상태 코드: 400

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++

- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

ListCertificates

로 가져온 현재 인증서 목록을 반환합니다 AWS Transfer Family. 결과를 특정 수로 제한하려면 `MaxResults` 파라미터 값을 제공하세요. 이전에 명령을 실행하여 `NextToken` 파라미터 값을 받은 경우 해당 값을 제공하여 중단한 부분부터 인증서를 계속 나열할 수 있습니다.

구문 요청

```
{
   "MaxResults": number,
   "NextToken": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

MaxResults

반환될 최대 인증서 수입니다.

타입: 정수

유효한 범위: 최소값은 1입니다. 최대값은 1000입니다.

필수 여부: 아니요

NextToken

`ListCertificates` 호출에서 추가 결과를 얻을 수 있는 경우 출력에 `NextToken` 파라미터가 반환됩니다. 그런 다음 `NextToken` 파라미터에 후속 명령을 전달하여 추가 인증서를 계속 나열할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 6144입니다.

필수 여부: 아니요

응답 구문

```
{
   "Certificates": [
      {
         "ActiveDate": number,
         "Arn": "string",
         "CertificateId": "string",
         "Description": "string",
         "InactiveDate": number,
         "Status": "string",
         "Type": "string",
         "Usage": "string"
      }
   ],
   "NextToken": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

Certificates

ListCertificates 호출에 지정된 인증서 배열을 반환합니다.

타입: ListedCertificate 객체 배열

NextToken

다음 인증서를 나열하는 데 사용할 수 있는 다음 토큰을 반환합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 6144입니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

AWS Transfer Family 서비스에 오류가 발생하면 이 예외가 발생합니다.

HTTP 상태 코드: 500

InvalidNextTokenException

전달된 `NextToken` 파라미터가 유효하지 않습니다.

HTTP 상태 코드: 400

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK

- AWS 루비 V3용 SDK

ListConnectors

지정된 지역의 커넥터를 나열합니다.

구문 요청

```
{
   "MaxResults": number,
   "NextToken": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

MaxResults

반환될 최대 커넥터 수입니다.

타입: 정수

유효한 범위: 최소값은 1입니다. 최대값은 1000입니다.

필수 여부: 아니요

NextToken

ListConnectors 호출에서 추가 결과를 얻을 수 있는 경우 출력에 NextToken 파라미터가 반환됩니다. 그런 다음 NextToken 파라미터에 후속 명령을 전달하여 추가 커넥터를 계속 나열할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 6144입니다.

필수 여부: 아니요

응답 구문

```
{
```

```
   "Connectors": [
      {
         "Arn": "string",
         "ConnectorId": "string",
         "Url": "string"
      }
   ],
   "NextToken": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

Connectors

각 항목에 커넥터의 세부 정보가 포함된 배열을 반환합니다.

타입: ListedConnector 객체 배열

NextToken

ListConnectors를 다시 호출하여 추가 결과(있는 경우)를 수신하는 데 사용할 수 있는 토큰을 반환합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 6144입니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidNextTokenException

전달된 NextToken 파라미터가 유효하지 않습니다.

HTTP 상태 코드: 400

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

ListExecutions

지정된 워크플로에 대해 진행 중인 모든 실행을 나열합니다.

> **Note**
>
> 지정된 워크플로 ID를 찾을 수 없는 경우 `ListExecutions`은(는) `ResourceNotFound` 예외를 반환합니다.

구문 요청

```
{
   "MaxResults": number,
   "NextToken": "string",
   "WorkflowId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

MaxResults

반환할 최대 집행 수를 지정합니다.

타입: 정수

유효한 범위: 최소값은 1입니다. 최대값은 1000입니다.

필수 여부: 아니요

NextToken

`ListExecutions`은(는) 출력에 `NextToken` 파라미터를 반환합니다. 그런 다음 후속 명령에 `NextToken` 파라미터를 전달하여 추가 실행을 계속 나열할 수 있습니다.

예를 들어, 이는 페이지 매김에 유용합니다. 워크플로 실행 횟수가 100개인 경우 처음 10개만 나열하는 것이 좋습니다. 그렇다면 `max-results`을 지정하여 API를 호출하세요:

```
aws transfer list-executions --max-results 10
```

그러면 처음 10개 집행에 대한 세부 정보와 11번째 실행에 대한 포인터(NextToken)가 반환됩니다. 이제 수신한 NextToken 값을 입력하여 API를 다시 호출할 수 있습니다:

```
aws transfer list-executions --max-results 10 --next-token
$somePointerReturnedFromPreviousListResult
```

이 호출은 다음 10번의 집행(11번째부터 20번째)을 반환합니다. 그런 다음 100개 실행에 대한 세부 정보가 모두 반환될 때까지 호출을 반복할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 6144입니다.

필수 여부: 아니요

WorkflowId

워크플로의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: w-([a-z0-9]{17})

필수 항목 여부: 예

응답 구문

```
{
   "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"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

Executions

각 실행의 세부 정보를 `ListedExecution` 배열로 반환합니다.

타입: ListedExecution 객체 배열

NextToken

`ListExecutions`은(는) 출력에 `NextToken` 파라미터를 반환합니다. 그런 다음 후속 명령에 `NextToken` 파라미터를 전달하여 추가 실행을 계속 나열할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 6144입니다.

WorkflowId

워크플로의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `w-([a-z0-9]{17})`

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidNextTokenException

전달된 `NextToken` 파라미터가 유효하지 않습니다.

HTTP 상태 코드: 400

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET

- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

ListHostKeys

파라미터로 지정된 서버의 호스트 키 목록을 반환합니다. `ServerId`

구문 요청

```
{
   "MaxResults": number,
   "NextToken": "string",
   "ServerId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

MaxResults

반환할 최대 호스트 키 수입니다.

타입: 정수

유효한 범위: 최소값은 1입니다. 최대값은 1000입니다.

필수 여부: 아니요

NextToken

반환되지 않은 추가 결과가 있는 경우 `NextToken` 파라미터가 반환됩니다. 이 값을 `ListHostKeys`에 대한 후속 호출에 사용하여 결과를 계속 나열할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 6144입니다.

필수 여부: 아니요

ServerId

확인할 호스트 키가 포함된 서버의 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

필수 항목 여부: 예

응답 구문

```
{
   "HostKeys": [
      {
         "Arn": "string",
         "DateImported": number,
         "Description": "string",
         "Fingerprint": "string",
         "HostKeyId": "string",
         "Type": "string"
      }
   ],
   "NextToken": "string",
   "ServerId": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

HostKeys

각 항목에 호스트 키의 세부 정보가 포함된 배열을 반환합니다.

타입: ListedHostKey 객체 배열

NextToken

`ListHostKeys`를 다시 호출하여 추가 결과(있는 경우)를 수신하는 데 사용할 수 있는 토큰을 반환합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 6144입니다.

ServerId

나열된 호스트 키가 포함된 서버 식별자를 반환합니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidNextTokenException

전달된 `NextToken` 파라미터가 유효하지 않습니다.

HTTP 상태 코드: 400

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

ListProfiles

시스템의 프로필 목록을 반환합니다. 결과를 특정 수로 제한하려면 `MaxResults` 파라미터 값을 제공하세요. 이전에 명령을 실행하여 `NextToken`에 대한 값을 받은 경우, 해당 값을 제공하여 중단한 부분부터 프로필을 계속 나열할 수 있습니다.

구문 요청

```
{
   "MaxResults": number,
   "NextToken": "string",
   "ProfileType": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

MaxResults

반환할 최대 프로필 수입니다.

타입: 정수

유효한 범위: 최소값은 1입니다. 최대값은 1000입니다.

필수 여부: 아니요

NextToken

반환되지 않은 추가 결과가 있는 경우 `NextToken` 파라미터가 반환됩니다. 이 값을 `ListProfiles`에 대한 후속 호출에 사용하여 결과를 계속 나열할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 6144입니다.

필수 여부: 아니요

ProfileType

`LOCAL` 타입 프로필만 나열할지 아니면 `PARTNER` 타입 프로필만 나열할지를 나타냅니다. 요청에 입력되지 않은 경우 명령은 모든 타입의 프로필을 나열합니다.

타입: 문자열

유효 값: `LOCAL | PARTNER`

필수 항목 여부: 아니요

응답 구문

```
{
   "NextToken": "string",
   "Profiles": [
      {
         "Arn": "string",
         "As2Id": "string",
         "ProfileId": "string",
         "ProfileType": "string"
      }
   ]
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

NextToken

다시 `ListProfiles`를 호출하여 추가 결과(있는 경우)를 수신하는 데 사용할 수 있는 토큰을 반환합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 6144입니다.

Profiles

각 항목에 프로필의 세부 정보가 포함된 배열을 반환합니다.

타입: ListedProfile 객체 배열

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidNextTokenException

전달된 `NextToken` 파라미터가 유효하지 않습니다.

HTTP 상태 코드: 400

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK

- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

ListSecurityPolicies

서버 및 SFTP 커넥터에 연결된 보안 정책을 나열합니다. 보안 정책에 대한 자세한 내용은 서버의 보안 정책 사용 또는 SFTP 커넥터의 보안 정책 사용을 참조하십시오.

구문 요청

```
{
   "MaxResults": number,
   "NextToken": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

MaxResults

ListSecurityPolicies 쿼리에 대한 응답으로 반환할 보안 정책 수를 지정합니다.

타입: 정수

유효한 범위: 최소값은 1입니다. 최대값은 1000입니다.

필수 여부: 아니요

NextToken

ListSecurityPolicies 명령에서 추가 결과를 얻으면 출력에 NextToken 파라미터가 반환됩니다. 그런 다음 후속 명령에 NextToken 파라미터를 전달하여 추가 보안 정책을 계속 나열할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 6144입니다.

필수 여부: 아니요

응답 구문

```
{
```

```
   "NextToken": "string",
   "SecurityPolicyNames": [ "string" ]
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

NextToken

ListSecurityPolicies 작업에서 추가 결과를 얻을 수 있는 경우 출력에 NextToken 파라미터가 반환됩니다. 다음 명령에서 NextToken 파라미터를 전달하여 보안 정책을 계속 나열할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 6144입니다.

SecurityPolicyNames

나열된 보안 정책의 배열.

타입: 문자열 배열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 100입니다.

패턴: Transfer[A-Za-z0-9]*SecurityPolicy-[A-Za-z0-9-]+

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 서비스에 오류가 발생할 때 발생합니다. AWS Transfer Family

HTTP 상태 코드: 500

InvalidNextTokenException

전달된 NextToken 파라미터가 유효하지 않습니다.

HTTP 상태 코드: 400

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

예

예

다음 예에는 사용 가능한 모든 보안 정책의 이름이 전부 나열되어 있습니다.

샘플 요청

```
aws transfer list-security-policies
```

샘플 응답

```
{
    "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"
    ]
}
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

ListServers

AWS 계정과 연결된 파일 전송 프로토콜 지원 서버를 나열합니다.

구문 요청

```
{
   "MaxResults": number,
   "NextToken": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

MaxResults

ListServers 쿼리에 대한 응답으로 반환할 서버 수를 지정합니다.

타입: 정수

유효한 범위: 최소값은 1입니다. 최대값은 1000입니다.

필수 여부: 아니요

NextToken

ListServers 명령에서 추가 결과를 얻으면 출력에 NextToken 파라미터가 반환됩니다. 그런 다음 후속 명령에 NextToken 파라미터를 전달하여 추가 서버를 계속 나열할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 6144입니다.

필수 여부: 아니요

응답 구문

```
{
```

```
   "NextToken": "string",
   "Servers": [
      {
         "Arn": "string",
         "Domain": "string",
         "EndpointType": "string",
         "IdentityProviderType": "string",
         "LoggingRole": "string",
         "ServerId": "string",
         "State": "string",
         "UserCount": number
      }
   ]
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

NextToken

ListServers 작업에서 추가 결과를 얻을 수 있는 경우 출력에 NextToken 파라미터가 반환됩니다. 다음 명령에서 NextToken 파라미터를 전달하여 추가 서버를 계속 나열할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 6144입니다.

Servers

나열된 서버 배열입니다.

타입: ListedServer 객체 배열

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidNextTokenException

전달된 NextToken 파라미터가 유효하지 않습니다.

HTTP 상태 코드: 400

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

예

예

다음 예에는 에 있는 서버의 목록이 나와 있습니다 AWS 계정.

참고로, 예의 NextToken 값은 실제 값이 아닙니다. 이 값은 파라미터 사용 방법을 나타내기 위한 것입니다.

샘플 요청

```
{
   "MaxResults": 1,
   "NextToken": "token-from-previous-API-call"
}
```

샘플 응답

```
{
   "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
        }
    ]
}
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

ListTagsForResource

지정한 Amazon 리소스 이름(ARN)과 연결된 태그를 모두 나열합니다. 리소스는 사용자, 서버 또는 역할입니다.

구문 요청

```
{
   "Arn": "string",
   "MaxResults": number,
   "NextToken": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

Arn

특정 Amazon 리소스 이름(ARN)과 연결된 태그를 요청합니다. ARN은 서버, 사용자 또는 역할과 같은 특정 AWS 리소스의 식별자입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 20입니다. 최대 길이는 1600입니다.

패턴: `arn:\S+`

필수 사항 여부: Yes

MaxResults

`ListTagsForResource` 요청에 대한 응답으로 반환할 태그 수를 지정합니다.

타입: 정수

유효한 범위: 최소값은 1입니다. 최대값은 1000입니다.

필수 여부: 아니요

NextToken

ListTagsForResource 작업의 추가 결과를 요청하면 입력에 NextToken 파라미터가 반환됩니다. 그런 다음 후속 명령에 NextToken 파라미터를 전달하여 추가 태그를 계속 나열할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 6144입니다.

필수 여부: 아니요

응답 구문

```
{
   "Arn": "string",
   "NextToken": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

Arn

태그를 나열하기 위해 지정한 ARN입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 20입니다. 최대 길이는 1600입니다.

패턴: arn:\S+

NextToken

ListTagsForResource 호출에서 추가 결과를 얻을 수 있는 경우 출력에 NextToken 파라미터가 반환됩니다. 그런 다음 후속 명령에 NextToken 파라미터를 전달하여 추가 태그를 계속 나열할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 6144입니다.

Tags

일반적으로 항목을 그룹화하고 검색하기 위해 리소스에 할당되는 키-값 쌍입니다. 태그는 사용자가 정의하는 메타데이터입니다.

타입: Tag 객체 배열

배열 멤버: 최소 항목 수는 1개입니다. 최대 항목 수는 50개입니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

서비스에 오류가 발생하면 이 예외가 AWS Transfer Family 발생합니다.

HTTP 상태 코드: 500

InvalidNextTokenException

전달된 NextToken 파라미터가 유효하지 않습니다.

HTTP 상태 코드: 400

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

예

예

다음 예는 지정한 ARN이 포함된 리소스의 태그를 나열합니다.

샘플 요청

```
{
   "Arn": "arn:aws:transfer:us-east-1:176354371281:server/s-01234567890abcdef"
}
```

예

이 예에서는 의 한 가지 사용법을 보여줍니다 ListTagsForResource.

샘플 응답

```
{
   "Tags": [
      {
         "Key": "Name",
         "Value": "MyServer"
      }
   ]
}
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK

- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

ListUsers

ServerId 파라미터를 전달하여 지정한 파일 전송 프로토콜 지원 서버의 사용자를 나열합니다.

구문 요청

```
{
   "MaxResults": number,
   "NextToken": "string",
   "ServerId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

MaxResults

ListUsers 요청에 대한 응답으로 반환할 사용자 수를 지정합니다.

타입: 정수

유효한 범위: 최소값은 1입니다. 최대값은 1000입니다.

필수 여부: 아니요

NextToken

ListUsers 호출의 추가 결과가 있는 경우 출력에 NextToken 파라미터가 반환됩니다. 그런 다음 NextToken을 후속 ListUsers 명령에 전달하여 추가 사용자를 계속 나열할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 6144입니다.

필수 여부: 아니요

ServerId

사용자가 할당된 서버에 대해 시스템에서 할당한 고유 식별자.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

필수 항목 여부: 예

응답 구문

```
{
   "NextToken": "string",
   "ServerId": "string",
   "Users": [
      {
         "Arn": "string",
         "HomeDirectory": "string",
         "HomeDirectoryType": "string",
         "Role": "string",
         "SshPublicKeyCount": number,
         "UserName": "string"
      }
   ]
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

NextToken

`ListUsers` 호출에서 추가 결과를 얻을 수 있는 경우 출력에 `NextToken` 파라미터가 반환됩니다. 그런 다음 후속 명령에 `NextToken` 파라미터를 전달하여 추가 사용자를 계속 나열할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 6144입니다.

ServerId

사용자가 할당된 서버에 대해 시스템에서 할당한 고유 식별자.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

Users

지정한 `ServerId` 값에 대한 Transfer Family 사용자 및 해당 속성을 반환합니다.

타입: ListedUser 객체 배열

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidNextTokenException

전달된 `NextToken` 파라미터가 유효하지 않습니다.

HTTP 상태 코드: 400

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

예

예

ListUsers API 호출은 지정한 서버와 관련된 사용자 목록을 반환합니다.

샘플 요청

```
        {
         "MaxResults": 100,
         "NextToken": "eyJNYXJrZXIiOiBudWxsLCAiYm90b1X0cnVuU2F0ZV9hbW91bnQiOiAyfQ==",
         "ServerId": "s-01234567890abcdef"
         }
```

예

다음은 이 API 직접 호출에 대한 샘플 응답입니다.

샘플 응답

```
{
   "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"
      }
   ]
}
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

ListWorkflows

현재 지역의 해당 지역과 관련된 모든 AWS 계정 워크플로를 나열합니다.

구문 요청

```
{
   "MaxResults": number,
   "NextToken": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

MaxResults

반환할 최대 워크플로 수를 지정합니다.

타입: 정수

유효한 범위: 최소값은 1입니다. 최대값은 1000입니다.

필수 여부: 아니요

NextToken

ListWorkflows은(는) 출력에 NextToken 파라미터를 반환합니다. 그런 다음 후속 명령에 NextToken 파라미터를 전달하여 추가 워크플로를 계속 나열할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 6144입니다.

필수 여부: 아니요

응답 구문

```
{
   "NextToken": "string",
```

```
   "Workflows": [
      {
         "Arn": "string",
         "Description": "string",
         "WorkflowId": "string"
      }
   ]
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

NextToken

ListWorkflows은(는) 출력에 NextToken 파라미터를 반환합니다. 그런 다음 후속 명령에 NextToken 파라미터를 전달하여 추가 워크플로를 계속 나열할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 6144입니다.

Workflows

각 워크플로에 대해 Arn, WorkflowId 및 Description를 반환합니다.

타입: ListedWorkflow 객체 배열

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidNextTokenException

전달된 NextToken 파라미터가 유효하지 않습니다.

HTTP 상태 코드: 400

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

SendWorkflowStepState

비동기 사용자 지정 단계에 대한 콜백을 전송할 수 있습니다.

`ExecutionId`, `WorkflowId`, 및 `Token` 는 워크플로의 사용자 지정 단계를 실행하는 동안 대상 리소스에 전달됩니다. 상태를 제공할 뿐만 아니라 콜백에도 이러한 항목을 포함해야 합니다.

구문 요청

```
{
   "ExecutionId": "string",
   "Status": "string",
   "Token": "string",
   "WorkflowId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

ExecutionId

워크플로 실행의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 36입니다.

패턴: `[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}`

필수 사항 여부: Yes

Status

지정된 단계의 성공 또는 실패 여부를 나타냅니다.

타입: 문자열

유효 값: `SUCCESS | FAILURE`

필수 사항 여부: 예

Token

동일한 실행 내에서 여러 Lambda 단계에 대한 여러 콜백을 구분하는 데 사용됩니다.

타입: 문자열

길이 제한: 최소 길이는 1. 최대 길이는 64.

패턴: `\w+`

필수 여부: 예

WorkflowId

워크플로의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `w-([a-z0-9]{17})`

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반송합니다.

Errors

모든 작업에서 공통적으로 발생하는 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

AccessDeniedException

이 작업을 수행할 수 있는 충분한 액세스 권한이 없습니다.

HTTP 상태 코드: 400

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

ThrottlingException

요청 제한 때문에 요청이 거부되었습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

StartDirectoryListing

원격 SFTP 서버에서 디렉토리 내용 목록을 검색합니다. 커넥터 ID, 출력 경로 및 원격 디렉터리 경로를 지정합니다. 선택적 `MaxItems` 값을 지정하여 원격 디렉터리에서 나열되는 최대 항목 수를 제어할 수도 있습니다. 이 API는 원격 디렉터리의 모든 파일 및 디렉터리 목록 (최대값까지) 을 반환하지만 하위 디렉터리의 파일 또는 폴더는 반환하지 않습니다. 즉, 한 수준 깊이의 파일 및 디렉터리 목록만 반환합니다.

목록 파일을 받은 후 `StartFileTransfer` API 호출의 `RetrieveFilePaths` 파라미터에 전송하려는 파일을 제공할 수 있습니다.

출력 파일의 명명 규칙은 다음과 *connector-ID-listing-ID*`.json` 같습니다. 출력 파일에는 다음 정보가 포함됩니다.

- `filePath`: 원격 서버의 SFTP 커넥터에 대한 목록 요청 디렉토리를 기준으로 한 원격 파일의 전체 경로입니다.
- `modifiedTimestamp`: 파일이 마지막으로 수정된 시간 (UTC 시간 형식) 이 필드는 선택 사항입니다. 원격 파일 속성에 타임스탬프가 없는 경우 파일 목록에서 제외됩니다.
- `size`: 파일 크기 (바이트). 이 필드는 선택 사항입니다. 원격 파일 속성에 파일 크기가 포함되지 않은 경우 파일 목록에서 제외됩니다.
- `path`: 원격 서버의 SFTP 커넥터에 대한 목록 작성 요청 디렉터리를 기준으로 한 원격 디렉터리의 전체 경로입니다.
- `truncated`: 목록 출력에 원격 디렉터리에 포함된 모든 항목이 포함되는지 여부를 나타내는 플래그입니다. `Truncated`출력 값이 true인 경우 선택적 `max-items` 입력 속성에 제공된 값을 늘려 더 많은 항목을 나열할 수 있습니다 (최대 허용 목록 크기인 10,000개 항목).

구문 요청

```
{
   "ConnectorId": "string",
   "MaxItems": number,
   "OutputDirectoryPath": "string",
   "RemoteDirectoryPath": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

ConnectorId

커넥터의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `c-([0-9a-f]{17})`

필수 사항 여부: Yes

MaxItems

검색할 파일/디렉터리 이름의 최대 수를 지정할 수 있는 선택적 매개 변수입니다. 기본값은 1,000입니다.

타입: 정수

유효한 범위: 최소값 1. 최대값은 10,000입니다.

필수 여부: 아니요

OutputDirectoryPath

디렉터리 목록 결과를 저장할 Amazon S3 스토리지의 경로 (버킷 및 접두사) 를 지정합니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: `(.)+`

필수 사항 여부: Yes

RemoteDirectoryPath

콘텐츠를 나열하려는 원격 SFTP 서버의 디렉토리를 지정합니다.

유형: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: `(.)+`

필수 항목 여부: 예

응답 구문

```
{
   "ListingId": "string",
   "OutputFileName": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

ListingId

디렉터리 목록 호출의 고유 식별자를 반환합니다.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 512입니다.

패턴: `[0-9a-zA-Z./-]+`

OutputFileName

결과가 저장된 파일 이름을 반환합니다. 커넥터 ID와 리스팅 ID:의 `<connector-id>-<listing-id>.json` 조합입니다.

타입: 문자열

길이 제약: 최소 길이는 26입니다. 최대 길이는 537개입니다.

패턴: `c-([0-9a-f]{17})-[0-9a-zA-Z./-]+.json`

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 서비스에 오류가 발생할 때 발생합니다. AWS Transfer Family

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

ThrottlingException

요청 제한 때문에 요청이 거부되었습니다.

HTTP 상태 코드: 400

예

예

다음 예에서는 지정된 커넥터로 식별되는 원격 SFTP 서버의 home 폴더 내용을 나열합니다. 결과는 Amazon S3 위치와 `/DOC-EXAMPLE-BUCKET/connector-files` 이름이 지정된 파일에 저장됩니다`c-AAAA1111BBBB2222C-6666abcd-11aa-22bb-cc33-0000aaaa3333.json`.

샘플 요청

```
{
    "ConnectorId": "c-AAAA1111BBBB2222C",
    "MaxItems": "10",
    "OutputDirectoryPath": "/DOC-EXAMPLE-BUCKET/connector-files",
    "RemoteDirectoryPath": "/home"
```

```
}
```

샘플 응답

```
{
    "ListingId": "6666abcd-11aa-22bb-cc33-0000aaaa3333",
    "OutputFileName": "c-AAAA1111BBBB2222C-6666abcd-11aa-22bb-cc33-0000aaaa3333.json"
}
```

```
// under bucket "DOC-EXAMPLE-BUCKET"
connector-files/c-AAAA1111BBBB2222C-6666abcd-11aa-22bb-cc33-0000aaaa3333.json
{
    "files": [
        {
            "filePath": "/home/what.txt",
            "modifiedTimestamp": "2024-01-30T20:34:54Z",
            "size" : 2323
        },
        {
            "filePath": "/home/how.pgp",
            "modifiedTimestamp": "2024-01-30T20:34:54Z",
            "size" : 51238
        }
    ],
    "paths": [
        {
            "path": "/home/magic"
        },
        {
            "path": "/home/aws"
        },
    ],
    "truncated": false
}
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET

- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

StartFileTransfer

로컬 AWS 스토리지와 원격 AS2 또는 SFTP 서버 간의 파일 전송을 시작합니다.

- AS2 커넥터의 경우, `ConnectorId` 및 하나 이상 `SendFilePaths`을 지정하여 전송할 파일을 식별합니다.
- SFTP 커넥터의 경우 File Transfer은 아웃바운드 또는 인바운드일 수 있습니다. 두 경우 모두 `ConnectorId`를 지정합니다. 전송 방향에 따라 다음 항목도 지정합니다.
  - 파트너의 SFTP 서버에서 Amazon Web Services 스토리지로 파일을 전송하는 경우, 전송할 파일을 식별하기 위해 하나 이상 `RetrieveFilePaths` 을 지정하고 대상 폴더를 지정하기 위해 `LocalDirectoryPath` 를 지정합니다.
  - AWS 스토리지에서 파트너의 SFTP 서버로 파일을 전송하는 경우 전송할 파일을 식별하기 위해 하나 이상을 `SendFilePaths`을 지정하고 대상 폴더를 지정하기 위해 `RemoteDirectoryPath`를 지정합니다.

구문 요청

```
{
   "ConnectorId": "string",
   "LocalDirectoryPath": "string",
   "RemoteDirectoryPath": "string",
   "RetrieveFilePaths": [ "string" ],
   "SendFilePaths": [ "string" ]
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

ConnectorId

커넥터의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `c-([0-9a-f]{17})`

필수 사항 여부: Yes

LocalDirectoryPath

인바운드 전송의 경우, LocaDirectoryPath는 파트너의 SFTP 서버에서 전송되는 하나 이상의 파일에 대한 대상을 지정합니다.

타입: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: (.)+

Required: No

RemoteDirectoryPath

아웃바운드 전송의 경우, RemoteDirectoryPath는 파트너의 SFTP 서버로 전송되는 하나 이상의 파일에 대한 대상을 지정합니다. RemoteDirectoryPath를 지정하지 않는 경우, 전송된 파일의 대상은 SFTP 사용자의 홈 디렉터리입니다.

타입: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: (.)+

Required: No

RetrieveFilePaths

파트너의 SFTP 서버에 대한 하나 이상의 소스 경로 각 문자열은 한 번의 인바운드 File Transfer에 대한 소스 파일 경로를 나타냅니다.

타입: 문자열 배열

배열 멤버: 최소 항목 수는 1개입니다. 최대 항목 수는 10개입니다.

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 1024입니다.

패턴: (.)+

Required: No

SendFilePaths

Amazon S3 스토리지를 위한 하나 이상의 소스 경로. 각 문자열은 한 번의 아웃바운드 파일 전송에 대한 소스 파일 경로를 나타냅니다. 예를 들어 *DOC-EXAMPLE-BUCKET/myfile.txt* 입니다.

> **Note**
>
> *DOC-EXAMPLE-BUCKET* 를 실제 버킷 중 하나와 바꾸세요.

타입: 문자열 배열

배열 멤버: 최소 항목 수는 1개입니다. 최대 항목 수는 10개입니다.

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 1024입니다.

패턴: `(.)+`

필수 여부: 아니요

응답 구문

```
{
   "TransferId": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

TransferId

파일 전송에 대해 고유 식별자를 반환합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 512입니다.

패턴: `[0-9a-zA-Z./-]+`

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

서비스에 오류가 발생하면 이 예외가 발생합니다. AWS Transfer Family

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

ThrottlingException

요청 제한 때문에 요청이 거부되었습니다.

HTTP 상태 코드: 400

예

예

다음 예시에서는 Transfer Family 서버에서 원격 거래 파트너의 엔드포인트로 AS2 File Transfer를 시작합니다. *DOC-EXAMPLE-BUCKET* 를 실제 버킷 중 하나로 바꾸세요.

샘플 요청

```
{
    "ConnectorId": "c-AAAA1111BBBB2222C",
```

```
    "SendFilePaths": [
        "/DOC-EXAMPLE-BUCKET/myfile-1.txt",
        "/DOC-EXAMPLE-BUCKET/myfile-2.txt",
        "/DOC-EXAMPLE-BUCKET/myfile-3.txt"
    ]
}
```

샘플 응답

```
{
   "TransferId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

예

다음 예에서는 로컬 AWS 스토리지에서 원격 SFTP 서버로의 파일 전송을 시작합니다.

샘플 요청

```
{
    "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"
}
```

샘플 응답

```
{
   "TransferId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

예

다음 예에서는 원격 SFTP 서버에서 로컬 AWS 스토리지로 File Transfer를 시작합니다.

샘플 요청

```
{
```

```
    "ConnectorId": "c-111122223333AAAAA",
    "RetrieveFilePaths": [
        "/MySFTPFolder/toTranferFamily/myfile-1.txt",
        "/MySFTPFolder/toTranferFamily/myfile-2.txt",
        "/MySFTPFolder/toTranferFamily/myfile-3.txt"
    ],
    "LocalDirectoryPath": "/DOC-EXAMPLE-BUCKET/mySourceFiles"
}
```

샘플 응답

```
{
   "TransferId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa"
}
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

StartServer

File Transfer 프로토콜이 활성화된 서버의 상태를 OFFLINE에서 ONLINE로 변경합니다. 이미 ONLINE 설치된 서버에는 영향을 주지 않습니다. ONLINE서버는 File Transfer 작업을 수락하고 처리할 수 있습니다.

STARTING 상태는 서버가 완전히 응답할 수 없거나 완전히 온라인 상태가 아닌 중간 상태임을 나타냅니다. START_FAILED의 값은 오류 상태를 나타낼 수 있습니다.

이 직접 호출에서 응답이 반환되지 않습니다.

구문 요청

```
{
   "ServerId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

ServerId

시작하려는 서버에 대해 시스템에서 할당한 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: s-([0-9a-f]{17})

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반송합니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

ThrottlingException

요청 제한 때문에 요청이 거부되었습니다.

HTTP 상태 코드: 400

예

예

다음 예에서는 서버를 시작합니다.

샘플 요청

```
{
   "ServerId": "s-01234567890abcdef"
}
```

예

다음은 이 API 직접 호출에 대한 샘플 응답입니다.

샘플 응답

```
{
   "ServerId": "s-01234567890abcdef"
}
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

StopServer

File Transfer 프로토콜이 활성화된 서버의 상태를 OFFLINE에서 ONLINE로 변경합니다. OFFLINE 서버는 File Transfer 작업을 수락하고 처리할 수 없습니다. 서버 및 사용자 속성과 같이 서버에 연결된 정보는 서버를 중지해도 영향을 받지 않습니다.

> **Note**
>
> 서버를 중지해도 File Transfer 프로토콜 엔드포인트 청구가 줄어들거나 영향을 미치지는 않습니다. 요금 청구를 중지하려면 서버를 삭제해야 합니다.

STOPPING 상태는 서버가 완전히 응답할 수 없거나 완전히 오프라인 상태가 아닌 중간 상태임을 나타냅니다. STOP_FAILED의 값은 오류 상태를 나타낼 수 있습니다.

이 직접 호출에서 응답이 반환되지 않습니다.

구문 요청

```
{
   "ServerId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

ServerId

중지한 서버에 대해 시스템에서 할당한 고유 식별자.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반송합니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

ThrottlingException

요청 제한 때문에 요청이 거부되었습니다.

HTTP 상태 코드: 400

예

예

다음 예에서는 서버를 중지합니다.

샘플 요청

```
{
   "ServerId": "s-01234567890abcdef"
}
```

예

다음은 이 API 직접 호출에 대한 샘플 응답입니다.

샘플 응답

```
{
   "ServerId": "s-01234567890abcdef"
}
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

TagResource

Amazon 리소스 이름(ARN)으로 식별되는 키-값 쌍을 리소스에 첨부합니다. 리소스는 사용자, 서버, 역할 및 기타 개체입니다.

이 직접적인 호출에서 응답이 반환되지 않았습니다.

구문 요청

```
{
   "Arn": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

Arn

서버, 사용자 또는 역할과 같은 특정 AWS 리소스의 Amazon 리소스 이름 (ARN).

타입: 문자열

길이 제약 조건: 최소 길이는 20입니다. 최대 길이는 1600입니다.

패턴: `arn:\S+`

필수 사항 여부: Yes

Tags

타입별로 리소스를 그룹화하고 검색하는 데 사용할 수 있는 ARN에 할당된 키-값 쌍입니다. 어떤 목적으로든 이 메타데이터를 리소스(서버, 사용자, 워크플로 등)에 연결할 수 있습니다.

타입: Tag 객체 배열

배열 멤버: 최소 항목 수는 1개입니다. 최대 항목 수는 50개입니다.

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반송합니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 서비스에 오류가 발생할 때 발생합니다. AWS Transfer Family

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

예

예

다음 예에서는 File Transfer 프로토콜이 활성화된 서버에 태그를 추가합니다.

샘플 요청

```
{
   "Arn": "arn:aws:transfer:us-east-1:176354371281:server/s-01234567890abcdef",
   "Tags": [
      {
         "Key": "Group",
         "Value": "Europe"
      }
   ]
}
```

예

이 예에서는 의 한 가지 사용법을 보여줍니다 TagResource.

샘플 응답

```
         HTTP 200 response with an empty HTTP body.
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

TestConnection

SFTP 커넥터가 성공적으로 설정되었는지 테스트합니다. 로컬 AWS 스토리지와 거래 파트너의 SFTP 서버 간에 파일을 전송할 수 있는지 테스트하려면 이 작업을 호출하는 것이 좋습니다.

구문 요청

```
{
   "ConnectorId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

ConnectorId

커넥터의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `c-([0-9a-f]{17})`

필수 항목 여부: 예

응답 구문

```
{
   "ConnectorId": "string",
   "Status": "string",
   "StatusMessage": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

ConnectorId

테스트 중인 커넥터 객체의 식별자를 반환합니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `c-([0-9a-f]{17})`

Status

테스트 성공 시 `OK` 또는 테스트 실패 시 `ERROR`를 반환됩니다.

타입: 문자열

StatusMessage

테스트가 성공한 경우 `Connection succeeded`를 반환합니다. 또는 테스트 실패 시 설명이 포함된 오류 메시지를 반환합니다. 다음 목록은 표시되는 오류 메시지에 따른 문제 해결 세부 정보를 제공합니다.

- 암호 이름이 역할 전송 권한의 이름과 일치하는지 확인하세요.
- 커넥터 구성에서 서버 URL을 확인하고 로그인 자격 증명이 커넥터 외부에서 제대로 작동하는지 확인합니다.
- 암호가 존재하고 형식이 올바른지 확인하세요.
- 커넥터 구성의 신뢰 호스트 키가 `ssh-keyscan` 출력과 일치하는지 확인하세요.

타입: 문자열

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 서비스에 오류가 발생할 때 발생합니다. AWS Transfer Family

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

예

예

다음 예에서는 원격 서버와의 연결을 테스트합니다.

```
aws transfer test-connection --connector-id c-abcd1234567890fff
```

샘플 응답

성공하면 API 호출은 다음 세부 정보를 반환합니다.

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

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++

- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

TestIdentityProvider

파일 전송 프로토콜 지원 서버의 IdentityProviderType가 AWS_DIRECTORY_SERVICE 또는 API_Gateway인 경우 ID 제공자가 성공적으로 설정되었는지 테스트합니다. 서버를 생성하는 즉시 이 작업을 호출하여 인증 방법을 테스트하는 것이 좋습니다. 이렇게 하면 ID 제공자 통합 문제를 해결하여 사용자가 서비스를 성공적으로 사용할 수 있도록 할 수 있습니다.

ServerId 및 UserName 파라미터가 필요합니다. ServerProtocol, SourceIp, 및 UserPassword은 모두 옵션입니다.

유념할 사항:

- 서버의 IdentityProviderType가 SERVICE_MANAGED인 경우에는 TestIdentityProvider를 사용할 수 없습니다.
- TestIdentityProvider는 키와 함께 사용할 수 없습니다. 암호만 받아들입니다.
- TestIdentityProvider는 키와 암호를 처리하는 맞춤 ID 공급자의 암호 작업을 테스트할 수 있습니다.
- 파라미터에 잘못된 값을 입력하면 Response 필드가 비어 있습니다.
- 서비스 관리 사용자를 사용하는 서버의 서버 ID를 제공하면 오류가 발생합니다.

  An error occurred (InvalidRequestException) when calling the TestIdentityProvider operation: s-*server-ID* not configured for external auth
- 실제 전송 서버를 식별하지 않는 --server-id 파라미터에 서버 ID를 입력하면 다음과 같은 오류 메시지가 나타납니다.

  An error occurred (ResourceNotFoundException) when calling the TestIdentityProvider operation: Unknown server.

  서버가 다른 지역에 있을 수 있습니다. 미국 동부(오하이오)에 있는 서버를 지정하는 것과 같이 --region region-code(예: --region us-east-2)를 추가하여 지역을 지정할 수 있습니다.

구문 요청

```
{
   "ServerId": "string",
   "ServerProtocol": "string",
   "SourceIp": "string",
```

```
    "UserName": "string",
    "UserPassword": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

ServerId

서버에 대해 시스템에서 할당한 식별자입니다. 사용자 이름과 암호를 사용해 서버의 사용자 인증 방법을 테스트합니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

필수 여부: 예

ServerProtocol

테스트할 파일 전송 프로토콜의 타입입니다.

사용 가능한 프로토콜은 다음과 같습니다:

- Secure Shell (SSH) File Transfer 프로토콜 (SFTP)
- File Transfer 프로토콜 보안(FTPS)
- File Transfer 프로토콜(FTP)
- 적용성 보고서 2(AS2)

타입: 문자열

유효 값: `SFTP | FTP | FTPS | AS2`

필수 여부: 아니요

SourceIp

테스트할 계정의 소스 IP 주소입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이 32.

패턴: `\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}`

필수 여부: 아니요

UserName

테스트할 계정의 명칭입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 3입니다. 최대 길이는 100입니다.

패턴: `[\w][\w@.-]{2,99}`

필수 여부: 예

UserPassword

테스트할 계정의 암호입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이 1,024.

필수 여부: 아니요

응답 구문

```
{
   "Message": "string",
   "Response": "string",
   "StatusCode": number,
   "Url": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

Message

테스트 성공 여부를 나타내는 메시지입니다.

> **Note**
>
> 빈 문자열이 반환되는 경우 가장 가능성이 높은 원인은 잘못된 사용자 이름 또는 암호로 인해 인증에 실패했기 때문입니다.

타입: 문자열

Response

API Gateway 또는 Lambda 함수에서 반환되는 응답입니다.

타입: 문자열

StatusCode

API Gateway 또는 Lambda 함수의 응답인 HTTP 상태 코드입니다.

타입: 정수

Url

사용자를 인증하는 데 사용되는 서비스의 엔드포인트입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 255입니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

AWS Transfer Family 서비스에 오류가 발생하면 이 예외가 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

예

예

다음 요청은 ID 공급자로부터 사용자 이름과 암호 조합이 사용할 수 있는 유효한 ID라는 메시지를 반환합니다 AWS Transfer Family.

샘플 요청

```
{
   "ServerID": "s-01234567890abcdef",
   "UserName": "my_user",
   "UserPassword": "MyPassword-1"
}
```

예

다음 응답은 성공적인 테스트에 대한 샘플 응답을 보여줍니다.

샘플 응답

```
      "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": ""
```

예

다음 응답은 지정된 사용자가 액세스 권한이 있는 둘 이상의 그룹에 속해 있음을 나타냅니다.

```
        "Response":"",
        "StatusCode":200,
        "Message":"More than one associated access found for user's groups."
```

예

API Gateway를 사용하여 맞춤 ID 공급자를 생성하고 구성한 경우 다음 명령을 입력하여 사용자를 테스트할 수 있습니다.

`aws transfer test-identity-provider --server-id` *`s-0123456789abcdefg`* `--user-name` *`myuser`*

여기서 s-0123456789abcdefg는 전송 서버이고 myuser는 맞춤 사용자의 사용자 이름입니다.

명령이 성공하면 다음과 유사한 응답이 표시됩니다, 여기서:

- AWS 계정 ID는 012345678901입니다.
- 사용자 역할은 user-role-api-gateway임
- 홈 디렉터리는 myuser-bucket임
- 퍼블릭 키는 public-key임
- 호출 URL은 invocation-URL임

```
{
    "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"
}
```

참고

언어별 SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오. AWS

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

UntagResource

Amazon 리소스 이름(ARN)으로 구분할 수 있는 키-값 쌍. 리소스는 사용자, 서버, 역할 및 기타 개체입니다.

이 직접 호출에서 응답이 반환되지 않습니다.

구문 요청

```
{
   "Arn": "string",
   "TagKeys": [ "string" ]
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

Arn

태그가 제거될 리소스의 값입니다. Amazon 리소스 이름 (ARN) 은 서버, 사용자 또는 역할과 같은 특정 AWS 리소스의 식별자입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 20입니다. 최대 길이는 1600입니다.

패턴: `arn:\S+`

필수 사항 여부: Yes

TagKeys

TagKeys ARN에 할당된 키-값 쌍으로, 유형별로 리소스를 그룹화하고 검색하는 데 사용할 수 있습니다. 이 메타데이터는 어떤 목적으로든 리소스에 첨부할 수 있습니다.

타입: 문자열 배열

배열 멤버: 최소 항목 수는 1개입니다. 최대 항목 수는 50개입니다.

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 128입니다.

필수 여부: 예

Response Elements

작업이 성공하면 서비스가 비어 있는 HTTP 본문과 함께 HTTP 200 응답을 반송합니다.

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 서비스에 오류가 발생할 때 발생합니다. AWS Transfer Family

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

예

예

다음 예에서는 File Transfer 프로토콜이 활성화된 서버의 태그를 제거합니다.

샘플 요청

```
{
   "Arn": "arn:aws:transfer:us-east-1:176354371281:server/s-01234567890abcdef",
   "TagKeys": "Europe" ]
}
```

예

이 예에서는 의 한 가지 사용법을 보여줍니다 UntagResource.

샘플 응답

```
        HTTP 200 response with an empty HTTP body.
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

UpdateAccess

ServerID 및 ExternalID 파라미터에 지정된 액세스의 파라미터를 업데이트할 수 있습니다.

구문 요청

```
{
   "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"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

ExternalId

디렉터리 내의 특정 그룹을 식별하는 데 필요한 고유 식별자입니다. 연결하는 그룹의 사용자는 사용하는 활성화된 프로토콜을 통해 Amazon S3 또는 Amazon EFS 리소스에 액세스할 수 AWS Transfer Family있습니다. 그룹 이름을 아는 경우 PowerShell Windows를 사용하여 다음 명령을 실행하여 SID 값을 볼 수 있습니다.

```
Get-ADGroup -Filter {samAccountName -like "YourGroupName*"} -Properties
* | Select SamAccountName,ObjectSid
```

이 명령에서 Active Directory 그룹의 YourGroupName이름으로 바꾸십시오.

이 파라미터를 확인하는 데 사용되는 정규 표현식은 공백 없이 대문자 및 소문자 영숫자로 구성된 문자의 문자열입니다. 밑줄이나 또한 =,.@:/- 문자도 포함할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: `S-1-[\d-]+`

필수 여부: 예

HomeDirectory

사용자가 클라이언트를 사용하여 서버에 로그인하는 경우 사용자를 위한 랜딩 디렉터리(폴더).

`HomeDirectory`의 예: `/bucket_name/home/mydirectory`

> **Note**
>
> `HomeDirectory` 파라미터는 `HomeDirectoryType`이(가) `PATH`(으)로 설정된 경우에만 사용합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 1024입니다.

패턴: `(|/.*)`

필수 여부: 아니요

HomeDirectoryMappings

사용자에게 표시할 Amazon S3 또는 Amazon EFS 경로 및 키와 이러한 경로 및 키를 표시할 방법을 지정하는 논리적 디렉터리 매핑입니다. `Entry` 및 `Target` 쌍을 지정해야 합니다. 여기서 `Entry`는 경로가 표시되는 방식을 보여주고 `Target`는 실제 Amazon S3 경로입니다. 대상만 지정하는 경우, 그대로 표시됩니다. 또한 AWS Identity and Access Management (IAM) 역할이 경로에 대한 액세스를 제공하는지 확인해야 합니다. `Target` 이 값은 LOGICAL로 설정된 경우에만 `HomeDirectoryType`를 설정할 수 있습니다.

다음은 `Entry` 및 `Target` 쌍의 예입니다.

```
[ { "Entry": "/directory1", "Target": "/bucket_name/home/
mydirectory" } ]
```

대부분의 경우 세션 정책 대신 이 값을 사용하여 사용자를 지정된 홈 디렉터리("chroot")로 제한할 수 있습니다. 이렇게 하려면 Entry을 /로 설정하고, Target을 HomeDirectory 파라미터 값으로 설정하면 됩니다.

다음은 chroot에 대한 Entry 및 Target 쌍의 예입니다.

```
[ { "Entry": "/", "Target": "/bucket_name/home/mydirectory" } ]
```

타입: HomeDirectoryMapEntry 객체 배열

어레이 멤버: 최소 항목 수 1개. 최대 항목 수는 50000개입니다.

필수 여부: 아니요

HomeDirectoryType

사용자가 서버에 로그인하는 경우 홈 디렉터리가 될 랜딩 디렉터리(폴더) 타입입니다. PATH로 설정하면, 사용자는 파일 전송 프로토콜 클라이언트에서와 같이 절대 Amazon S3 버킷 또는 EFS 경로를 볼 수 있습니다. LOGICAL로 설정하면, HomeDirectoryMappings에서 Amazon S3 또는 Amazon EFS 경로를 사용자에게 표시할 방법에 대한 매핑을 제공해야 합니다.

> **Note**
>
> HomeDirectoryType이 LOGICAL인 경우, HomeDirectoryMappings 파라미터를 사용하여 매핑을 제공해야 합니다. 반면에 HomeDirectoryType이 PATH인 경우, HomeDirectory 파라미터를 사용하여 절대 경로를 제공하세요. 템플릿에 HomeDirectory 및 HomeDirectoryMappings를 모두 포함할 수는 없습니다.

타입: 문자열

유효 값: PATH | LOGICAL

필수 여부: 아니요

Policy

여러 사용자가 동일한 AWS Identity and Access Management (IAM) 역할을 사용할 수 있도록 하기 위한 사용자 세션 정책. 이 정책은 Amazon S3 버킷의 부분에 대한 사용자의 액세

스 범위를 축소합니다. 이 정책 내에서 사용할 수 있는 변수는 `${Transfer:UserName}`, `${Transfer:HomeDirectory}` 및 `${Transfer:HomeBucket}`입니다.

> **Note**
>
> 이 정책은 `ServerId`의 도메인이 Amazon S3인 경우에만 적용됩니다. Amazon EFS는 세션 정책을 사용하지 않습니다.
> 세션 정책의 경우 정책을 정책의 Amazon 리소스 이름 (ARN) 대신 JSON Blob으로 AWS Transfer Family 저장합니다. 정책을 JSON Blob으로 저장하고 `Policy` 인수로 전달합니다.
> 세션 정책의 예는 세션 정책 예를 참조하세요.
> 자세한 내용은 AWS보안 토큰 서비스 AssumeRoleAPI 참조를 참조하십시오.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 2048입니다.

필수 여부: 아니요

PosixProfile

Amazon EFS 파일 시스템에 대한 사용자의 액세스를 통제하는 사용자 ID(`Uid`), 그룹 ID(`Gid`) 및 보조 그룹 ID(`SecondaryGids`)를 포함한 전체 POSIX 자격 증명입니다. 파일 시스템의 파일 및 디렉터리에 설정된 POSIX 권한에 따라 Amazon EFS 파일 시스템에서 파일을 송수신할 때 사용자에게 제공되는 액세스 수준이 결정됩니다.

타입: PosixProfile 객체

필수 항목 여부: 아니요

Role

Amazon S3 버킷 또는 Amazon EFS 파일 시스템에 대한 사용자의 액세스를 제어하는 AWS Identity and Access Management (IAM) 역할의 Amazon 리소스 이름 (ARN). 이 역할에 연결된 정책은 Amazon S3 버킷 또는 Amazon EFS 파일 시스템에 대한 파일 송수신 시 사용자에게 제공할 액세스의 수준을 결정합니다. 또한 IAM 역할에는 사용자의 전송 요청을 처리할 때 서버가 해당 리소스에 액세스할 수 있도록 허용하는 신뢰 관계가 포함되어야 합니다.

타입: 문자열

길이 제약: 최소 길이는 20. 최대 길이는 2,048.

패턴: `arn:.*role/\S+`

필수 여부: 아니요

ServerId

서버 인스턴스에 대해 시스템에서 할당한 고유 식별자입니다. 이 항목은 사용자를 추가한 특정 서버입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

필수 항목 여부: 예

응답 구문

```
{
   "ExternalId": "string",
   "ServerId": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

ExternalId

AWS Transfer Family를 사용하여 활성화된 프로토콜을 통해 Amazon S3 또는 Amazon EFS 리소스에 액세스할 수 있는 그룹의 외부 식별자입니다.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: `S-1-[\d-]+`

ServerId

사용자가 연결된 서버의 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

AWS Transfer Family 서비스에 오류가 발생하면 이 예외가 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceExistsException

요청된 리소스가 존재하지 않거나 명령에 지정된 리전이 아닌 다른 리전에 있습니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

ThrottlingException

요청 제한 때문에 요청이 거부되었습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

UpdateAgreement

기존 계약의 일부 파라미터를 업데이트합니다. 업데이트할 파라미터의 새 값과 함께 업데이트하려는 계약의 AgreementId과 ServerId를 입력합니다.

구문 요청

```
{
   "AccessRole": "string",
   "AgreementId": "string",
   "BaseDirectory": "string",
   "Description": "string",
   "LocalProfileId": "string",
   "PartnerProfileId": "string",
   "ServerId": "string",
   "Status": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

AccessRole

커넥터는 AS2 또는 SFTP 프로토콜을 사용하여 파일을 전송하는 데 사용됩니다. 액세스 역할에는 사용할 AWS Identity and Access Management 역할의 Amazon 리소스 이름 (ARN) 을 제공합니다.

AS2 커넥터의 경우

AS2를 사용하여 StartFileTransfer를 호출하고 요청 파라미터인 SendFilePaths에 파일 경로를 지정하여 파일을 전송할 수 있습니다. 파일의 상위 디렉터리(예: `--send-file-paths /bucket/dir/file.txt`의 경우 상위 디렉터리는 /bucket/dir/임)를 사용하여 처리된 AS2 메시지 파일을 임시로 저장하고, 파트너로부터 수신 시 MDN을 저장하고, 전송의 관련 메타데이터를 포함하는 최종 JSON 파일을 작성합니다. 따라서 AccessRole은(는) StartFileTransfer 요청에 사용된 파일 위치의 상위 디렉터리에 대한 읽기 및 쓰기 액세스를 제공해야 합니다. 또한 StartFileTransfer와(과) 함께 전송하려는 파일의 상위 디렉터리에 대한 읽기 및 쓰기 액세스 권한을 제공해야 합니다.

AS2 커넥터에 기본 인증을 사용하는 경우 액세스 역할에는 암호에 대한 secretsmanager:GetSecretValue 권한이 필요합니다. Secrets Manager에서 관리 키 대신 고

객 AWS 관리 키를 사용하여 암호를 암호화하는 경우 역할에도 해당 키에 대한 `kms:Decrypt` 권한이 필요합니다.

SFTP 커넥터의 경우

액세스 역할이 `StartFileTransfer` 요청에서 사용된 파일 위치의 상위 디렉터리에 대한 읽기 및 쓰기 액세스 권한을 제공하는지 확인하세요. 또한 역할이 `secretsmanager:GetSecretValue` 권한을 제공하는지 확인하십시오. AWS Secrets Manager

타입: 문자열

길이 제약: 최소 길이는 20. 최대 길이는 2,048.

패턴: `arn:.*role/\S+`

필수 여부: 아니요

AgreementId

계약의 고유 식별자입니다. 계약을 생성하면 이 식별자가 반환됩니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `a-([0-9a-f]{17})`

필수 여부: 예

BaseDirectory

전송되는 파일의 랜딩 디렉터리(폴더)를 변경하려면 사용할 버킷 폴더(예: */DOC-EXAMPLE-BUCKET/home/mydirectory*)를 입력합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 1024입니다.

패턴: `(|/.*)`

필수 여부: 아니요

Description

기존 설명을 대체하려면 계약에 대한 간단한 설명을 제공하세요.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 200입니다.

패턴: `[\p{Graph}]+`

필수 여부: 아니요

LocalProfileId

AS2 로컬 프로필의 고유 식별자입니다.

로컬 프로필 식별자를 변경하려면 여기에 새 값을 입력합니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `p-([0-9a-f]{17})`

필수 여부: 아니요

PartnerProfileId

파트너 프로필의 고유 식별자입니다. 파트너 프로필 식별자를 변경하려면 여기에 새 값을 입력하세요.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `p-([0-9a-f]{17})`

필수 여부: 아니요

ServerId

서버 인스턴스에 대해 시스템에서 할당한 고유 식별자입니다. 이는 계약에서 사용하는 특정 서버를 나타냅니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

필수 여부: 예

Status

계약 상태를 업데이트하여 비활성 계약을 활성화하거나 그 반대로 업데이트할 수 있습니다.

타입: 문자열

유효 값: `ACTIVE | INACTIVE`

필수 항목 여부: 아니요

응답 구문

```
{
   "AgreementId": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

AgreementId

계약의 고유 식별자입니다. 계약을 생성하면 이 식별자가 반환됩니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `a-([0-9a-f]{17})`

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

AWS Transfer Family 서비스에 오류가 발생하면 이 예외가 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceExistsException

요청된 리소스가 존재하지 않거나 명령에 지정된 리전이 아닌 다른 리전에 있습니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

ThrottlingException

요청 제한 때문에 요청이 거부되었습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK

- AWS Python용 SDK
- AWS 루비 V3용 SDK

UpdateCertificate

인증서의 활성 및 비활성 날짜를 업데이트합니다.

구문 요청

```
{
   "ActiveDate": number,
   "CertificateId": "string",
   "Description": "string",
   "InactiveDate": number
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

ActiveDate

인증서가 활성화되는 시기를 지정하는 선택적 날짜입니다.

타입: Timestamp

필수 여부: 아니요

CertificateId

업데이트하려는 인증서 객체의 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 22입니다.

패턴: `cert-([0-9a-f]{17})`

필수 여부: 예

Description

인증서를 식별하는 데 도움이 되는 간단한 설명입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 200입니다.

패턴: `[\p{Graph}]+`

필수 여부: 아니요

InactiveDate

인증서가 비활성화되는 시기를 지정하는 선택적 날짜.

타입: Timestamp

필수 여부: 아니요

응답 구문

```
{
   "CertificateId": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

CertificateId

업데이트 중인 인증서 객체의 식별자를 반환합니다.

타입: 문자열

길이 제약 조건: 고정 길이는 22입니다.

패턴: `cert-([0-9a-f]{17})`

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

ThrottlingException

요청 제한 때문에 요청이 거부되었습니다.

HTTP 상태 코드: 400

예

예

다음 예에서는 활성 날짜를 2022년 1월 16일 16:12:07 UTC -5시간으로 설정하여 인증서의 활성 날짜를 업데이트합니다.

샘플 요청

```
aws transfer update-certificate --certificate-id c-abcdefg123456hijk --active-date
 2022-01-16T16:12:07-05:00
```

예

다음은 이 API 호출의 샘플 응답입니다.

샘플 응답

```
"CertificateId": "c-abcdefg123456hijk"
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

UpdateConnector

기존 커넥터의 일부 파라미터를 업데이트합니다. 업데이트할 파라미터의 새 값과 함께 업데이트하려는 프로필의 `ConnectorId`를 입력합니다.

구문 요청

```
{
   "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"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

AccessRole

커넥터는 AS2 또는 SFTP 프로토콜을 사용하여 파일을 전송하는 데 사용됩니다. 액세스 역할에는 사용할 AWS Identity and Access Management 역할의 Amazon 리소스 이름 (ARN) 을 제공합니다.

AS2 커넥터의 경우

AS2를 사용하여 StartFileTransfer를 호출하고 요청 파라미터인 SendFilePaths에 파일 경로를 지정하여 파일을 전송할 수 있습니다. 파일의 상위 디렉터리(예: --send-file-paths /bucket/dir/file.txt의 경우 상위 디렉터리는 /bucket/dir/임)를 사용하여 처리된 AS2 메시지 파일을 임시로 저장하고, 파트너로부터 수신 시 MDN을 저장하고, 전송의 관련 메타데이터를 포함하는 최종 JSON 파일을 작성합니다. 따라서 AccessRole은(는) StartFileTransfer 요청에 사용된 파일 위치의 상위 디렉터리에 대한 읽기 및 쓰기 액세스를 제공해야 합니다. 또한 StartFileTransfer와(과) 함께 전송하려는 파일의 상위 디렉터리에 대한 읽기 및 쓰기 액세스 권한을 제공해야 합니다.

AS2 커넥터에 기본 인증을 사용하는 경우 액세스 역할에는 암호에 대한 secretsmanager:GetSecretValue 권한이 필요합니다. Secrets Manager에서 관리 키 대신 고객 AWS 관리 키를 사용하여 암호를 암호화하는 경우 역할에도 해당 키에 대한 kms:Decrypt 권한이 필요합니다.

SFTP 커넥터의 경우

액세스 역할이 StartFileTransfer 요청에서 사용된 파일 위치의 상위 디렉터리에 대한 읽기 및 쓰기 액세스 권한을 제공하는지 확인하세요. 또한 역할이 secretsmanager:GetSecretValue 권한을 제공하는지 확인하십시오. AWS Secrets Manager

타입: 문자열

길이 제약: 최소 길이는 20. 최대 길이는 2,048.

패턴: arn:.*role/\S+

필수 여부: 아니요

As2Config

AS2 커넥터 객체의 파라미터를 포함하는 구조입니다.

타입: As2ConnectorConfig 객체

필수 여부: 아니요

ConnectorId

커넥터의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `c-([0-9a-f]{17})`

필수 사항 여부: Yes

LoggingRole

커넥터가 Amazon S3 이벤트에 대한 CloudWatch 로깅을 활성화할 수 있도록 하는 AWS Identity and Access Management (IAM) 역할의 Amazon 리소스 이름 (ARN) 입니다. 설정하면 로그에서 커넥터 활동을 볼 수 있습니다. CloudWatch

타입: 문자열

길이 제약: 최소 길이는 20. 최대 길이는 2,048.

패턴: `arn:.*role/\S+`

Required: No

SecurityPolicyName

커넥터의 보안 정책 이름을 지정합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 100입니다.

패턴: `TransferSFTPConnectorSecurityPolicy-[A-Za-z0-9-]+`

필수 여부: 아니요

SftpConfig

SFTP 커넥터 객체의 파라미터를 포함하는 구조입니다.

타입: SftpConnectorConfig 객체

필수 여부: 아니요

Url

파트너의 AS2 또는 SFTP 엔드포인트의 URL입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 255입니다.

필수 여부: 아니요

응답 구문

```
{
   "ConnectorId": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

ConnectorId

업데이트 중인 커넥터 객체의 식별자를 반환합니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `c-([0-9a-f]{17})`

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

AWS Transfer Family 서비스에 오류가 발생하면 이 예외가 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceExistsException

요청된 리소스가 존재하지 않거나 명령에 지정된 리전이 아닌 다른 리전에 있습니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

ThrottlingException

요청 제한 때문에 요청이 거부되었습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

UpdateHostKey

ServerId 및 HostKeyId 파라미터로 지정된 호스트 키의 설명을 업데이트합니다.

구문 요청

```
{
   "Description": "string",
   "HostKeyId": "string",
   "ServerId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

Description

호스트 키에 대한 업데이트된 설명입니다.

타입: 문자열

길이 제한: 최소 길이는 0. 최대 길이는 200입니다.

패턴: [\p{Print}]*

필수 사항 여부: Yes

HostKeyId

업데이트 중인 호스트 키의 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 25입니다.

패턴: hostkey-[0-9a-f]{17}

필수 사항 여부: Yes

ServerId

업데이트 중인 호스트 키가 들어 있는 서버의 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

필수 항목 여부: 예

응답 구문

```
{
   "HostKeyId": "string",
   "ServerId": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

HostKeyId

업데이트된 호스트 키의 호스트 키 식별자를 반환합니다.

타입: 문자열

길이 제약 조건: 고정 길이는 25입니다.

패턴: `hostkey-[0-9a-f]{17}`

ServerId

업데이트된 호스트 키가 들어 있는 서버에 대한 서버 식별자를 반환합니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

ThrottlingException

요청 제한 때문에 요청이 거부되었습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++

- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

UpdateProfile

기존 프로필의 일부 파라미터를 업데이트합니다. 업데이트할 파라미터의 새 값과 함께 업데이트하려는 프로필의 `ProfileId`를 입력합니다.

구문 요청

```
{
   "CertificateIds": [ "string" ],
   "ProfileId": "string"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

CertificateIds

가져온 인증서의 식별자 배열입니다. 프로필 및 파트너 프로필 작업에 이 식별자를 사용합니다.

타입: 문자열 배열

길이 제약 조건: 고정 길이는 22입니다.

패턴: `cert-([0-9a-f]{17})`

Required: No

ProfileId

업데이트하려는 프로필 객체의 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `p-([0-9a-f]{17})`

필수 항목 여부: 예

응답 구문

```
{
   "ProfileId": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

ProfileId

업데이트 중인 프로필의 식별자를 반환합니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `p-([0-9a-f]{17})`

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

이 예외는 AWS Transfer Family 서비스에 오류가 발생할 때 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

ThrottlingException

요청 제한 때문에 요청이 거부되었습니다.

HTTP 상태 코드: 400

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

UpdateServer

서버가 생성된 후 파일 전송 프로토콜이 활성화된 서버의 속성을 업데이트합니다.

이 `UpdateServer` 호출은 업데이트한 서버의 `ServerId`를 반환합니다.

구문 요청

```
{
   "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"
         }
      ]
   }
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

Certificate

AWS인증서 관리자 (ACM) 인증서의 아마존 리소스 이름 (ARN). `Protocols`이(가) FTPS(으)로 설정된 경우에 필요합니다.

새 퍼블릭 인증서를 요청하려면 AWS Certificate Manager 사용 설명서의 퍼블릭 인증서 요청을 참조하세요.

기존 인증서를 ACM으로 가져오려면 AWS Certificate Manager 사용 설명서의 ACM으로 인증서 가져오기를 참조하세요.

프라이빗 IP 주소를 통해 FTPS를 사용하도록 프라이빗 인증서를 요청하려면 AWS Certificate Manager 사용 설명서의 프라이빗 인증서 요청을 참조하세요.

다음 암호화 알고리즘 및 키 크기를 사용하는 인증서가 지원됩니다:

- 2048비트 RSA(RSA_2048)
- 4096비트 RSA(RSA_4096)
- 타원 프라임 곡선 256비트(EC_prime256v1)
- 타원 프라임 곡선 384 비트(EC_secp384r1)

- 타원 프라임 곡선 521비트(EC_secp521r1)

> **Note**
>
> 인증서는 FQDN 또는 IP 주소가 지정되고 발행자에 대한 정보가 있는 유효한 SSL/TLS X.509 버전 3 인증서여야 합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 1600입니다.

필수 여부: 아니요

EndpointDetails

서버에 대해 구성된 Virtual Private Cloud(VPC) 엔드포인트 설정입니다. VPC 내에서 엔드포인트를 호스팅할 때 VPC 내의 리소스에만 액세스할 수 있도록 하거나 탄력적 IP 주소를 연결하여 인터넷을 통해 클라이언트에 액세스하도록 할 수 있습니다. VPC의 기본 보안 그룹은 엔드포인트에 자동으로 할당됩니다.

타입: EndpointDetails 객체

필수 여부: 아니요

EndpointType

서버에서 사용할 엔드포인트 타입입니다. 서버의 엔드포인트를 공개적으로 액세스(PUBLIC)하거나 VPC 내부에서 호스팅하도록 선택할 수 있습니다. VPC에서 호스팅되는 엔드포인트를 사용하면 VPC 내에서만 서버 및 리소스에 대한 액세스를 제한하거나 탄력적 IP 주소를 직접 연결하여 인터넷에 연결하도록 선택할 수 있습니다.

> **Note**
>
> 2021년 5월 19일 이후, 2021년 5월 19일 이전에 계정을 아직 생성하지 않았다면 `EndpointType=VPC_ENDPOINT` AWS계정에서 사용하여 서버를 생성할 수 없습니다. 2021년 5월 19일 또는 그 이전에 AWS계정에 서버를 이미 생성한 경우 영향을 받지 않습니다. `EndpointType=VPC_ENDPOINT` 이 날짜 이후에는 `EndpointType=VPC`를 사용하세요.
> 자세한 내용은 VPC_ENDPOINT 사용 중단를 참조하세요.
> VPC를 `EndpointType`(으)로 사용하는 것이 좋습니다. 이 엔드포인트 타입을 사용하면 최대 3개의 탄력적 IPv4 주소(BYO IP 포함)를 서버 엔드포인트에 직접 연결하고 VPC 보

안 그룹을 사용하여 클라이언트의 퍼블릭 IP 주소로부터 트래픽을 제한할 수 있습니다. `EndpointType`를 `VPC_ENDPOINT`(으)로 설정하면 이 작업을 할 수 없습니다.

타입: 문자열

유효 값: `PUBLIC | VPC | VPC_ENDPOINT`

필수 여부: 아니요

HostKey

SFTP 지원 서버에 사용할 RSA, ECDSA 또는 ED25519 프라이빗 키입니다. 키를 교체하거나 다른 알고리즘을 사용하는 활성 키 집합이 있는 경우, 여러 호스트 키를 추가할 수 있습니다.

다음 명령을 사용하여 암호가 없는 RSA 2048비트 키 생성:

```
ssh-keygen -t rsa -b 2048 -N "" -m PEM -f my-new-server-key.
```

`-b` 옵션에는 최소값인 2048을 사용합니다. 3072 또는 4096을 사용하여 더 강력한 키를 만들 수 있습니다.

다음 명령을 사용하여 암호가 없는 ECDSA 256비트 키 생성:

```
ssh-keygen -t ecdsa -b 256 -N "" -m PEM -f my-new-server-key.
```

ECDSA에 대한 `-b` 옵션의 유효한 값은 256, 384, 521입니다.

다음 명령을 사용하여 암호가 없는 ED25519 키 생성:

```
ssh-keygen -t ed25519 -N "" -f my-new-server-key.
```

이 모든 명령을 원하는 my-new-server-key문자열로 바꿀 수 있습니다.

Important

기존 사용자를 기존 SFTP 지원 서버에서 새 SFTP 지원 서버로 마이그레이션할 계획이 아니라면 호스트 키를 업데이트하지 마세요. 실수로 서버의 호스트 키를 변경하면 작업에 영향을 줄 수 있습니다.

자세한 내용은 사용 설명서의 SFTP 지원 서버의 호스트 키 업데이트를 참조하십시오. AWS Transfer Family

타입: 문자열

길이 제약: 최소 길이는 0입니다. 최대 길이는 4096입니다.

필수 여부: 아니요

IdentityProviderDetails

고객의 인증 API 메서드를 호출하는 데 필요한 모든 정보를 포함하는 배열입니다.

타입: IdentityProviderDetails 객체

필수 항목 여부: 아니요

LoggingRole

서버가 Amazon S3 또는 Amazon EFSevents에 대한 아마존 CloudWatch 로깅을 활성화하도록 허용하는 AWS Identity and Access Management (IAM) 역할의 Amazon 리소스 이름 (ARN). 설정하면 로그에서 사용자 활동을 볼 수 있습니다. CloudWatch

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 2,048.

패턴: `(|arn:.*role/\S+)`

필수 여부: 아니요

PostAuthenticationLoginBanner

사용자가 서버에 연결할 때 표시할 문자열을 지정합니다. 이 문자열은 사용자가 인증한 후에 표시됩니다.

> **Note**
>
> SFTP 프로토콜은 인증 후 디스플레이 배너를 지원하지 않습니다.

타입: 문자열

길이 제약: 최소 길이는 0입니다. 최대 길이는 4096자입니다.

패턴: `[\x09-\x0D\x20-\x7E]*`

필수 여부: 아니요

PreAuthenticationLoginBanner

사용자가 서버에 연결할 때 표시할 문자열을 지정합니다. 이 문자열은 사용자가 인증하기 전에 표시됩니다. 예를 들어 다음 배너는 시스템 사용에 대한 세부 정보를 표시합니다.

```
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.
```

타입: 문자열

길이 제약: 최소 길이는 0입니다. 최대 길이는 4096자입니다.

패턴: `[\x09-\x0D\x20-\x7E]*`

필수 여부: 아니요

ProtocolDetails

서버에 대해 구성된 프로토콜 설정입니다.

- 수동 모드(FTP 및 FTPS 프로토콜의 경우)를 표시하려면 `PassiveIp` 파라미터를 사용합니다. 방화벽, 라우터 또는 로드 밸런서의 외부 IP 주소와 같은 점으로 분리된 단일 쿼드 IPv4 주소를 입력합니다.
- Amazon S3 버킷에 업로드하는 파일에 대해 클라이언트가 `SETSTAT` 명령을 사용하려 할 때 생성되는 오류를 무시하려면 `SetStatOption` 파라미터를 사용합니다. AWS Transfer Family 서버에서 `SETSTAT` 명령을 무시하고 SFTP 클라이언트를 변경할 필요 없이 파일을 업로드하도록 하려면 값을 로 설정합니다. `ENABLE_NO_OP` `SetStatOption`파라미터를 로 `ENABLE_NO_OP` 설정하면 Transfer Family가 Amazon Logs에 CloudWatch 로그 항목을 생성하여 고객이 `SETSTAT` 전화를 거는 시기를 확인할 수 있습니다.
- AWS Transfer Family 서버가 고유한 세션 ID를 통해 최근 협상된 세션을 재개할지 여부를 확인하려면 파라미터를 사용하십시오. `TlsSessionResumptionMode`
- `As2Transports`는 AS2 메시지의 전송 방법을 나타냅니다. 현재는 HTTP만 지원됩니다.

타입: ProtocolDetails 객체

필수 여부: 아니요

Protocols

파일 전송 프로토콜 클라이언트가 서버의 엔드포인트에 연결할 수 있는 파일 전송 프로토콜을 지정합니다. 사용 가능한 프로토콜은 다음과 같습니다:

- SFTP (Secure Shell(SSH) File Transfer Protocol):: SSH를 통한 파일 전송
- FTPS (File Transfer Protocol Secure): TLS 암호화를 사용한 파일 전송
- FTP (File Transfer Protocol): 암호화되지 않은 파일 전송
- AS2(적용 설명 2): 구조화된 데이터를 전송하는 데 사용됩니다. business-to-business

> **Note**
>
> - 선택하는 경우 클라이언트가 FTPS FTPS를 통해 서버에 연결할 때 서버를 식별하는 데 사용되는 ACM AWS Certificate Manager (저장된 인증서) 을 선택해야 합니다.
> - `Protocol`에 FTP 또는 FTPS이(가) 포함된 경우 `EndpointType`은(는) VPC이고 `IdentityProviderType`은(는) `AWS_DIRECTORY_SERVICE`, `AWS_LAMBDA` 또는 `API_GATEWAY`여야 합니다.
> - `Protocol`에 FTP이(가) 포함된 경우 `AddressAllocationIds`를 연결할 수 없습니다.
> - `Protocol`이 SFTP로만 설정된 경우 `EndpointType`을 PUBLIC으로 설정하고 `IdentityProviderType`을 지원되는 ID 타입(`SERVICE_MANAGED`, `AWS_DIRECTORY_SERVICE`, `AWS_LAMBDA`, `API_GATEWAY`) 중 하나로 설정할 수 있습니다.
> - `Protocol`에 AS2이(가) 포함된 경우 `EndpointType`은(는) VPC여야 하고, 도메인은 Amazon S3여야 합니다.

타입: 문자열 배열

배열 멤버: 최소수는 1개입니다. 최대 항목 수는 4개입니다.

유효 값: `SFTP | FTP | FTPS | AS2`

필수 여부: 아니요

S3StorageOptions

Amazon S3 디렉터리의 성능이 최적화되었는지 여부를 지정합니다. 이 옵션은 기본적으로 비활성화되어 있습니다.

기본적으로 홈 디렉터리 매핑에는 a가 있습니다. `TYPE DIRECTORY` 이 옵션을 활성화한 경우 매핑에 파일 대상이 포함되도록 하려면 `HomeDirectoryMapEntry Type` 를 `FILE` 명시적으로 설정해야 합니다.

유형: S3StorageOptions객체

필수 항목 여부: 아니요

SecurityPolicyName

서버의 보안 정책 이름을 지정합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 100입니다.

패턴: `Transfer[A-Za-z0-9]*SecurityPolicy-[A-Za-z0-9-]+`

Required: No

ServerId

Transfer Family 사용자가 할당된 서버 인스턴스에 대한 시스템 할당 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

필수 사항 여부: Yes

StructuredLogDestinations

사용자의 서버 로그가 전송될 로그 그룹을 지정합니다.

로그 그룹을 지정하려면 기존 로그 그룹의 ARN을 제공해야 합니다. 이 경우, 로그 그룹의 형식은 다음과 같습니다:

`arn:aws:logs:region-name:amazon-account-id:log-group:log-group-name:*`

예제: `arn:aws:logs:us-east-1:111122223333:log-group:mytestgroup:*`

이전에 서버의 로그 그룹을 지정한 경우, `update-server` 호출 시 이 파라미터에 빈 값을 제공하여 로그 그룹을 지우고 사실상 구조화된 로깅을 끌 수 있습니다. 예:

```
update-server --server-id s-1234567890abcdef0 --structured-log-
destinations
```

타입: 문자열 배열

배열 멤버: 최소 항목 수는 0개입니다. 최대 항목 수는 1개입니다.

길이 제약 조건: 최소 길이는 20입니다. 최대 길이는 1600입니다.

패턴: `arn:\S+`

Required: No

WorkflowDetails

할당할 워크플로의 워크플로 ID와 워크플로 실행에 사용되는 실행 역할을 지정합니다.

파일이 완전히 업로드될 때 실행되는 워크플로 외에도 `WorkflowDetails`에는 부분 업로드 시 실행할 워크플로의 워크플로 ID와 실행 역할이 포함될 수 있습니다. 파일이 업로드되는 동안 서버 세션 연결이 끊기면 부분 업로드가 수행됩니다.

서버에서 연결된 워크플로를 제거하려면 다음 예와 같이 빈 `OnUpload` 객체를 제공하면 됩니다.

```
aws transfer update-server --server-id s-01234567890abcdef --workflow-
details '{"OnUpload":[]}'
```

타입: WorkflowDetails 객체

필수 항목 여부: 아니요

응답 구문

```
{
   "ServerId": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터는 서비스에 의해 JSON 형식으로 반환됩니다.

ServerId

Transfer Family 사용자가 할당된 서버에 대한 시스템 할당 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

Errors

모든 작업에서 공통적으로 발생하는 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

AccessDeniedException

이 작업을 수행할 수 있는 충분한 액세스 권한이 없습니다.

HTTP 상태 코드: 400

ConflictException

VPC를 엔드포인트 타입으로 사용하고 서버 `VpcEndpointID`이(가) 가용 상태가 아닌 파일 전송 프로토콜 지원 서버에 대해 `UpdateServer`이(가) 호출되면 이 예외가 발생합니다.

HTTP 상태 코드: 400

InternalServiceError

AWS Transfer Family 서비스에 오류가 발생하면 이 예외가 발생합니다.

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceExistsException

요청된 리소스가 존재하지 않거나 명령에 지정된 리전이 아닌 다른 리전에 있습니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

ThrottlingException

요청 제한 때문에 요청이 거부되었습니다.

HTTP 상태 코드: 400

예

예

다음 예에서는 서버의 역할을 업데이트합니다.

샘플 요청

```
{
   "EndpointDetails": {
   "VpcEndpointId": "vpce-01234f056f3g13",
   "LoggingRole": "CloudWatchS3Events",
   "ServerId": "s-01234567890abcdef"
   }
}
```

예

다음 예에서는 모든 관련 워크플로를 서버에서 제거합니다.

샘플 요청

```
aws transfer update-server --server-id s-01234567890abcdef --workflow-details
 '{"OnUpload":[]}'
```

예

다음은 이 API 직접 호출에 대한 샘플 응답입니다.

샘플 응답

```
{
   "ServerId": "s-01234567890abcdef"
```

```
}
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK
- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS 파이썬용 SDK
- AWS 루비 V3용 SDK

UpdateUser

사용자에게 새 속성을 할당합니다. 전달한 파라미터는 지정한 UserName 및 ServerId의 홈 디렉터리, 역할, 정책 중 일부 또는 전부를 수정합니다.

응답은 업데이트된 사용자에 대해 ServerId 및 UserName를 반환합니다.

콘솔에서 사용자를 만들거나 업데이트할 때 제한됨을 선택할 수 있습니다. 이렇게 하면 사용자가 홈 디렉터리 외부의 어떤 항목에도 액세스할 수 없습니다. 프로그래밍 방식으로 이 동작을 구성하는 방법은 사용자를 업데이트하는 것입니다. 로 설정하고 HomeDirectoryType with LOGICAL 를 root (/) HomeDirectoryMappings Entry 로 지정하고 홈 디렉토리로 지정합니다. Target

예를 들어, 사용자의 홈 디렉터리가 /test/admin-user 인 경우 다음 명령을 실행하면 콘솔의 구성에 Restricted 플래그가 선택된 것으로 표시되도록 사용자를 업데이트합니다.

```
aws transfer update-user --server-id <server-id> --user-name admin-user --
home-directory-type LOGICAL --home-directory-mappings "[{\"Entry\":\"/\",
\"Target\":\"/test/admin-user\"}]"
```

구문 요청

```
{
   "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"
}
```

요청 파라미터

모든 작업에서 사용하는 파라미터에 대한 자세한 내용은 범용 파라미터를 참조하세요.

요청은 JSON 형식으로 다음 데이터를 받습니다.

HomeDirectory

사용자가 클라이언트를 사용하여 서버에 로그인하는 경우 사용자를 위한 랜딩 디렉터리(폴더).

HomeDirectory의 예: /bucket_name/home/mydirectory

> **Note**
>
> HomeDirectory 파라미터는 HomeDirectoryType이(가) PATH(으)로 설정된 경우에만 사용합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 1024입니다.

패턴: (|/.*)

필수 여부: 아니요

HomeDirectoryMappings

사용자에게 표시할 Amazon S3 또는 Amazon EFS 경로 및 키와 이러한 경로 및 키를 표시할 방법을 지정하는 논리적 디렉터리 매핑입니다. Entry 및 Target 쌍을 지정해야 합니다. 여기서 Entry는 경로가 표시되는 방식을 보여주고 Target는 실제 Amazon S3 경로입니다. 대상만 지정하는 경우, 그대로 표시됩니다. 또한 AWS Identity and Access Management (IAM) 역할이 경로에 대한 액세스를 제공하는지 확인해야 합니다. Target 이 값은 LOGICAL로 설정된 경우에만 HomeDirectoryType를 설정할 수 있습니다.

다음은 Entry 및 Target 쌍의 예입니다.

```
[ { "Entry": "/directory1", "Target": "/bucket_name/home/
mydirectory" } ]
```

대부분의 경우 세션 정책 대신 이 값을 사용하여 사용자를 지정된 홈 디렉터리("chroot")로 제한할 수 있습니다. 이렇게 Entry 하려면 '/'로 설정하고 Target HomeDirectory 파라미터 값으로 설정하면 됩니다.

다음은 chroot에 대한 Entry 및 Target 쌍의 예입니다.

```
[ { "Entry": "/", "Target": "/bucket_name/home/mydirectory" } ]
```

타입: HomeDirectoryMapEntry 객체 배열

어레이 멤버: 최소 항목 수 1개. 최대 항목 수는 50000개입니다.

필수 여부: 아니요

HomeDirectoryType

사용자가 서버에 로그인하는 경우 홈 디렉터리가 될 랜딩 디렉터리(폴더) 타입입니다. PATH로 설정하면, 사용자는 파일 전송 프로토콜 클라이언트에서와 같이 절대 Amazon S3 버킷 또는 EFS 경로를 볼 수 있습니다. LOGICAL로 설정하면, HomeDirectoryMappings에서 Amazon S3 또는 Amazon EFS 경로를 사용자에게 표시할 방법에 대한 매핑을 제공해야 합니다.

> **Note**
>
> HomeDirectoryType이 LOGICAL인 경우, HomeDirectoryMappings 파라미터를 사용하여 매핑을 제공해야 합니다. 반면에 HomeDirectoryType이 PATH인 경우, HomeDirectory 파라미터를 사용하여 절대 경로를 제공하세요. 템플릿에 HomeDirectory 및 HomeDirectoryMappings를 모두 포함할 수는 없습니다.

타입: 문자열

유효 값: PATH | LOGICAL

필수 여부: 아니요

Policy

여러 사용자가 동일한 AWS Identity and Access Management (IAM) 역할을 사용할 수 있도록 하기 위한 사용자 세션 정책. 이 정책은 Amazon S3 버킷의 부분에 대한 사용자의 액세스 범위를 축소합니다. 이 정책 내에서 사용할 수 있는 변수는 ${Transfer:UserName}, ${Transfer:HomeDirectory} 및 ${Transfer:HomeBucket}입니다.

> **Note**
>
> 이 정책은 ServerId의 도메인이 Amazon S3인 경우에만 적용됩니다. Amazon EFS는 세션 정책을 사용하지 않습니다.

세션 정책의 경우 정책을 정책의 Amazon 리소스 이름 (ARN) 대신 JSON Blob으로 AWS Transfer Family 저장합니다. 정책을 JSON Blob으로 저장하고 `Policy` 인수로 전달합니다. 세션 정책의 예는 세션 정책 예를 참조하세요.
자세한 내용은 AWS 보안 토큰 서비스 AssumeRoleAPI 참조를 참조하십시오.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 2048입니다.

필수 여부: 아니요

PosixProfile

Amazon Elastic File System(Amazon EFS) 파일 시스템에 대한 사용자의 액세스를 제어하는 사용자 ID(`Uid`), 그룹 ID(`Gid`) 및 보조 그룹 ID(`SecondaryGids`)를 포함한 전체 POSIX 자격 증명을 지정합니다. 파일 시스템의 파일 및 디렉터리에 설정된 POSIX 권한에 따라 Amazon EFS 파일 시스템에서 파일을 송수신할 때 사용자에게 제공되는 액세스 수준이 결정됩니다.

타입: PosixProfile 객체

필수 항목 여부: 아니요

Role

Amazon S3 버킷 또는 Amazon EFS 파일 시스템에 대한 사용자의 액세스를 제어하는 AWS Identity and Access Management (IAM) 역할의 Amazon 리소스 이름 (ARN). 이 역할에 연결된 정책은 Amazon S3 버킷 또는 Amazon EFS 파일 시스템에 대한 파일 송수신 시 사용자에게 제공할 액세스의 수준을 결정합니다. 또한 IAM 역할에는 사용자의 전송 요청을 처리할 때 서버가 해당 리소스에 액세스할 수 있도록 허용하는 신뢰 관계가 포함되어야 합니다.

타입: 문자열

길이 제약: 최소 길이는 20. 최대 길이는 2,048.

패턴: `arn:.*role/\S+`

Required: No

ServerId

사용자가 할당된 Transfer Family 서버 인스턴스에 대한 시스템 할당 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

필수 사항 여부: Yes

UserName

`ServerId`에서 지정된 바와 같이 사용자를 식별하고 서버와 연결되는 고유한 문자열입니다. 이 사용자 이름은 3~100자여야 합니다. a-z, A-Z, 0-9, _(밑줄), -(하이픈), .(마침표) 및 @ 기호를 유효한 문자로 사용할 수 있습니다. 사용자 이름은 하이픈, 마침표 및 @ 기호로 시작할 수 없습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 3입니다. 최대 길이는 100입니다.

패턴: `[\w][\w@.-]{2,99}`

필수 항목 여부: 예

응답 구문

```
{
   "ServerId": "string",
   "UserName": "string"
}
```

응답 요소

작업이 성공하면 서비스가 HTTP 200 응답을 반송합니다.

다음 데이터가 서비스에 의해 JSON 형식으로 반환됩니다.

ServerId

계정이 할당된 Transfer Family 서버 인스턴스에 대한 시스템 할당 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

UserName

요청에 지정된 서버 인스턴스에 할당된 사용자의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 3입니다. 최대 길이는 100입니다.

패턴: `[\w][\w@.-]{2,99}`

Errors

모든 작업에서 발생하는 일반적인 오류에 대한 자세한 내용은 일반적인 오류 섹션을 참조하세요.

InternalServiceError

서비스에 오류가 발생하면 이 예외가 발생합니다. AWS Transfer Family

HTTP 상태 코드: 500

InvalidRequestException

클라이언트가 잘못된 형식의 요청을 제출하면 이 예외가 발생합니다.

HTTP 상태 코드: 400

ResourceNotFoundException

AWS Transfer Family 서비스에서 리소스를 찾을 수 없는 경우 이 예외가 발생합니다.

HTTP 상태 코드: 400

ServiceUnavailableException

AWS Transfer Family 서비스를 이용할 수 없어 요청이 실패했습니다.

HTTP 상태 코드: 500

ThrottlingException

요청 제한 때문에 요청이 거부되었습니다.

HTTP 상태 코드: 400

예

예

다음 예에서는 Transfer Family 사용자를 업데이트합니다.

샘플 요청

```
{
   "HomeDirectory": "/bucket2/documentation",
   "HomeDirectoryMappings": [
      {
         "Entry": "/directory1",
         "Target": "/bucket_name/home/mydirectory"
      }
   ],
   "HomeDirectoryType:" "PATH",
   "Role": "AssumeRole",
   "ServerId": "s-01234567890abcdef",
   "UserName": "my_user"
}
```

예

다음은 이 API 직접 호출에 대한 샘플 응답입니다.

샘플 응답

```
{
   "ServerId": "s-01234567890abcdef",
   "UserName": "my_user"
}
```

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS Go v2를 위한 SDK

- AWS Java V2용 SDK
- AWS V3용 SDK JavaScript
- AWS PHP V3용 SDK
- AWS Python용 SDK
- AWS 루비 V3용 SDK

데이터 유형

다음 데이터 유형이 지원됩니다.

- 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

AS2 커넥터 개체에 대한 세부 정보가 들어 있습니다. 커넥터 객체는 AWS Transfer Family 고객과 거래 파트너를 연결하는 AS2 아웃바운드 프로세스에 사용됩니다.

내용

BasicAuthSecretId

AS2 커넥터 API에 대한 기본 인증 지원을 제공합니다. 기본 인증을 사용하려면 AWS Secrets Manager의 암호의 이름이나 Amazon 리소스 이름(ARN)을 제공해야 합니다.

이 파라미터의 기본값은 `null`이며, 커넥터에 대해 기본 인증을 사용할 수 없음을 나타냅니다.

커넥터에서 기본 인증을 사용해야 하는 경우 암호는 다음 형식이어야 합니다.

```
{ "Username": "user-name", "Password": "user-password" }
```

`user-name` 및 `user-password`를 인증 중인 실제 사용자의 보안 인증 정보로 바꾸세요.

유념할 사항:

- 이러한 보안 인증 정보를 이 API에 직접 전달하지 않고 Secrets Manager에 저장하고 있습니다.
- API, SDK를 사용하거나 커넥터를 구성하는 경우 기본 인증을 활성화하기 전에 암호를 생성해야 합니다. CloudFormation 하지만 AWS 관리 콘솔을 사용하는 경우 시스템에서 암호를 자동으로 생성하도록 할 수 있습니다.

이전에 커넥터에 대한 기본 인증을 활성화한 경우 `UpdateConnector` API 직접 호출을 사용하여 그 기본 인증을 비활성화할 수 있습니다. 예를 들어 사용자가 CLI를 사용하는 경우, 다음 명령을 실행하여 기본 인증을 제거할 수 있습니다.

```
update-connector --connector-id my-connector-id --as2-config
'BasicAuthSecretId=""'
```

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 2048입니다.

필수 여부: 아니요

Compression

AS2 파일이 압축되었는지의 여부를 지정합니다.

타입: 문자열

유효 값: `ZLIB | DISABLED`

필수 여부: 아니요

EncryptionAlgorithm

알고리즘은 파일을 암호화하는 데 사용됩니다.

유의할 사항:

- 이 알고리즘은 취약한 암호화 DES_EDE3_CBC 알고리즘이므로 이 알고리즘이 필요한 레거시 클라이언트를 지원해야 하는 경우가 아니면 사용하지 마십시오.
- 커넥터의 URL이 HTTPS를 사용하는지 여부만 NONE지정할 수 있습니다. HTTPS를 사용하면 트래픽이 일반 텍스트로 전송되지 않습니다.

타입: 문자열

유효 값: `AES128_CBC | AES192_CBC | AES256_CBC | DES_EDE3_CBC | NONE`

필수 여부: 아니요

LocalProfileId

AS2 로컬 프로필의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `p-([0-9a-f]{17})`

Required: No

MdnResponse

전송에 대한 파트너 응답이 AWS Transfer Family 동기식인지 비동기식인지 확인하기 위한 아웃바운드 요청 (서버에서 파트너 AS2 서버로의) 에 사용됩니다. 다음 값 중 하나를 지정합니다.

- SYNC: 시스템은 파일이 성공적으로 전송되었는지(혹은 아닌지) 여부를 확인하는 동기식 MDN 응답을 예상합니다.
- NONE:MDN 응답이 필요하지 않은 것을 지정합니다.

타입: 문자열

유효 값: SYNC | NONE

필수 여부: 아니요

MdnSigningAlgorithm

MDN 응답의 서명 알고리즘.

> **Note**
>
> DEFAULT로 설정된 경우(또는 전혀 설정하지 않은 경우) SigningAlgorithm에 대한 값이 사용됩니다.

타입: 문자열

유효 값: SHA256 | SHA384 | SHA512 | SHA1 | NONE | DEFAULT

필수 여부: 아니요

MessageSubject

커넥터와 함께 전송되는 AS2 메시지의 Subject HTTP 헤더 속성으로 사용됩니다.

타입: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: [\p{Print}\p{Blank}]+

Required: No

PartnerProfileId

커넥터의 파트너 프로필의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: p-([0-9a-f]{17})

Required: No

SigningAlgorithm

커넥터와 함께 전송된 AS2 메시지에 서명하는 데 사용되는 알고리즘입니다.

타입: 문자열

유효 값: `SHA256 | SHA384 | SHA512 | SHA1 | NONE`

필수 여부: 아니요

참고

AWS 언어별 SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

CopyStepDetails

각 단계 타입에는 자체의 StepDetails 구조가 있습니다.

내용

DestinationFileLocation

복사할 파일의 위치를 지정합니다. 이 필드에서 ${Transfer:UserName} 또는 ${Transfer:UploadDate}를 사용하여 사용자 이름 또는 업로드 날짜를 기준으로 대상 접두사를 파라미터화할 수 있습니다.

- 파일을 업로드한 Transfer Family 사용자의 명칭이 접두사로 붙은 Amazon S3 버킷에 업로드된 파일을 복사하려면 DestinationFileLocation의 값을 ${Transfer:UserName}(으)로 설정합니다.
- 업로드 날짜 접두사가 붙은 Amazon S3 버킷에 업로드된 파일을 복사하려면 DestinationFileLocation의 값을 ${Transfer:UploadDate}(으)로 설정합니다.

> **Note**
>
> 시스템은 파일이 UTC로 업로드된 날짜를 기준으로 UploadDate를 YYYY-MM-DD의 날짜 형식으로 변환합니다.

타입: InputFileLocation 객체

필수 여부: 아니요

Name

단계의 명칭, 식별자로 사용됨.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 30입니다.

패턴: [\w-]*

필수 여부: 아니요

OverwriteExisting

같은 명칭의 기존 파일을 덮어쓸지 여부를 나타내는 플래그입니다. 기본값은 FALSE입니다.

워크플로에서 기존 파일과 명칭이 같은 파일을 처리하는 경우, 동작은 다음과 같습니다.

- `OverwriteExisting`이(가) `TRUE`인 경우, 기존 파일이 처리 중인 파일로 대체됩니다.
- `OverwriteExisting`이(가) `FALSE`인 경우, 아무 일도 일어나지 않고 워크플로 처리가 중지됩니다.

타입: 문자열

유효 값: `TRUE | FALSE`

필수 여부: 아니요

SourceFileLocation

워크플로 단계의 입력으로 사용할 파일(이전 단계의 출력 또는 워크플로에 처음 업로드한 파일)을 지정합니다.

- 이전 파일을 입력으로 사용하려면 `${previous.file}`를 입력합니다. 이 경우 이 워크플로 단계에서는 이전 워크플로 단계의 출력 파일을 입력으로 사용합니다. 이것이 기본값입니다.
- 시초로 업로드된 파일 위치를 이 단계의 입력으로 사용하려면 `${original.file}`를 입력합니다.

타입: 문자열

길이 제약: 최소 길이는 0. 최대 길이는 256입니다.

패턴: `\$\{(\w+.)+\w+\}`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

CustomStepDetails

각 단계 타입에는 자체의 `StepDetails` 구조가 있습니다.

내용

Name

단계의 명칭, 식별자로 사용됨.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 30입니다.

패턴: `[\w-]*`

필수 여부: 아니요

SourceFileLocation

워크플로 단계의 입력으로 사용할 파일(이전 단계의 출력 또는 워크플로에 처음 업로드한 파일)을 지정합니다.

- 이전 파일을 입력으로 사용하려면 `${previous.file}`를 입력합니다. 이 경우 이 워크플로 단계에서는 이전 워크플로 단계의 출력 파일을 입력으로 사용합니다. 이것이 기본값입니다.
- 시초로 업로드된 파일 위치를 이 단계의 입력으로 사용하려면 `${original.file}`를 입력합니다.

타입: 문자열

길이 제약: 최소 길이는 0. 최대 길이는 256입니다.

패턴: `\$\{(\w+.)+\w+\}`

필수 여부: 아니요

Target

호출되는 Lambda 함수의 ARN입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 170입니다.

패턴: `arn:[a-z-]+:lambda:.*`

필수 여부: 아니요

TimeoutSeconds

단계에 대한 시간 제한 (단위: 초).

타입: 정수

유효한 범위: 최소값은 1입니다. 최대값은 1800입니다.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

DecryptStepDetails

각 단계 타입에는 자체의 `StepDetails` 구조가 있습니다.

내용

DestinationFileLocation

해독되는 파일의 위치를 지정합니다. 이 필드에서 `${Transfer:UserName}` 또는 `${Transfer:UploadDate}`를 사용하여 사용자 이름 또는 업로드 날짜를 기준으로 대상 접두사를 파라미터화할 수 있습니다.

- 파일을 업로드한 Transfer Family 사용자의 명칭이 접두사로 붙은 Amazon S3 버킷에 업로드된 파일을 복호화하려면 `DestinationFileLocation`의 값을 `${Transfer:UserName}`으(로) 설정합니다.
- 업로드 날짜가 접두사가 붙은 Amazon S3 버킷에 업로드된 파일을 복호화하려면 `DestinationFileLocation`의 값을 `${Transfer:UploadDate}`으(로) 설정합니다.

> **Note**
>
> 시스템은 파일이 UTC로 업로드된 날짜를 기준으로 `UploadDate`를 YYYY-MM-DD의 날짜 형식으로 변환합니다.

타입: InputFileLocation 객체

필수 여부: 예

Type

사용된 암호화 타입. 현재, 이 값은 `PGP`이어야 합니다.

타입: 문자열

유효 값: `PGP`

필수 여부: 예

Name

단계의 명칭, 식별자로 사용됨.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 30입니다.

패턴: `[\w-]*`

필수 여부: 아니요

OverwriteExisting

같은 명칭의 기존 파일을 덮어쓸지 여부를 나타내는 플래그입니다. 기본값은 FALSE입니다.

워크플로에서 기존 파일과 명칭이 같은 파일을 처리하는 경우, 동작은 다음과 같습니다.

- `OverwriteExisting`이(가) `TRUE`인 경우, 기존 파일이 처리 중인 파일로 대체됩니다.
- `OverwriteExisting`이(가) `FALSE`인 경우, 아무 일도 일어나지 않고 워크플로 처리가 중지됩니다.

타입: 문자열

유효 값: `TRUE | FALSE`

필수 여부: 아니요

SourceFileLocation

워크플로 단계의 입력으로 사용할 파일(이전 단계의 출력 또는 워크플로에 처음 업로드한 파일)을 지정합니다.

- 이전 파일을 입력으로 사용하려면 `${previous.file}`를 입력합니다. 이 경우 이 워크플로 단계에서는 이전 워크플로 단계의 출력 파일을 입력으로 사용합니다. 이것이 기본값입니다.
- 시초로 업로드된 파일 위치를 이 단계의 입력으로 사용하려면 `${original.file}`를 입력합니다.

타입: 문자열

길이 제약: 최소 길이는 0. 최대 길이는 256입니다.

패턴: `\$\{(\w+.)+\w+\}`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

DeleteStepDetails

단계의 명칭, 삭제 단계를 나타내는 데 사용됨.

내용

Name

단계의 명칭, 식별자로 사용됨.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 30입니다.

패턴: `[\w-]*`

필수 여부: 아니요

SourceFileLocation

워크플로 단계의 입력으로 사용할 파일(이전 단계의 출력 또는 워크플로에 처음 업로드한 파일)을 지정합니다.

- 이전 파일을 입력으로 사용하려면 `${previous.file}`를 입력합니다. 이 경우 이 워크플로 단계에서는 이전 워크플로 단계의 출력 파일을 입력으로 사용합니다. 이것이 기본값입니다.
- 시초로 업로드된 파일 위치를 이 단계의 입력으로 사용하려면 `${original.file}`를 입력합니다.

타입: 문자열

길이 제약: 최소 길이는 0. 최대 길이는 256입니다.

패턴: `\$\{(\w+.)+\w+\}`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK

- AWS 루비 V3용 SDK

DescribedAccess

지정된 액세스 속성을 설명합니다.

내용

ExternalId

디렉터리 내의 특정 그룹을 식별하는 데 필요한 고유 식별자입니다. 연결하는 그룹의 사용자는 사용하는 활성화된 프로토콜을 통해 Amazon S3 또는 Amazon EFS 리소스에 액세스할 수 AWS Transfer Family있습니다. 그룹 이름을 아는 경우 PowerShell Windows를 사용하여 다음 명령을 실행하여 SID 값을 볼 수 있습니다.

```
Get-ADGroup -Filter {samAccountName -like "YourGroupName*"} -Properties
* | Select SamAccountName,ObjectSid
```

이 명령에서 Active Directory 그룹의 YourGroupName이름으로 바꾸십시오.

이 파라미터를 확인하는 데 사용되는 정규 표현식은 공백 없이 대문자 및 소문자 영숫자로 구성된 문자의 문자열입니다. 밑줄이나 또한 =,.@:/- 문자도 포함할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: `S-1-[\d-]+`

필수 여부: 아니요

HomeDirectory

사용자가 클라이언트를 사용하여 서버에 로그인하는 경우 사용자를 위한 랜딩 디렉터리(폴더).

`HomeDirectory`의 예: `/bucket_name/home/mydirectory`

> **Note**
>
> `HomeDirectory` 파라미터는 `HomeDirectoryType`이(가) `PATH`(으)로 설정된 경우에만 사용합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 1024입니다.

패턴: `(|/.*)`

필수 여부: 아니요

HomeDirectoryMappings

사용자에게 표시할 Amazon S3 또는 Amazon EFS 경로 및 키와 이러한 경로 및 키를 표시할 방법을 지정하는 논리적 디렉터리 매핑입니다. `Entry` 및 `Target` 쌍을 지정해야 합니다. 여기서 `Entry`는 경로가 표시되는 방식을 보여주고 `Target`는 실제 Amazon S3 경로입니다. 대상만 지정하는 경우, 그대로 표시됩니다. 또한 AWS Identity and Access Management (IAM) 역할이 경로에 대한 액세스를 제공하는지 확인해야 합니다. `Target` 이 값은 LOGICAL로 설정된 경우에만 `HomeDirectoryType`를 설정할 수 있습니다.

대부분의 경우 세션 정책 대신 이 값을 사용하여 사용자를 지정된 홈 디렉터리("chroot")의 연계된 액세스로 제약할 수 있습니다. 이렇게 하려면 `Entry`를 '/'로 설정하고, `Target`를 `HomeDirectory` 파라미터 값으로 설정하면 됩니다.

타입: HomeDirectoryMapEntry 객체 배열

어레이 멤버: 최소 항목 수 1개. 최대 항목 수는 50000개입니다.

필수 여부: 아니요

HomeDirectoryType

사용자가 서버에 로그인하는 경우 홈 디렉터리가 될 랜딩 디렉터리(폴더) 타입입니다. `PATH`로 설정하면, 사용자는 파일 전송 프로토콜 클라이언트에서와 같이 절대 Amazon S3 버킷 또는 EFS 경로를 볼 수 있습니다. `LOGICAL`로 설정하면, `HomeDirectoryMappings`에서 Amazon S3 또는 Amazon EFS 경로를 사용자에게 표시할 방법에 대한 매핑을 제공해야 합니다.

> **Note**
>
> `HomeDirectoryType`이 `LOGICAL`인 경우, `HomeDirectoryMappings` 파라미터를 사용하여 매핑을 제공해야 합니다. 반면에 `HomeDirectoryType`이 `PATH`인 경우, `HomeDirectory` 파라미터를 사용하여 절대 경로를 제공하세요. 템플릿에 `HomeDirectory` 및 `HomeDirectoryMappings`를 모두 포함할 수는 없습니다.

타입: 문자열

유효 값: `PATH | LOGICAL`

필수 여부: 아니요

Policy

여러 사용자가 동일한 AWS Identity and Access Management (IAM) 역할을 사용할 수 있도록 하기 위한 사용자 세션 정책. 이 정책은 Amazon S3 버킷의 부분에 대한 사용자의 액세스 범위를 축소합니다. 이 정책 내에서 사용할 수 있는 변수는 `${Transfer:UserName}`, `${Transfer:HomeDirectory}` 및 `${Transfer:HomeBucket}`입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 2048입니다.

필수 여부: 아니요

PosixProfile

Amazon EFS 파일 시스템에 대한 사용자의 액세스를 통제하는 사용자 ID(`Uid`), 그룹 ID(`Gid`) 및 보조 그룹 ID(`SecondaryGids`)를 포함한 전체 POSIX 자격 증명입니다. 파일 시스템의 파일 및 디렉터리에 설정된 POSIX 권한에 따라 Amazon EFS 파일 시스템에서 파일을 송수신할 때 사용자에게 제공되는 액세스 수준이 결정됩니다.

타입: PosixProfile 객체

필수 항목 여부: 아니요

Role

Amazon S3 버킷 또는 Amazon EFS 파일 시스템에 대한 사용자의 액세스를 제어하는 AWS Identity and Access Management (IAM) 역할의 Amazon 리소스 이름 (ARN). 이 역할에 연결된 정책은 Amazon S3 버킷 또는 Amazon EFS 파일 시스템에 대한 파일 송수신 시 사용자에게 제공할 액세스의 수준을 결정합니다. 또한 IAM 역할에는 사용자의 전송 요청을 처리할 때 서버가 해당 리소스에 액세스할 수 있도록 허용하는 신뢰 관계가 포함되어야 합니다.

타입: 문자열

길이 제약: 최소 길이는 20. 최대 길이는 2,048.

패턴: `arn:.*role/\S+`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

DescribedAgreement

계약의 속성을 설명합니다.

내용

Arn

계약의 고유한 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 20입니다. 최대 길이는 1600입니다.

패턴: `arn:\S+`

필수 여부: 예

AccessRole

커넥터는 AS2 또는 SFTP 프로토콜을 사용하여 파일을 전송하는 데 사용됩니다. 액세스 역할에는 사용할 AWS Identity and Access Management 역할의 Amazon 리소스 이름 (ARN) 을 제공합니다.

AS2 커넥터의 경우

AS2를 사용하여 `StartFileTransfer`를 호출하고 요청 파라미터인 `SendFilePaths`에 파일 경로를 지정하여 파일을 전송할 수 있습니다. 파일의 상위 디렉터리(예: `--send-file-paths /bucket/dir/file.txt`의 경우 상위 디렉터리는 `/bucket/dir/`임)를 사용하여 처리된 AS2 메시지 파일을 임시로 저장하고, 파트너로부터 수신 시 MDN을 저장하고, 전송의 관련 메타데이터를 포함하는 최종 JSON 파일을 작성합니다. 따라서 `AccessRole`은(는) `StartFileTransfer` 요청에 사용된 파일 위치의 상위 디렉터리에 대한 읽기 및 쓰기 액세스를 제공해야 합니다. 또한 `StartFileTransfer`와(과) 함께 전송하려는 파일의 상위 디렉터리에 대한 읽기 및 쓰기 액세스 권한을 제공해야 합니다.

AS2 커넥터에 기본 인증을 사용하는 경우 액세스 역할에는 암호에 대한 `secretsmanager:GetSecretValue` 권한이 필요합니다. Secrets Manager에서 관리 키 대신 고객 AWS 관리 키를 사용하여 암호를 암호화하는 경우 역할에도 해당 키에 대한 `kms:Decrypt` 권한이 필요합니다.

SFTP 커넥터의 경우

액세스 역할이 StartFileTransfer 요청에서 사용된 파일 위치의 상위 디렉터리에 대한 읽기 및 쓰기 액세스 권한을 제공하는지 확인하세요. 또한 역할이 secretsmanager:GetSecretValue 권한을 제공하는지 확인하십시오. AWS Secrets Manager

타입: 문자열

길이 제약: 최소 길이는 20. 최대 길이는 2,048.

패턴: arn:.*role/\S+

필수 여부: 아니요

AgreementId

계약의 고유 식별자입니다. 계약을 생성하면 이 식별자가 반환됩니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: a-([0-9a-f]{17})

필수 여부: 아니요

BaseDirectory

AS2 프로토콜을 사용하여 전송되는 파일의 랜딩 디렉터리(폴더)입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 1024입니다.

패턴: (|/.*)

필수 여부: 아니요

Description

계약을 식별하는 데 사용되는 명칭 또는 간단한 설명입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 200입니다.

패턴: [\p{Graph}]+

필수 여부: 아니요

LocalProfileId

AS2 로컬 프로필의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `p-([0-9a-f]{17})`

필수 여부: 아니요

PartnerProfileId

계약에 사용되는 파트너 프로필의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `p-([0-9a-f]{17})`

필수 여부: 아니요

ServerId

서버 인스턴스에 대해 시스템에서 할당한 고유 식별자입니다. 이 식별자는 계약에서 사용하는 특정 서버를 나타냅니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

필수 여부: 아니요

Status

계약의 현재 상태는 `ACTIVE` 또는 `INACTIVE`입니다.

타입: 문자열

유효 값: `ACTIVE | INACTIVE`

필수 여부: 아니요

Tags

계약을 그룹화하고 검색하는 데 사용할 수 있는 키-값 쌍입니다.

타입: Tag 객체 배열

배열 멤버: 최소 항목 수는 1개입니다. 최대 항목 수는 50개입니다.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

DescribedCertificate

인증서의 속성을 설명합니다.

내용

Arn

인증서에 대한 고유한 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 20입니다. 최대 길이는 1600입니다.

패턴: `arn:\S+`

필수 여부: 예

ActiveDate

인증서가 활성화되는 시기를 지정하는 선택적 날짜입니다.

타입: Timestamp

필수 여부: 아니요

Certificate

인증서의 파일 명칭입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 16384입니다.

패턴: `[\u0009\u000A\u000D\u0020-\u00FF]*`

필수 여부: 아니요

CertificateChain

인증서의 체인을 구성하는 인증서의 목록입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 2097152입니다.

패턴: `[\u0009\u000A\u000D\u0020-\u00FF]*`

필수 여부: 아니요

CertificateId

가져온 인증서의 식별자 배열입니다. 프로필 및 파트너 프로필 작업에 이 식별자를 사용합니다.

타입: 문자열

길이 제약 조건: 고정 길이는 22입니다.

패턴: `cert-([0-9a-f]{17})`

필수 여부: 아니요

Description

인증서를 식별하는 데 사용되는 명칭 또는 설명입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 200입니다.

패턴: `[\p{Graph}]+`

필수 여부: 아니요

InactiveDate

인증서가 비활성화되는 시기를 지정하는 선택적 날짜.

타입: Timestamp

필수 여부: 아니요

NotAfterDate

인증서가 유효한 최종 날짜입니다.

타입: Timestamp

필수 여부: 아니요

NotBeforeDate

인증서가 유효한 가장 이른 날짜입니다.

타입: Timestamp

필수 여부: 아니요

Serial

인증서의 일련 번호입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 48입니다.

패턴: `[\p{XDigit}{2}:?]*`

필수 여부: 아니요

Status

인증서는 ACTIVE, PENDING_ROTATION 또는 INACTIVE일 수 있습니다. PENDING_ROTATION은(는) 이 인증서가 만료되면 현재 인증서를 대체함을 의미합니다.

타입: 문자열

유효 값: `ACTIVE | PENDING_ROTATION | INACTIVE`

필수 여부: 아니요

Tags

인증서를 그룹화하고 검색하는 데 사용할 수 있는 키-값 쌍입니다.

타입: Tag 객체 배열

배열 멤버: 최소 항목 수는 1개입니다. 최대 항목 수는 50개입니다.

필수 여부: 아니요

Type

인증서에 대해 프라이빗 키가 지정된 경우 해당 타입은 CERTIFICATE_WITH_PRIVATE_KEY입니다. 프라이빗 키가 없는 경우 타입은 CERTIFICATE입니다.

타입: 문자열

유효 값: `CERTIFICATE | CERTIFICATE_WITH_PRIVATE_KEY`

필수 여부: 아니요

Usage

이 인증서를 사용하는 방법을 지정합니다. 다음과 같은 방법으로 사용할 수 있습니다.

- `SIGNING`: AS2 메시지 서명용
- `ENCRYPTION`: AS2 메시지 암호화용
- `TLS`: HTTPS를 통해 전송되는 AS2 통신을 보호하는 데 사용됩니다.

타입: 문자열

유효 값: `SIGNING | ENCRYPTION`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

DescribedConnector

ConnectorId(으)로 식별되는 커넥터의 파라미터를 설명합니다.

내용

Arn

커넥터의 고유한 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 20입니다. 최대 길이는 1600입니다.

패턴: arn:\S+

필수 여부: 예

AccessRole

커넥터는 AS2 또는 SFTP 프로토콜을 사용하여 파일을 전송하는 데 사용됩니다. 액세스 역할에는 사용할 AWS Identity and Access Management 역할의 Amazon 리소스 이름 (ARN) 을 제공합니다.

AS2 커넥터의 경우

AS2를 사용하여 StartFileTransfer를 호출하고 요청 파라미터인 SendFilePaths에 파일 경로를 지정하여 파일을 전송할 수 있습니다. 파일의 상위 디렉터리(예: --send-file-paths /bucket/dir/file.txt의 경우 상위 디렉터리는 /bucket/dir/임)를 사용하여 처리된 AS2 메시지 파일을 임시로 저장하고, 파트너로부터 수신 시 MDN을 저장하고, 전송의 관련 메타데이터를 포함하는 최종 JSON 파일을 작성합니다. 따라서 AccessRole은(는) StartFileTransfer 요청에 사용된 파일 위치의 상위 디렉터리에 대한 읽기 및 쓰기 액세스를 제공해야 합니다. 또한 StartFileTransfer와(과) 함께 전송하려는 파일의 상위 디렉터리에 대한 읽기 및 쓰기 액세스 권한을 제공해야 합니다.

AS2 커넥터에 기본 인증을 사용하는 경우 액세스 역할에는 암호에 대한 secretsmanager:GetSecretValue 권한이 필요합니다. Secrets Manager에서 관리 키 대신 고객 AWS 관리 키를 사용하여 암호를 암호화하는 경우 역할에도 해당 키에 대한 kms:Decrypt 권한이 필요합니다.

SFTP 커넥터의 경우

액세스 역할이 StartFileTransfer 요청에서 사용된 파일 위치의 상위 디렉터리에 대한 읽기 및 쓰기 액세스 권한을 제공하는지 확인하세요. 또한 역할이 secretsmanager:GetSecretValue 권한을 제공하는지 확인하십시오. AWS Secrets Manager

타입: 문자열

길이 제약: 최소 길이는 20. 최대 길이는 2,048.

패턴: arn:.*role/\S+

필수 여부: 아니요

As2Config

AS2 커넥터 객체의 파라미터를 포함하는 구조입니다.

타입: As2ConnectorConfig 객체

필수 여부: 아니요

ConnectorId

커넥터의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: c-([0-9a-f]{17})

Required: No

LoggingRole

커넥터가 Amazon S3 이벤트에 대한 CloudWatch 로깅을 활성화할 수 있도록 하는 AWS Identity and Access Management (IAM) 역할의 Amazon 리소스 이름 (ARN) 입니다. 설정하면 로그에서 커넥터 활동을 볼 수 있습니다. CloudWatch

타입: 문자열

길이 제약: 최소 길이는 20. 최대 길이는 2,048.

패턴: arn:.*role/\S+

Required: No

SecurityPolicyName

지정된 커넥터에 대한 보안 정책의 텍스트 이름.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 100입니다.

패턴: `TransferSFTPConnectorSecurityPolicy-[A-Za-z0-9-]+`

Required: No

ServiceManagedEgressIpAddresses

이 커넥터의 송신 IP 주소 목록. 이러한 IP 주소는 커넥터를 생성할 때 자동으로 할당됩니다.

유형: 문자열 어레이

패턴: `\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}`

필수 여부: 아니요

SftpConfig

SFTP 커넥터 객체의 파라미터를 포함하는 구조입니다.

타입: SftpConnectorConfig 객체

필수 여부: 아니요

Tags

커넥터를 그룹화하고 검색하는 데 사용할 수 있는 키-값 쌍.

타입: Tag 객체 배열

배열 멤버: 최소 항목 수는 1개입니다. 최대 항목 수는 50개입니다.

필수 여부: 아니요

Url

파트너의 AS2 또는 SFTP 엔드포인트의 URL입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 255입니다.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

DescribedExecution

실행 객체에 대한 세부 정보입니다.

내용

ExecutionId

워크플로 실행의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 36입니다.

패턴: `[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}`

Required: No

ExecutionRole

실행과 관련된 IAM 역할.

타입: 문자열

길이 제약: 최소 길이는 20. 최대 길이는 2,048.

패턴: `arn:.*role/\S+`

Required: No

InitialFileLocation

Amazon S3 또는 EFS 파일 위치를 설명하는 구조입니다. 실행이 시작될 때의 파일 위치입니다. 파일이 복사되는 경우 이 위치는 대상이 아닌 초기 파일 위치입니다.

타입: FileLocation 객체

필수 항목 여부: 아니요

LoggingConfiguration

실행과 관련된 IAM 로깅 역할입니다.

타입: LoggingConfiguration 객체

필수 항목 여부: 아니요

PosixProfile

Amazon EFS 파일 시스템에 대한 사용자의 액세스를 통제하는 사용자 ID(Uid), 그룹 ID(Gid) 및 보조 그룹 ID(SecondaryGids)를 포함한 전체 POSIX 자격 증명입니다. 파일 시스템의 파일 및 디렉터리에 설정된 POSIX 권한에 따라 Amazon EFS 파일 시스템에서 파일을 송수신할 때 사용자에게 제공되는 액세스 수준이 결정됩니다.

타입: PosixProfile 객체

필수 항목 여부: 아니요

Results

실행 결과를 설명하는 구조입니다. 여기에는 각 단계의 세부 정보, 오류 타입 및 메시지(있는 경우), OnExceptionSteps 구조와 함께 단계 목록이 포함됩니다.

타입: ExecutionResults 객체

필수 항목 여부: 아니요

ServiceMetadata

워크플로와 관련된 세션 세부 정보를 위한 컨테이너 객체입니다.

타입: ServiceMetadata 객체

필수 항목 여부: 아니요

Status

상태는 실행 중 하나입니다. 진행 중이거나, 완료되었거나, 예외가 발생했거나, 예외를 처리 중일 수 있습니다.

타입: 문자열

유효 값: IN_PROGRESS | COMPLETED | EXCEPTION | HANDLING_EXCEPTION

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

DescribedHostKey

서버 호스트 키에 대한 세부 정보입니다.

내용

Arn

호스트 키에 대한 고유한 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 20입니다. 최대 길이는 1600입니다.

패턴: `arn:\S+`

필수 사항 여부: Yes

DateImported

호스트 키가 서버에 추가된 날짜입니다.

타입: Timestamp

필수 여부: 아니요

Description

이 호스트 키에 대한 텍스트 설명입니다.

타입: 문자열

길이 제한: 최소 길이는 0. 최대 길이는 200입니다.

패턴: `[\p{Print}]*`

Required: No

HostKeyFingerprint

퍼블릭 키 핑거프린트는 더 긴 퍼블릭 키를 식별하는 데 사용되는 짧은 시퀀스의 바이트입니다.

타입: 문자열

필수사항: 아니요

HostKeyId

호스트 키의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 25입니다.

패턴: `hostkey-[0-9a-f]{17}`

Required: No

Tags

그룹화 및 호스트 키 검색에 사용될 수 있는 키-값 쌍입니다.

타입: Tag 객체 배열

배열 멤버: 최소 항목 수는 1개입니다. 최대 항목 수는 50개입니다.

필수 여부: 아니요

Type

호스트 키에 사용되는 암호화 알고리즘입니다. Type 파라미터는 다음 값 중 하나를 사용하여 지정합니다.

- `ssh-rsa`
- `ssh-ed25519`
- `ecdsa-sha2-nistp256`
- `ecdsa-sha2-nistp384`
- `ecdsa-sha2-nistp521`

타입: 문자열

필수 항목 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++

- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

DescribedProfile

로컬 또는 파트너 AS2 프로필의 세부 정보.

내용

Arn

프로필의 고유한 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 20입니다. 최대 길이는 1600입니다.

패턴: `arn:\S+`

필수 사항 여부: Yes

As2Id

`As2Id`은(는) RFC 4130에 정의된 AS2-name입니다. 인바운드 전송의 경우 파트너로부터 전송된 AS2 메시지의 `AS2-From` 헤더입니다. 아웃바운드 커넥터의 경우 `StartFileTransfer` API 작업을 사용하여 파트너에게 전송된 AS2 메시지의 `AS2-To` 헤더입니다. 이 ID는 공백을 포함할 수 없습니다.

타입: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: `[\p{Print}\s]*`

필수 여부: 아니요

CertificateIds

가져온 인증서의 식별자 배열입니다. 프로필 및 파트너 프로필 작업에 이 식별자를 사용합니다.

타입: 문자열 배열

길이 제약 조건: 고정 길이는 22입니다.

패턴: `cert-([0-9a-f]{17})`

Required: No

ProfileId

로컬 또는 파트너 AS2 프로필의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `p-([0-9a-f]{17})`

Required: No

ProfileType

LOCAL 타입 프로필만 나열할지 아니면 PARTNER 타입 프로필만 나열할지를 나타냅니다. 요청에 입력되지 않은 경우 명령은 모든 타입의 프로필을 나열합니다.

타입: 문자열

유효 값: `LOCAL | PARTNER`

필수 여부: 아니요

Tags

프로필을 그룹화하고 검색하는 데 사용할 수 있는 키-값 쌍입니다.

타입: Tag 객체 배열

배열 멤버: 최소 항목 수는 1개입니다. 최대 항목 수는 50개입니다.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

DescribedSecurityPolicy

지정한 보안 정책의 속성을 설명합니다. 보안 정책에 대한 자세한 내용은 서버의 보안 정책 사용 또는 SFTP 커넥터의 보안 정책 사용을 참조하십시오.

내용

SecurityPolicyName

지정된 보안 정책의 텍스트 이름.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 100입니다.

패턴: `Transfer[A-Za-z0-9]*SecurityPolicy-[A-Za-z0-9-]+`

필수 사항 여부: Yes

Fips

이 정책이 Federal Information Processing Standards(FIPS)를 사용하는지 여부를 지정합니다. 이 매개 변수는 서버 및 커넥터 보안 정책 모두에 적용됩니다.

타입: 부울

필수 항목 여부: 아니요

Protocols

보안 정책이 적용되는 파일 전송 프로토콜을 나열합니다.

유형: 문자열 어레이

배열 구성원: 최소수는 1개입니다. 최대 항목 수 5개.

유효 값: `SFTP | FTPS`

필수 여부: 아니요

SshCiphers

서버 또는 커넥터에 연결된 보안 정책에서 활성화된 SSH (Secure Shell) 암호 암호화 알고리즘을 나열합니다. 이 매개 변수는 서버 및 커넥터 보안 정책 모두에 적용됩니다.

타입: 문자열 배열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 50입니다.

필수 여부: 아니요

SshHostKeyAlgorithms

보안 정책의 호스트 키 알고리즘을 나열합니다.

> **Note**
> 이 매개 변수는 커넥터의 보안 정책에만 적용됩니다.

타입: 문자열 배열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 50입니다.

필수 여부: 아니요

SshKexs

서버 또는 커넥터에 연결된 보안 정책에서 활성화된 SSH 키 교환 (KEX) 암호화 알고리즘을 나열합니다. 이 매개 변수는 서버 및 커넥터 보안 정책 모두에 적용됩니다.

타입: 문자열 배열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 50입니다.

필수 여부: 아니요

SshMacs

서버 또는 커넥터에 연결된 보안 정책에서 활성화된 SSH 메시지 인증 코드 (MAC) 암호화 알고리즘을 나열합니다. 이 매개 변수는 서버 및 커넥터 보안 정책 모두에 적용됩니다.

타입: 문자열 배열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 50입니다.

필수 여부: 아니요

TlsCiphers

서버에 연결된 보안 정책에서 사용하도록 설정된 전송 계층 보안 (TLS) 암호 암호화 알고리즘을 나열합니다.

> **Note**
>
> 이 매개 변수는 서버의 보안 정책에만 적용됩니다.

타입: 문자열 배열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 50입니다.

필수 여부: 아니요

Type

보안 정책이 적용되는 리소스 유형 (서버 또는 커넥터)

타입: 문자열

유효 값: `SERVER | CONNECTOR`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

DescribedServer

지정된 File Transfer 프로토콜 지원 서버의 속성을 설명합니다.

내용

Arn

서버의 고유한 Amazon 리소스 이름(ARN)을 지정합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 20입니다. 최대 길이는 1600입니다.

패턴: `arn:\S+`

필수 사항 여부: Yes

As2ServiceManagedEgressIpAddresses

이 서버의 송신 IP 주소 목록 이러한 IP 주소는 AS2 프로토콜을 사용하는 서버에만 해당됩니다. 비동기 mDN을 전송하는 데 사용됩니다.

이러한 IP 주소는 AS2 서버를 생성할 때 자동으로 할당됩니다. 또한 기존 서버를 업데이트하고 AS2 프로토콜을 추가하면 고정 IP 주소도 할당됩니다.

유형: 문자열 어레이

패턴: `\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}`

Required: No

Certificate

AWS인증서 관리자 (ACM) 인증서의 ARN을 지정합니다. `Protocols`이(가) FTPS(으)로 설정된 경우에 필요합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 1600입니다.

필수 여부: 아니요

Domain

File Transfer에 사용되는 스토리지 시스템의 도메인을 지정합니다. Amazon Simple Storage Service (Amazon S3)와 Amazon Elastic File System (Amazon EFS)의 두 가지 도메인을 사용할 수 있습니다. 기본값은 S3입니다.

타입: 문자열

유효 값: `S3 | EFS`

필수 여부: 아니요

EndpointDetails

서버에 대해 구성된 Virtual Private Cloud(VPC) 엔드포인트 설정입니다. VPC 내에서 엔드포인트를 호스팅할 때 VPC 내의 리소스에만 액세스할 수 있도록 하거나 탄력적 IP 주소를 연결하여 인터넷을 통해 클라이언트에 액세스하도록 할 수 있습니다. VPC의 기본 보안 그룹은 엔드포인트에 자동으로 할당됩니다.

타입: EndpointDetails 객체

필수 항목 여부: 아니요

EndpointType

서버가 연결된 엔드포인트의 타입을 정의합니다. 서버를 VPC 엔드포인트에 연결하면 퍼블릭 인터넷을 통해 서버에 액세스할 수 없습니다.

타입: 문자열

유효 값: `PUBLIC | VPC | VPC_ENDPOINT`

필수 여부: 아니요

HostKeyFingerprint

서버 호스트 키의 Base64로 인코딩된 SHA256 지문을 지정합니다. 이 값은 `ssh-keygen -l -f my-new-server-key` 명령의 출력과 동등합니다.

타입: 문자열

필수사항: 아니요

IdentityProviderDetails

고객 제공 인증 API를 직접 호출하기 위한 정보를 지정합니다. 서버의 `IdentityProviderType`이 `AWS_DIRECTORY_SERVICE`이거나 또는 `SERVICE_MANAGED`인 경우, 이 필드는 채워지지 않습니다.

타입: IdentityProviderDetails 객체

필수 여부: 아니요

IdentityProviderType

서버 인증 모드. 기본값은 이며`SERVICE_MANAGED`, 이를 통해 서비스 내에서 사용자 자격 증명을 저장하고 액세스할 수 있습니다. AWS Transfer Family

온-프레미스 환경에서 AWS Directory Service for Microsoft Active Directory 또는 AD Connector를 AWS 사용하여 Microsoft Active Directory 내의 Active Directory 그룹에 대한 액세스를 제공하는 데 사용합니다`AWS_DIRECTORY_SERVICE`. 또한 이 옵션을 사용하려면 `IdentityProviderDetails` 파라미터를 사용하여 디렉터리 ID를 제공해야 합니다.

`API_GATEWAY` 값을 사용하여 선택하는 자격 증명 제공자와 통합합니다. `API_GATEWAY`를 설정하려면 `IdentityProviderDetails` 파라미터를 사용하여 인증을 요구하도록 Amazon API Gateway 엔드포인트 URL을 제공해야 합니다.

`AWS_LAMBDA`값을 사용하여 AWS Lambda 함수를 ID 공급자로 직접 사용할 수 있습니다. 이 값을 선택하는 경우 `IdentityProviderDetails` 데이터 타입에 대한 `Function` 파라미터에서 lLambda 함수에 대한 ARN을 지정해야 합니다.

타입: 문자열

유효 값: `SERVICE_MANAGED | API_GATEWAY | AWS_DIRECTORY_SERVICE | AWS_LAMBDA`

필수 여부: 아니요

LoggingRole

서버가 Amazon S3 또는 Amazon EFSevents에 대한 아마존 CloudWatch 로깅을 활성화할 수 있도록 허용하는 AWS Identity and Access Management (IAM) 역할의 Amazon 리소스 이름 (ARN). 설정하면 로그에서 사용자 활동을 볼 수 있습니다. CloudWatch

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 2,048.

패턴: `(|arn:.*role/\S+)`

필수 여부: 아니요

PostAuthenticationLoginBanner

사용자가 서버에 연결할 때 표시할 문자열을 지정합니다. 이 문자열은 사용자가 인증한 후에 표시됩니다.

> **Note**
>
> SFTP 프로토콜은 인증 후 디스플레이 배너를 지원하지 않습니다.

타입: 문자열

길이 제약: 최소 길이는 0입니다. 최대 길이는 4096자입니다.

패턴: `[\x09-\x0D\x20-\x7E]*`

필수 여부: 아니요

PreAuthenticationLoginBanner

사용자가 서버에 연결할 때 표시할 문자열을 지정합니다. 이 문자열은 사용자가 인증하기 전에 표시됩니다. 예를 들어 다음 배너는 시스템 사용에 대한 세부 정보를 표시합니다.

```
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.
```

타입: 문자열

길이 제약: 최소 길이는 0입니다. 최대 길이는 4096자입니다.

패턴: `[\x09-\x0D\x20-\x7E]*`

필수 여부: 아니요

ProtocolDetails

서버에 대해 구성된 프로토콜 설정입니다.

- 수동 모드(FTP 및 FTPS 프로토콜의 경우)를 표시하려면 `PassiveIp` 파라미터를 사용합니다. 방화벽, 라우터 또는 로드 밸런서의 외부 IP 주소와 같은 점으로 분리된 단일 쿼드 IPv4 주소를 입력합니다.
- Amazon S3 버킷에 업로드하는 파일에 대해 클라이언트가 `SETSTAT` 명령을 사용하려 할 때 생성되는 오류를 무시하려면 `SetStatOption` 파라미터를 사용합니다. AWS Transfer Family 서버에서 `SETSTAT` 명령을 무시하고 SFTP 클라이언트를 변경할 필요 없이 파일을 업로드하도록 하려면 값을 로 설정합니다. `ENABLE_NO_OP` `SetStatOption`파라미터를 로 `ENABLE_NO_OP` 설정하면 Transfer Family가 Amazon Logs에 CloudWatch 로그 항목을 생성하여 고객이 `SETSTAT` 전화를 거는 시기를 확인할 수 있습니다.
- AWS Transfer Family 서버가 고유한 세션 ID를 통해 최근 협상된 세션을 재개할지 여부를 확인하려면 파라미터를 사용하십시오. `TlsSessionResumptionMode`
- `As2Transports`는 AS2 메시지의 전송 방법을 나타냅니다. 현재는 HTTP만 지원됩니다.

타입: ProtocolDetails 객체

필수 여부: 아니요

Protocols

파일 전송 프로토콜 클라이언트가 서버의 엔드포인트에 연결할 수 있는 파일 전송 프로토콜을 지정합니다. 사용 가능한 프로토콜은 다음과 같습니다:

- SFTP (Secure Shell(SSH) File Transfer Protocol):: SSH를 통한 파일 전송
- FTPS (File Transfer Protocol Secure): TLS 암호화를 사용한 파일 전송
- FTP (File Transfer Protocol): 암호화되지 않은 파일 전송
- AS2(적용 설명 2): 구조화된 데이터를 전송하는 데 사용됩니다. business-to-business

> **Note**
>
> - 선택하는 경우 클라이언트가 FTPS FTPS를 통해 서버에 연결할 때 서버를 식별하는 데 사용되는 ACM AWS Certificate Manager (저장된 인증서) 을 선택해야 합니다.
> - `Protocol`에 FTP 또는 `FTPS`이(가) 포함된 경우 `EndpointType`은(는) `VPC`이고 `IdentityProviderType`은(는) `AWS_DIRECTORY_SERVICE`, `AWS_LAMBDA` 또는 `API_GATEWAY`여야 합니다.
> - `Protocol`에 `FTP`이(가) 포함된 경우 `AddressAllocationIds`를 연결할 수 없습니다.
> - `Protocol`이 `SFTP`로만 설정된 경우 `EndpointType`을 `PUBLIC`으로 설정하고 `IdentityProviderType`을 지원되는 ID 타입(`SERVICE_MANAGED`,

AWS_DIRECTORY_SERVICE, AWS_LAMBDA, API_GATEWAY) 중 하나로 설정할 수 있습니다.

- Protocol에 AS2이(가) 포함된 경우 EndpointType은(는) VPC여야 하고, 도메인은 Amazon S3여야 합니다.

타입: 문자열 배열

배열 멤버: 최소수는 1개입니다. 최대 항목 수는 4개입니다.

유효 값: SFTP | FTP | FTPS | AS2

필수 여부: 아니요

S3StorageOptions

Amazon S3 디렉터리의 성능이 최적화되었는지 여부를 지정합니다. 이 옵션은 기본적으로 비활성화되어 있습니다.

기본적으로 홈 디렉터리 매핑에는 a가 있습니다. TYPE DIRECTORY 이 옵션을 활성화한 경우 매핑에 파일 대상이 포함되도록 하려면 HomeDirectoryMapEntry Type 를 FILE 명시적으로 설정해야 합니다.

유형: S3StorageOptions객체

필수 항목 여부: 아니요

SecurityPolicyName

서버의 보안 정책 이름을 지정합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 100입니다.

패턴: Transfer[A-Za-z0-9]*SecurityPolicy-[A-Za-z0-9-]+

Required: No

ServerId

인스턴스화하는 서버의 고유한 시스템 할당 식별자를 지정합니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

Required: No

State

설명된 서버의 상태입니다. 값이 `ONLINE` 이면 서버가 작업을 수락하고 파일을 전송할 수 있음을 나타냅니다. `State`값이 `OFFLINE`(이)면 서버에서 File Transfer 작업을 수행할 수 없습니다.

`STARTING` 및 `STOPPING` 상태는 서버가 완전히 응답할 수 없거나 완전히 오프라인 상태가 아닌 중간 상태임을 나타냅니다. `START_FAILED` 또는 `STOP_FAILED`의 값은 오류 상태를 나타낼 수 있습니다.

타입: 문자열

유효 값: `OFFLINE | ONLINE | STARTING | STOPPING | START_FAILED | STOP_FAILED`

필수 여부: 아니요

StructuredLogDestinations

사용자의 서버 로그가 전송될 로그 그룹을 지정합니다.

로그 그룹을 지정하려면 기존 로그 그룹의 ARN을 제공해야 합니다. 이 경우, 로그 그룹의 형식은 다음과 같습니다:

`arn:aws:logs:region-name:amazon-account-id:log-group:log-group-name:*`

예제: `arn:aws:logs:us-east-1:111122223333:log-group:mytestgroup:*`

이전에 서버의 로그 그룹을 지정한 경우, `update-server` 호출 시 이 파라미터에 빈 값을 제공하여 로그 그룹을 지우고 사실상 구조화된 로깅을 끌 수 있습니다. 예:

```
update-server --server-id s-1234567890abcdef0 --structured-log-destinations
```

타입: 문자열 배열

배열 멤버: 최소 항목 수는 0개입니다. 최대 항목 수는 1개입니다.

길이 제약 조건: 최소 길이는 20입니다. 최대 길이는 1600입니다.

패턴: `arn:\S+`

Required: No

Tags

설명된 서버에 할당된 서버를 검색하고 그룹화하는 데 사용할 수 있는 키-값 쌍을 지정합니다.

타입: Tag 객체 배열

배열 멤버: 최소 항목 수는 1개입니다. 최대 항목 수는 50개입니다.

필수 여부: 아니요

UserCount

`ServerId`를 사용하여 지정한 서버에 할당되는 사용자 수를 지정합니다.

타입: 정수

필수 항목 여부: 아니요

WorkflowDetails

할당할 워크플로의 워크플로 ID와 워크플로 실행에 사용되는 실행 역할을 지정합니다.

파일이 완전히 업로드될 때 실행되는 워크플로 외에도 `WorkflowDetails`에는 부분 업로드 시 실행할 워크플로의 워크플로 ID와 실행 역할이 포함될 수 있습니다. 파일이 업로드되는 동안 서버 세션 연결이 끊기면 부분 업로드가 수행됩니다.

타입: WorkflowDetails 객체

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

DescribedUser

지정된 사용자의 속성을 설명합니다.

내용

Arn

설명이 요청된 사용자의 고유한 Amazon 리소스 이름(ARN)을 지정합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 20입니다. 최대 길이는 1600입니다.

패턴: `arn:\S+`

필수 여부: 예

HomeDirectory

사용자가 클라이언트를 사용하여 서버에 로그인하는 경우 사용자를 위한 랜딩 디렉터리(폴더).

`HomeDirectory`의 예: `/bucket_name/home/mydirectory`

> **Note**
>
> `HomeDirectory` 파라미터는 `HomeDirectoryType`이(가) `PATH`(으)로 설정된 경우에만 사용합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 1024입니다.

패턴: `(|/.*)`

필수 여부: 아니요

HomeDirectoryMappings

사용자에게 표시할 Amazon S3 또는 Amazon EFS 경로 및 키와 이러한 경로 및 키를 표시할 방법을 지정하는 논리적 디렉터리 매핑입니다. `Entry` 및 `Target` 쌍을 지정해야 합니다. 여기서 `Entry`는 경로가 표시되는 방식을 보여주고 `Target`는 실제 Amazon S3 경로입니다. 대상만 지정하는 경우, 그대로 표시됩니다. 또한 AWS Identity and Access Management (IAM) 역할이 경

로에 대한 액세스를 제공하는지 확인해야 합니다. `Target` 이 값은 LOGICAL로 설정된 경우에만 `HomeDirectoryType`를 설정할 수 있습니다.

대부분의 경우, 세션 정책 대신 이 값을 사용하여 사용자를 지정된 홈 디렉터리(“chroot“)로 제한할 수 있습니다. 이렇게 `Entry` 하려면 '/'로 설정하고 `Target` HomeDirectory 파라미터 값으로 설정하면 됩니다.

유형: HomeDirectoryMapEntry객체 어레이

어레이 멤버: 최소 항목 수 1개. 최대 항목 수는 50000개입니다.

필수 여부: 아니요

HomeDirectoryType

사용자가 서버에 로그인하는 경우 홈 디렉터리가 될 랜딩 디렉터리(폴더) 타입입니다. `PATH`로 설정하면, 사용자는 파일 전송 프로토콜 클라이언트에서와 같이 절대 Amazon S3 버킷 또는 EFS 경로를 볼 수 있습니다. `LOGICAL`로 설정하면, `HomeDirectoryMappings`에서 Amazon S3 또는 Amazon EFS 경로를 사용자에게 표시할 방법에 대한 매핑을 제공해야 합니다.

> **Note**
>
> `HomeDirectoryType`이 `LOGICAL`인 경우, `HomeDirectoryMappings` 파라미터를 사용하여 매핑을 제공해야 합니다. 반면에 `HomeDirectoryType`이 `PATH`인 경우, `HomeDirectory` 파라미터를 사용하여 절대 경로를 제공하세요. 템플릿에 `HomeDirectory` 및 `HomeDirectoryMappings`를 모두 포함할 수는 없습니다.

타입: 문자열

유효 값: `PATH | LOGICAL`

필수 여부: 아니요

Policy

여러 사용자가 동일한 AWS Identity and Access Management (IAM) 역할을 사용할 수 있도록 하기 위한 사용자 세션 정책. 이 정책은 Amazon S3 버킷의 부분에 대한 사용자의 액세스 범위를 축소합니다. 이 정책 내에서 사용할 수 있는 변수는 `${Transfer:UserName}`, `${Transfer:HomeDirectory}` 및 `${Transfer:HomeBucket}`입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 2048입니다.

필수 여부: 아니요

PosixProfile

Amazon Elastic File System(Amazon EFS) 파일 시스템에 대한 사용자의 액세스를 통제하는 사용자 ID(`Uid`), 그룹 ID(`Gid`) 및 보조 그룹 ID(`SecondaryGids`)를 포함한 전체 POSIX 자격 증명을 지정합니다. 파일 시스템의 파일 및 디렉터리에 설정된 POSIX 권한에 따라 Amazon EFS 파일 시스템에서 파일을 송수신할 때 사용자에게 제공되는 액세스 수준이 결정됩니다.

타입: PosixProfile 객체

필수 항목 여부: 아니요

Role

Amazon S3 버킷 또는 Amazon EFS 파일 시스템에 대한 사용자의 액세스를 제어하는 AWS Identity and Access Management (IAM) 역할의 Amazon 리소스 이름 (ARN). 이 역할에 연결된 정책은 Amazon S3 버킷 또는 Amazon EFS 파일 시스템에 대한 파일 송수신 시 사용자에게 제공할 액세스의 수준을 결정합니다. 또한 IAM 역할에는 사용자의 전송 요청을 처리할 때 서버가 해당 리소스에 액세스할 수 있도록 허용하는 신뢰 관계가 포함되어야 합니다.

타입: 문자열

길이 제약: 최소 길이는 20. 최대 길이는 2,048.

패턴: `arn:.*role/\S+`

필수 여부: 아니요

SshPublicKeys

설명된 사용자를 위해 저장된 SSH(Secure Shell) 키의 퍼블릭 키 부분을 지정합니다.

타입: SshPublicKey 객체 배열

배열 멤버: 최소 항목 수 0개. 최대 항목 수 5개.

필수 여부: 아니요

Tags

요청된 사용자의 키-값 쌍을 지정합니다. 태그를 사용하여 다양한 목적으로 사용자를 검색하고 그룹화할 수 있습니다.

타입: Tag 객체 배열

배열 멤버: 최소 항목 수는 1개입니다. 최대 항목 수는 50개입니다.

필수 여부: 아니요

UserName

설명이 요청된 사용자의 명칭을 지정합니다. 사용자 이름은 인증 목적으로 사용됩니다. 사용자가 서버에 로그인할 때 사용하는 문자열입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 3입니다. 최대 길이는 100입니다.

패턴: `[\w][\w@.-]{2,99}`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

DescribedWorkflow

지정된 워크플로에 저장된 속성을 설명합니다.

내용

Arn

워크플로에 고유한 Amazon 리소스 이름(ARN)를 지정합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 20입니다. 최대 길이는 1600입니다.

패턴: `arn:\S+`

필수 사항 여부: Yes

Description

워크플로에 대한 텍스트 설명을 지정합니다.

타입: 문자열

길이 제약: 최소 길이는 0. 최대 길이는 256입니다.

패턴: `[\w- ]*`

Required: No

OnExceptionSteps

워크플로를 실행하는 동안 오류가 발생할 경우 수행할 단계(작업)를 지정합니다.

타입: WorkflowStep 객체 배열

배열 멤버: 최소 항목 수는 0개입니다. 최대 항목 수는 8개입니다.

필수 여부: 아니요

Steps

지정된 워크플로에 있는 단계에 대한 세부 정보를 지정합니다.

타입: WorkflowStep 객체 배열

배열 멤버: 최소 항목 수는 0개입니다. 최대 항목 수는 8개입니다.

필수 여부: 아니요

Tags

워크플로 그룹화 및 검색에 사용될 수 있는 키-값 쌍입니다. 태그는 어떠한 목적으로 워크플로에 연결되는 메타데이터입니다.

타입: Tag 객체 배열

배열 멤버: 최소 항목 수는 1개입니다. 최대 항목 수는 50개입니다.

필수 여부: 아니요

WorkflowId

워크플로의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `w-([a-z0-9]{17})`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

EfsFileLocation

워크플로에서 사용 중인 파일의 파일 위치에 대한 세부 정보를 지정합니다. Amazon Elastic File System(Amazon EFS)을 스토리지로 사용하는 경우에만 적용됩니다.

내용

FileSystemId

Amazon EFS에서 할당한 파일 시스템의 식별자.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 128입니다.

패턴: `(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})`

Required: No

Path

워크플로에서 사용 중인 폴더의 경로 이름.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 65536입니다.

패턴: `[^\x00]+`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

EndpointDetails

File Transfer 프로토콜 지원 서버에 대해 구성된 가상 Virtual Private Cloud(VPC) 엔드포인트 설정입니다. VPC 엔드포인트를 사용하여 서버 및 VPC 내부 전용 리소스에 대한 액세스를 제한할 수 있습니다. 인입 인터넷 트래픽을 제어하려면 `UpdateServer` API를 간접적으로 호출하고 서버의 엔드포인트에 탄력적 IP 주소를 연결합니다.

> **Note**
>
> 2021년 5월 19일 이후에는 계정에서 2021년 5월 19일 이전에 서버를 생성하지 않은 경우 `EndpointType=VPC_ENDPOINT` AWS계정에서 를 사용하여 서버를 생성할 수 없습니다. 2021년 5월 19일 또는 그 이전에 AWS계정에 서버를 이미 생성한 경우 영향을 받지 않습니다. `EndpointType=VPC_ENDPOINT` 이 날짜 이후에는 `EndpointType=VPC`를 사용하세요. 자세한 내용은 VPC_ENDPOINT 사용 중단를 참조하세요.

목차

AddressAllocationIds

서버의 엔드포인트에 탄력적 IP 주소를 연결하는 데 필요한 주소 할당 ID 목록입니다.

주소 할당 ID는 엘라스틱 IP 주소의 할당 ID에 해당합니다. 이 값은 Amazon EC2 주소 데이터 `allocationId` 유형의 필드에서 검색할 수 있습니다. 이 값을 검색하는 한 가지 방법은 EC2 DescribeAddressesAPI를 호출하는 것입니다.

이 파라미터는 선택 사항입니다. VPC 엔드포인트를 퍼블릭으로 설정하려면 이 파라미터를 설정하세요. 자세한 내용은 서버용 인터넷 연결 엔드포인트 만들기를 참조하십시오.

> **Note**
>
> 이 속성은 다음과 같은 방식으로만 설정할 수 있습니다.
>
> - `EndpointType`로 설정해야 합니다. `VPC`
> - Transfer Family 서버는 오프라인 상태여야 합니다.
> - FTP 프로토콜을 사용하는 Transfer Family 서버에는 이 매개 변수를 설정할 수 없습니다.
> - 서버가 이미 `SubnetIds` 채워져 있어야 `SubnetIds` 하며 동시에 업데이트할 `AddressAllocationIds` 수는 없습니다.

- AddressAllocationIds중복된 항목을 포함할 수 없으며 길이가 다음과 같아야 합니다. SubnetIds 예를 들어 서브넷 ID가 3개인 경우 주소 할당 ID도 3개를 지정해야 합니다.
- UpdateServerAPI를 호출하여 이 파라미터를 설정하거나 변경합니다.

유형: String 배열

필수 여부: 아니요

SecurityGroupIds

서버의 엔드포인트에 연결할 수 있는 보안 그룹 ID 목록입니다.

> **Note**
>
> 이 속성은 EndpointType이(가) VPC(으)로 설정된 경우에만 사용할 수 있습니다. 에서 PUBLIC 또는 VPC_ENDPOINT 로 변경하는 경우에만 UpdateServerEndpointTypeAPI에서 SecurityGroupIds 속성을 편집할 수 VPC 있습니다. 생성 후 서버의 VPC 엔드포인트와 연결된 보안 그룹을 변경하려면 Amazon ModifyVpcEndpointEC2 API를 사용하십시오.

유형: 문자열 어레이

길이 제약 조건: 최소 길이는 11입니다. 최대 길이는 20입니다.

패턴: sg-[0-9a-f]{8,17}

Required: No

SubnetIds

VPC에서 서버 엔드포인트를 호스팅하는 데 필요한 서브넷 ID 목록입니다.

> **Note**
>
> 이 속성은 EndpointType이(가) VPC(으)로 설정된 경우에만 사용할 수 있습니다.

타입: 문자열 배열

필수 여부: 아니요

VpcEndpointId

VPC 엔드포인트의 식별자입니다.

> **Note**
>
> 이 속성은 `EndpointType`이(가) `VPC_ENDPOINT`(으)로 설정된 경우에만 사용할 수 있습니다.
> 자세한 내용은 VPC_ENDPOINT 사용 중단를 참조하세요.

타입: 문자열

길이 제약 조건: 고정 길이는 22입니다.

패턴: `vpce-[0-9a-f]{17}`

Required: No

VpcId

서버의 엔드포인트가 호스팅될 VPC의 VPC 식별자입니다.

> **Note**
>
> 이 속성은 `EndpointType`이(가) `VPC`(으)로 설정된 경우에만 사용할 수 있습니다.

타입: 문자열

필수 항목 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

ExecutionError

워크플로 실행 중에 발생하는 오류에 대한 오류 메시지와 타입을 지정합니다.

내용

Message

`ErrorType`에 해당하는 설명 메시지를 지정합니다.

타입: 문자열

필수 항목 여부: 예

Type

오류 타입을 지정합니다.

- `ALREADY_EXISTS`: 덮어쓰기 옵션을 선택하지 않고 대상 위치에 같은 이름의 파일이 이미 있는 경우 복사 단계에서 발생합니다.
- `BAD_REQUEST`: 일반적으로 잘못된 요청: 예를 들어, S3 파일에만 태그를 지정할 수 있으므로 EFS 파일에 태그를 지정하려고 시도하는 단계에서는 `BAD_REQUEST`를 반환합니다.
- `CUSTOM_STEP_FAILED`: 사용자 지정 단계에서 실패를 나타내는 콜백을 제공한 경우 발생합니다.
- `INTERNAL_SERVER_ERROR`: 다양한 이유로 발생할 수 있는 포괄적인 오류입니다.
- `NOT_FOUND`: 요청된 개체(예: 복사 단계의 소스 파일)가 존재하지 않을 때 발생합니다.
- `PERMISSION_DENIED`: 워크플로에서 하나 이상의 단계를 완료할 수 있는 올바른 권한이 정책에 포함되어 있지 않은 경우 발생합니다.
- `TIMEOUT`: 실행 시간이 초과되면 발생합니다.

  > **Note**
  >
  > 사용자 지정 단계에 대해 `TimeoutSeconds`을 1초에서 1800(30분) 사이로 설정할 수 있습니다.

- `THROTTLED`: 초당 한 워크플로의 새 실행 리필 비율을 초과할 경우 발생합니다.

타입: 문자열

유효 값: `PERMISSION_DENIED | CUSTOM_STEP_FAILED | THROTTLED | ALREADY_EXISTS | NOT_FOUND | BAD_REQUEST | TIMEOUT | INTERNAL_SERVER_ERROR`

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

ExecutionResults

워크플로의 단계와 워크플로 실행 중 오류가 발생할 경우 실행할 단계를 지정합니다.

내용

OnExceptionSteps

워크플로를 실행하는 동안 오류가 발생할 경우 수행할 단계(작업)를 지정합니다.

타입: ExecutionStepResult 객체 배열

배열 멤버: 최소 항목 수는 1개입니다. 최대 항목 수는 50개입니다.

필수 여부: 아니요

Steps

지정된 워크플로에 있는 단계에 대한 세부 정보를 지정합니다.

타입: ExecutionStepResult 객체 배열

배열 멤버: 최소 항목 수는 1개입니다. 최대 항목 수는 50개입니다.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

ExecutionStepResult

단계에 대해 오류(있는 경우), 출력(있는 경우), 단계 타입 등의 세부 정보를 지정합니다.

내용

Error

지정된 워크플로 단계를 실행하는 동안 오류가 발생한 경우 해당 오류의 세부 정보를 지정합니다.

타입: ExecutionError 객체

필수 여부: 아니요

Outputs

키/값 쌍의 값이 파일에 태그로 적용되었습니다. 단계 타입이 TAG인 경우에만 적용됩니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 65536입니다.

필수 여부: 아니요

StepType

사용 가능한 단계 타입 중 하나입니다.

- **COPY** - 다른 위치에 파일을 복사합니다.
- **CUSTOM**- AWS Lambda 함수 타겟을 사용하여 사용자 지정 단계를 수행합니다.
- **DECRYPT** - 업로드되기 전에 암호화된 파일을 복호화합니다.
- **DELETE** - 파일을 삭제합니다.
- **TAG** - 파일에 태그를 추가합니다.

타입: 문자열

유효 값: `COPY | CUSTOM | TAG | DELETE | DECRYPT`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

FileLocation

이 단계에서 사용할 Amazon S3 또는 EFS 파일 세부 정보를 지정합니다.

내용

EfsFileLocation

Amazon EFS 식별자와 사용 중인 파일의 경로를 지정합니다.

타입: EfsFileLocation 객체

필수 여부: 아니요

S3FileLocation

사용 중인 파일의 S3 세부 정보(예: 버킷, ETag 등)를 지정합니다.

타입: S3FileLocation 객체

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

HomeDirectoryMapEntry

HomeDirectoryMappings을 위한 항목과 대상을 포함하는 객체를 나타냅니다.

다음은 chroot에 대한 Entry 및 Target 쌍의 예입니다.

```
[ { "Entry": "/", "Target": "/bucket_name/home/mydirectory" } ]
```

내용

Entry

HomeDirectoryMappings에 대한 항목을 나타냅니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 1024입니다.

패턴: /.*

필수 여부: 예

Target

HomeDirectoryMapEntry에서 사용되는 맵 대상을 나타냅니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 1024입니다.

패턴: /.*

필수 사항 여부: Yes

Type

매핑 유형을 지정합니다. 매핑이 파일을 가리키거나 DIRECTORY 디렉토리가 디렉토리를 가리키도록 하려면 FILE 유형을 로 설정합니다.

> **Note**
>
> 기본적으로 Transfer Family 서버를 DIRECTORY 생성하면 홈 디렉토리 매핑이 해제됩니다. Type 매핑에 파일 대상을 지정하려면 명시적으로 Type 로 FILE 설정해야 합니다.

타입: 문자열

유효 값: `FILE | DIRECTORY`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

IdentityProviderDetails

파일 전송 프로토콜을 사용하는 서버의 사용자에게 사용되는 사용자 인증 타입과 관련된 정보를 반환합니다. 서버는 인증 메서드로 하나만 사용할 수 있습니다.

내용

DirectoryId

ID 제공자로 사용하려는 AWS Directory Service 디렉토리의 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 12입니다.

패턴: `d-[0-9a-f]{10}`

필수 여부: 아니요

Function

ID 제공업체에 사용할 Lambda 함수의 ARN입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 170입니다.

패턴: `arn:[a-z-]+:lambda:.*`

필수 여부: 아니요

InvocationRole

이 파라미터는 `IdentityProviderType`가 `API_GATEWAY`인 경우에만 적용됩니다. 사용자 계정을 인증하는 데 사용되는 `InvocationRole`의 타입을 제공합니다.

타입: 문자열

길이 제약: 최소 길이는 20. 최대 길이는 2,048.

패턴: `arn:.*role/\S+`

필수 여부: 아니요

SftpAuthenticationMethods

SFTP 지원 서버와 맞춤 ID 공급자의 경우에만 암호, SSH 키 쌍 또는 둘 다를 사용하여 인증할지 여부를 지정할 수 있습니다.

- PASSWORD - 연결하려면 사용자가 암호를 입력해야 합니다.
- PUBLIC_KEY - 연결하려면 사용자가 프라이빗 키를 제공해야 합니다.
- PUBLIC_KEY_OR_PASSWORD - 사용자는 암호나 키로 인증할 수 있습니다. 이것이 기본값입니다.
- PUBLIC_KEY_AND_PASSWORD - 연결하려면 사용자가 프라이빗 키와 암호를 모두 제공해야 합니다. 서버가 먼저 키를 확인한 다음 키가 유효하면 암호를 입력하라는 메시지가 표시됩니다. 제공된 프라이빗 키가 저장된 퍼블릭 키와 일치하지 않는 경우 인증이 실패합니다.

타입: 문자열

유효 값: PASSWORD | PUBLIC_KEY | PUBLIC_KEY_OR_PASSWORD | PUBLIC_KEY_AND_PASSWORD

필수 여부: 아니요

Url

사용자를 인증하는 데 사용되는 서비스 엔드포인트의 위치를 제공합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 255입니다.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

InputFileLocation

처리 중인 파일의 위치를 지정합니다.

내용

EfsFileLocation

복호화 중인 Amazon Elastic File System(Amazon EFS) 파일의 세부 정보를 지정합니다.

타입: EfsFileLocation 객체

필수 여부: 아니요

S3FileLocation

복사 또는 복호화 중인 Amazon S3 파일의 세부 정보를 지정합니다.

타입: S3InputFileLocation 객체

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

ListedAccess

하나 이상의 지정된 관련 액세스의 속성을 나열합니다.

내용

ExternalId

디렉터리 내의 특정 그룹을 식별하는 데 필요한 고유 식별자입니다. 연결하는 그룹의 사용자는 사용하는 활성화된 프로토콜을 통해 Amazon S3 또는 Amazon EFS 리소스에 액세스할 수 AWS Transfer Family있습니다. 그룹 이름을 아는 경우 PowerShell Windows를 사용하여 다음 명령을 실행하여 SID 값을 볼 수 있습니다.

```
Get-ADGroup -Filter {samAccountName -like "YourGroupName*"} -Properties
* | Select SamAccountName,ObjectSid
```

이 명령에서 Active Directory 그룹의 YourGroupName이름으로 바꾸십시오.

이 파라미터를 확인하는 데 사용되는 정규 표현식은 공백 없이 대문자 및 소문자 영숫자로 구성된 문자의 문자열입니다. 밑줄이나 또한 =,.@:/- 문자도 포함할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 256입니다.

패턴: `S-1-[\d-]+`

필수 여부: 아니요

HomeDirectory

사용자가 클라이언트를 사용하여 서버에 로그인하는 경우 사용자를 위한 랜딩 디렉터리(폴더).

`HomeDirectory`의 예: `/bucket_name/home/mydirectory`

> **Note**
>
> `HomeDirectory` 파라미터는 `HomeDirectoryType`이(가) `PATH`(으)로 설정된 경우에만 사용합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 1024입니다.

패턴: `(|/.*)`

Required: No

HomeDirectoryType

사용자가 서버에 로그인하는 경우 홈 디렉터리가 될 랜딩 디렉터리(폴더) 타입입니다. `PATH`로 설정하면, 사용자는 파일 전송 프로토콜 클라이언트에서와 같이 절대 Amazon S3 버킷 또는 EFS 경로를 볼 수 있습니다. `LOGICAL`로 설정하면, `HomeDirectoryMappings`에서 Amazon S3 또는 Amazon EFS 경로를 사용자에게 표시할 방법에 대한 매핑을 제공해야 합니다.

> **Note**
>
> `HomeDirectoryType`이 `LOGICAL`인 경우, `HomeDirectoryMappings` 파라미터를 사용하여 매핑을 제공해야 합니다. 반면에 `HomeDirectoryType`이 `PATH`인 경우, `HomeDirectory` 파라미터를 사용하여 절대 경로를 제공하세요. 템플릿에 `HomeDirectory` 및 `HomeDirectoryMappings`를 모두 포함할 수는 없습니다.

타입: 문자열

유효 값: `PATH | LOGICAL`

필수 여부: 아니요

Role

Amazon S3 버킷 또는 Amazon EFS 파일 시스템에 대한 사용자의 액세스를 제어하는 AWS Identity and Access Management (IAM) 역할의 Amazon 리소스 이름 (ARN). 이 역할에 연결된 정책은 Amazon S3 버킷 또는 Amazon EFS 파일 시스템에 대한 파일 송수신 시 사용자에게 제공할 액세스의 수준을 결정합니다. 또한 IAM 역할에는 사용자의 전송 요청을 처리할 때 서버가 해당 리소스에 액세스할 수 있도록 허용하는 신뢰 관계가 포함되어야 합니다.

타입: 문자열

길이 제약: 최소 길이는 20. 최대 길이는 2,048.

패턴: `arn:.*role/\S+`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

ListedAgreement

계약의 속성을 설명합니다.

내용

AgreementId

계약의 고유 식별자입니다. 계약을 생성하면 이 식별자가 반환됩니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `a-([0-9a-f]{17})`

Required: No

Arn

지정된 계약의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 20입니다. 최대 길이는 1600입니다.

패턴: `arn:\S+`

Required: No

Description

계약에 대한 현재 설명입니다. `UpdateAgreement` 작업을 직접적으로 호출하고 새 설명을 제공하여 변경할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 200입니다.

패턴: `[\p{Graph}]+`

필수 여부: 아니요

LocalProfileId

AS2 로컬 프로필의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `p-([0-9a-f]{17})`

필수 여부: 아니요

PartnerProfileId

파트너 프로필의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `p-([0-9a-f]{17})`

Required: No

ServerId

계약의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

Required: No

Status

계약은 `ACTIVE` 또는 `INACTIVE` 둘 중 하나일 수 있습니다.

타입: 문자열

유효 값: `ACTIVE | INACTIVE`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

ListedCertificate

인증서의 속성을 설명합니다.

내용

ActiveDate

인증서가 활성화되는 시기를 지정하는 선택적 날짜입니다.

타입: Timestamp

필수 여부: 아니요

Arn

지정된 인증서의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 20입니다. 최대 길이는 1600입니다.

패턴: `arn:\S+`

필수 여부: 아니요

CertificateId

가져온 인증서의 식별자 배열입니다. 프로필 및 파트너 프로필 작업에 이 식별자를 사용합니다.

타입: 문자열

길이 제약 조건: 고정 길이는 22입니다.

패턴: `cert-([0-9a-f]{17})`

Required: No

Description

인증서를 식별하는 데 사용되는 이름 또는 간단한 설명.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 200입니다.

패턴: [\p{Graph}]+

필수 여부: 아니요

InactiveDate

인증서가 비활성화되는 시기를 지정하는 선택적 날짜.

타입: Timestamp

필수 여부: 아니요

Status

인증서는 ACTIVE, PENDING_ROTATION 또는 INACTIVE일 수 있습니다. PENDING_ROTATION은(는) 이 인증서가 만료되면 현재 인증서를 대체함을 의미합니다.

타입: 문자열

유효 값: ACTIVE | PENDING_ROTATION | INACTIVE

필수 여부: 아니요

Type

인증서의 타입입니다. 인증서에 프라이빗 키가 지정된 경우 해당 타입은 CERTIFICATE_WITH_PRIVATE_KEY입니다. 프라이빗 키가 없는 경우 타입은 CERTIFICATE입니다.

타입: 문자열

유효 값: CERTIFICATE | CERTIFICATE_WITH_PRIVATE_KEY

필수 여부: 아니요

Usage

이 인증서를 사용하는 방법을 지정합니다. 다음과 같은 방법으로 사용할 수 있습니다.

- SIGNING: AS2 메시지 서명용
- ENCRYPTION: AS2 메시지 암호화용
- TLS: HTTPS를 통해 전송되는 AS2 통신을 보호하는 데 사용됩니다.

타입: 문자열

유효 값: SIGNING | ENCRYPTION

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

ListedConnector

지정된 커넥터의 세부 정보를 반환합니다.

내용

Arn

지정된 커넥터의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 20입니다. 최대 길이는 1600입니다.

패턴: `arn:\S+`

Required: No

ConnectorId

커넥터의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `c-([0-9a-f]{17})`

Required: No

Url

파트너의 AS2 또는 SFTP 엔드포인트의 URL입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 255입니다.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

ListedExecution

지정된 실행의 속성을 반환합니다.

내용

ExecutionId

워크플로 실행의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 36입니다.

패턴: `[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}`

Required: No

InitialFileLocation

Amazon S3 또는 EFS 파일 위치를 설명하는 구조입니다. 실행이 시작될 때의 파일 위치입니다. 파일이 복사되는 경우 이 위치는 대상이 아닌 초기 파일 위치입니다.

타입: FileLocation 객체

필수 항목 여부: 아니요

ServiceMetadata

워크플로와 관련된 세션 세부 정보를 위한 컨테이너 객체입니다.

타입: ServiceMetadata 객체

필수 항목 여부: 아니요

Status

상태는 실행 중 하나입니다. 진행 중이거나, 완료되었거나, 예외가 발생했거나, 예외를 처리 중일 수 있습니다.

타입: 문자열

유효 값: `IN_PROGRESS | COMPLETED | EXCEPTION | HANDLING_EXCEPTION`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

ListedHostKey

지정된 호스트 키의 속성을 반환합니다.

내용

Arn

호스트 키의 고유한 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 20입니다. 최대 길이는 1600입니다.

패턴: `arn:\S+`

필수 사항 여부: Yes

DateImported

호스트 키가 서버에 추가된 날짜입니다.

타입: Timestamp

필수 여부: 아니요

Description

호스트 키에 대한 현재 설명. `UpdateHostKey` 작업을 직접적으로 호출하고 새 설명을 제공하여 변경할 수 있습니다.

타입: 문자열

길이 제한: 최소 길이는 0. 최대 길이는 200입니다.

패턴: `[\p{Print}]*`

Required: No

Fingerprint

퍼블릭 키 핑거프린트는 더 긴 퍼블릭 키를 식별하는 데 사용되는 짧은 시퀀스의 바이트입니다.

타입: 문자열

필수사항: 아니요

HostKeyId

호스트 키의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 25입니다.

패턴: `hostkey-[0-9a-f]{17}`

Required: No

Type

호스트 키에 사용되는 암호화 알고리즘입니다. `Type` 파라미터는 다음 값 중 하나를 사용하여 지정합니다.

- `ssh-rsa`
- `ssh-ed25519`
- `ecdsa-sha2-nistp256`
- `ecdsa-sha2-nistp384`
- `ecdsa-sha2-nistp521`

타입: 문자열

필수 항목 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

ListedProfile

지정된 프로필의 속성을 반환합니다.

내용

Arn

지정된 프로필의 Amazon 리소스 이름(ARN)입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 20입니다. 최대 길이는 1600입니다.

패턴: `arn:\S+`

Required: No

As2Id

`As2Id`은(는) RFC 4130에 정의된 AS2-name입니다. 인바운드 전송의 경우 파트너로부터 전송된 AS2 메시지의 `AS2-From` 헤더입니다. 아웃바운드 커넥터의 경우 `StartFileTransfer` API 작업을 사용하여 파트너에게 전송된 AS2 메시지의 `AS2-To` 헤더입니다. 이 ID는 공백을 포함할 수 없습니다.

타입: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: `[\p{Print}\s]*`

Required: No

ProfileId

로컬 또는 파트너 AS2 프로필의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `p-([0-9a-f]{17})`

Required: No

ProfileType

LOCAL 타입 프로필만 나열할지 아니면 PARTNER 타입 프로필만 나열할지를 나타냅니다. 요청에 입력되지 않은 경우 명령은 모든 타입의 프로필을 나열합니다.

타입: 문자열

유효 값: LOCAL | PARTNER

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

ListedServer

지정된 File Transfer 프로토콜 지원 서버의 속성을 반환합니다.

내용

Arn

나열할 서버의 고유한 Amazon 리소스 이름(ARN)을 지정합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 20입니다. 최대 길이는 1600입니다.

패턴: `arn:\S+`

필수 사항 여부: Yes

Domain

File Transfer에 사용되는 스토리지 시스템의 도메인을 지정합니다. Amazon Simple Storage Service (Amazon S3)와 Amazon Elastic File System (Amazon EFS)의 두 가지 도메인을 사용할 수 있습니다. 기본값은 S3입니다.

타입: 문자열

유효 값: `S3 | EFS`

필수 여부: 아니요

EndpointType

서버가 연결된 VPC 엔드포인트의 타입을 지정합니다. 서버를 VPC 엔드포인트에 연결하면 퍼블릭 인터넷을 통해 서버에 액세스할 수 없습니다.

타입: 문자열

유효 값: `PUBLIC | VPC | VPC_ENDPOINT`

필수 여부: 아니요

IdentityProviderType

서버 인증 모드. 기본값은 이며`SERVICE_MANAGED`, 이를 통해 AWS Transfer Family 서비스 내에서 사용자 자격 증명을 저장하고 액세스할 수 있습니다.

온-프레미스 환경에서 AWS Directory Service for Microsoft Active Directory 또는 AD Connector를 AWS 사용하여 Microsoft Active Directory 내의 Active Directory 그룹에 대한 액세스를 제공하는 데 사용합니다AWS_DIRECTORY_SERVICE. 또한 이 옵션을 사용하려면 `IdentityProviderDetails` 파라미터를 사용하여 디렉터리 ID를 제공해야 합니다.

`API_GATEWAY` 값을 사용하여 선택하는 자격 증명 제공자와 통합합니다. `API_GATEWAY`를 설정하려면 `IdentityProviderDetails` 파라미터를 사용하여 인증을 요구하도록 Amazon API Gateway 엔드포인트 URL을 제공해야 합니다.

`AWS_LAMBDA`값을 사용하여 AWS Lambda 함수를 ID 공급자로 직접 사용할 수 있습니다. 이 값을 선택하는 경우 `IdentityProviderDetails` 데이터 타입에 대한 `Function` 파라미터에서 lLambda 함수에 대한 ARN을 지정해야 합니다.

타입: 문자열

유효 값: `SERVICE_MANAGED | API_GATEWAY | AWS_DIRECTORY_SERVICE | AWS_LAMBDA`

필수 여부: 아니요

LoggingRole

서버가 Amazon S3 또는 Amazon EFSevents에 대한 아마존 CloudWatch 로깅을 활성화할 수 있도록 허용하는 AWS Identity and Access Management (IAM) 역할의 Amazon 리소스 이름 (ARN). 설정하면 로그에서 사용자 활동을 볼 수 있습니다. CloudWatch

타입: 문자열

길이 제약: 최소 길이는 20. 최대 길이는 2,048.

패턴: `arn:.*role/\S+`

Required: No

ServerId

나열된 서버의 고유한 시스템 할당 식별자를 지정합니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

Required: No

State

설명된 서버의 상태입니다. 값이 `ONLINE` 이면 서버가 작업을 수락하고 파일을 전송할 수 있음을 나타냅니다. `State`값이 `OFFLINE`(이)면 서버에서 File Transfer 작업을 수행할 수 없습니다.

`STARTING` 및 `STOPPING` 상태는 서버가 완전히 응답할 수 없거나 완전히 오프라인 상태가 아닌 중간 상태임을 나타냅니다. `START_FAILED` 또는 `STOP_FAILED`의 값은 오류 상태를 나타낼 수 있습니다.

타입: 문자열

유효 값: `OFFLINE | ONLINE | STARTING | STOPPING | START_FAILED | STOP_FAILED`

필수 여부: 아니요

UserCount

`ServerId`를 사용하여 지정한 서버에 할당되는 사용자 수를 지정합니다.

타입: 정수

필수 항목 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

ListedUser

지정한 사용자의 속성을 반환합니다.

내용

Arn

알아보려고 하는 사용자 고유의 Amazon 리소스 이름(ARN)을 제공합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 20입니다. 최대 길이는 1600입니다.

패턴: `arn:\S+`

필수 여부: 예

HomeDirectory

사용자가 클라이언트를 사용하여 서버에 로그인하는 경우 사용자를 위한 랜딩 디렉터리(폴더).

`HomeDirectory`의 예: `/bucket_name/home/mydirectory`

> **Note**
>
> `HomeDirectory` 파라미터는 `HomeDirectoryType`이(가) `PATH`(으)로 설정된 경우에만 사용합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 1024입니다.

패턴: `(|/.*)`

Required: No

HomeDirectoryType

사용자가 서버에 로그인하는 경우 홈 디렉터리가 될 랜딩 디렉터리(폴더) 타입입니다. `PATH`로 설정하면, 사용자는 파일 전송 프로토콜 클라이언트에서와 같이 절대 Amazon S3 버킷 또는 EFS 경

로를 볼 수 있습니다. LOGICAL로 설정하면, HomeDirectoryMappings에서 Amazon S3 또는 Amazon EFS 경로를 사용자에게 표시할 방법에 대한 매핑을 제공해야 합니다.

> **Note**
>
> HomeDirectoryType이 LOGICAL인 경우, HomeDirectoryMappings 파라미터를 사용하여 매핑을 제공해야 합니다. 반면에 HomeDirectoryType이 PATH인 경우, HomeDirectory 파라미터를 사용하여 절대 경로를 제공하세요. 템플릿에 HomeDirectory 및 HomeDirectoryMappings를 모두 포함할 수는 없습니다.

타입: 문자열

유효 값: `PATH | LOGICAL`

필수 여부: 아니요

Role

Amazon S3 버킷 또는 Amazon EFS 파일 시스템에 대한 사용자의 액세스를 제어하는 AWS Identity and Access Management (IAM) 역할의 Amazon 리소스 이름 (ARN). 이 역할에 연결된 정책은 Amazon S3 버킷 또는 Amazon EFS 파일 시스템에 대한 파일 송수신 시 사용자에게 제공할 액세스의 수준을 결정합니다. 또한 IAM 역할에는 사용자의 전송 요청을 처리할 때 서버가 해당 리소스에 액세스할 수 있도록 허용하는 신뢰 관계가 포함되어야 합니다.

> **Note**
>
> IAM 역할은 사용자가 Domain=S3를 가진 서버에 대한 Amazon S3 버킷 또는 Domain=EFS를 가진 서버에 대한 EFS 파일 시스템에 대한 액세스하는 것을 제어하는 입니다.
> 이 역할에 연결된 정책은 S3 버킷 또는 EFS 파일 시스템에 대한 파일 송수신 시 사용자에게 제공할 액세스의 수준을 결정합니다.

타입: 문자열

길이 제약: 최소 길이는 20. 최대 길이는 2,048.

패턴: `arn:.*role/\S+`

Required: No

SshPublicKeyCount

지정한 사용자에 대해 저장된 SSH 퍼블릭 키 수를 지정합니다.

타입: 정수

필수 항목 여부: 아니요

UserName

ARN이 지정된 사용자의 이름을 지정합니다. 사용자 이름은 인증 목적으로 사용됩니다.

타입: 문자열

길이 제약 조건: 최소 길이는 3입니다. 최대 길이는 100입니다.

패턴: `[\w][\w@.-]{2,99}`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

ListedWorkflow

워크플로의 식별자, 텍스트 설명 및 Amazon 리소스 이름(ARN)을 포함합니다.

내용

Arn

워크플로에 고유한 Amazon 리소스 이름(ARN)를 지정합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 20입니다. 최대 길이는 1600입니다.

패턴: `arn:\S+`

Required: No

Description

워크플로에 대한 텍스트 설명을 지정합니다.

타입: 문자열

길이 제약: 최소 길이는 0. 최대 길이는 256입니다.

패턴: `[\w- ]*`

Required: No

WorkflowId

워크플로의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `w-([a-z0-9]{17})`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

LoggingConfiguration

로깅 역할과 로그 그룹 명칭으로 구성됩니다.

내용

LoggingRole

서버가 Amazon S3 또는 Amazon EFSevents에 대한 아마존 CloudWatch 로깅을 활성화할 수 있도록 허용하는 AWS Identity and Access Management (IAM) 역할의 Amazon 리소스 이름 (ARN). 설정하면 로그에서 사용자 활동을 볼 수 있습니다. CloudWatch

타입: 문자열

길이 제약: 최소 길이는 20. 최대 길이는 2,048.

패턴: `arn:.*role/\S+`

Required: No

LogGroupName

이 워크플로가 속한 AWS Transfer Family 서버의 CloudWatch 로깅 그룹 이름.

유형: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 512입니다.

패턴: `[\.\-_/#A-Za-z0-9]*`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

PosixProfile

Amazon EFS 파일 시스템에 대한 사용자의 액세스를 통제하는 사용자 ID(`Uid`), 그룹 ID(`Gid`) 및 보조 그룹 ID(`SecondaryGids`)를 포함한 전체 POSIX 자격 증명입니다. 파일 시스템의 파일 및 디렉터리에 설정된 POSIX 권한에 따라 Amazon EFS 파일 시스템에서 파일을 송수신할 때 사용자에게 제공되는 액세스 수준이 결정됩니다.

내용

Gid

이 사용자의 모든 EFS 작업에 사용되는 POSIX 그룹 ID입니다.

타입: Long

유효한 범위: 최소값 0. 최댓값은 4294967295입니다.

필수 여부: 예

Uid

이 사용자의 모든 EFS 작업에 사용하는 POSIX 사용자 ID입니다.

타입: Long

유효한 범위: 최소값 0. 최댓값은 4294967295입니다.

필수 여부: 예

SecondaryGids

이 사용자의 모든 EFS 작업에 사용되는 보조 POSIX 그룹 ID입니다.

타입: 로그 배열

어레이 멤버: 최소 항목 수 0개. 최대 항목 수는 16개입니다.

유효한 범위: 최소값 0. 최댓값은 4294967295입니다.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

ProtocolDetails

서버에 대해 구성된 프로토콜 설정입니다.

내용

As2Transports

AS2 메시지의 전송 방법을 나타냅니다. 현재는 HTTP만 지원됩니다.

타입: 문자열 배열

배열 멤버: 고정된 항목 수는 1개입니다.

유효 값: HTTP

필수 여부: 아니요

PassiveIp

FTP 및 FTPS 프로토콜에 대한 수동 모드를 나타냅니다. 방화벽, 라우터 또는 로드 밸런서의 퍼블릭 IP 주소와 같은 단일 IPv4 주소를 입력합니다. 예:

```
aws transfer update-server --protocol-details PassiveIp=0.0.0.0
```

위의 예에서 `0.0.0.0`을 사용하려는 실제 IP 주소로 바꿉니다.

> **Note**
>
> `PassiveIp` 값을 변경하는 경우 변경 사항을 적용하려면 Transfer Family 서버를 중지했다가 다시 시작해야 합니다. NAT 환경에서 패시브 모드 (PASV) 를 사용하는 방법에 대한 자세한 내용은 방화벽 또는 NAT를 사용하여 FTPS 서버 구성을 참조하십시오. AWS Transfer Family

특수 값

`AUTO` 및 `0.0.0.0`은 `PassiveIp` 파라미터에 대한 특수 값입니다. 값 `PassiveIp=AUTO`는 기본적으로 FTP 및 FTPS 타입 서버에 할당됩니다. 이 경우 서버는 PASV 응답 내의 엔드포인트 IP 중 하나를 사용하여 자동으로 응답합니다. `PassiveIp=0.0.0.0`에는 더욱 고유한 사용 방법이 있습니다. 예를 들어 서브넷이 3개 있는 고가용성(HA) Network Load Balancer(NLB) 환경이 있

는 경우 PassiveIp 파라미터를 사용하여 단일 IP 주소만 지정할 수 있습니다. 이렇게 하면 고가용성의 효과가 감소합니다. 이 경우 PassiveIp=0.0.0.0을 지정할 수 있습니다. 이렇게 하면 클라이언트에 Control 연결과 동일한 IP 주소를 사용하고 연결에 모든 AZ를 사용하도록 지시합니다. 하지만 모든 FTP 클라이언트가 응답을 지원하는 것은 아닙니다. PassiveIp=0.0.0.0 FileZilla 그리고 WinSCP는 그것을 지원합니다. 다른 클라이언트를 사용하는 경우 해당 클라이언트가 PassiveIp=0.0.0.0 응답을 지원하는지 확인하세요.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 15입니다.

필수 여부: 아니요

SetStatOption

SetStatOption을 사용하여 S3 버킷에 업로드하는 파일에 대해 클라이언트가 SETSTAT를 사용하려고 할 때 생성되는 오류를 무시합니다.

일부 SFTP 파일 전송 클라이언트는 파일을 업로드할 때 SETSTAT와 같은 명령을 사용하여 타임스탬프 및 권한을 포함한 원격 파일의 속성을 변경하려고 시도할 수 있습니다. 그러나 이러한 명령은 Amazon S3와 같은 객체 스토리지 시스템과 호환되지 않습니다. 이러한 비호환성으로 인해 이러한 클라이언트에서 파일을 업로드하면 파일이 성공적으로 업로드된 경우에도 오류가 발생할 수 있습니다.

Transfer Family 서버가 SETSTAT 명령을 무시하고 SFTP 클라이언트를 변경할 필요 없이 파일을 업로드하려면 값을 ENABLE_NO_OP로 설정합니다. SetStatOptionENABLE_NO_OP설정은 오류를 무시하지만 Amazon CloudWatch Logs에 로그 항목을 생성하므로 클라이언트가 언제 SETSTAT 전화를 걸는지 확인할 수 있습니다.

> **Note**
>
> 파일의 원래 타임스탬프를 유지하고 SETSTAT를 사용하여 다른 파일 속성을 수정하려는 경우 Transfer Family를 통해 Amazon EFS를 백엔드 스토리지로 사용할 수 있습니다.

타입: 문자열

유효 값: DEFAULT | ENABLE_NO_OP

필수 여부: 아니요

TlsSessionResumptionMode

FTPS 프로토콜을 사용하는 Transfer Family 서버와 함께 사용되는 속성입니다. TLS 세션 재개는 FTPS 세션에 대한 통제 및 데이터 연결 간에 협상된 암호 키를 재개하거나 공유하는 메커니즘을 제공합니다. `TlsSessionResumptionMode`는 고유한 세션 ID를 통해 서버가 최근 협상된 세션을 재개할지 여부를 결정합니다. 이 속성은 `CreateServer` 및 `UpdateServer` 호출 중 사용할 수 있습니다. `CreateServer` 중에 `TlsSessionResumptionMode` 값을 지정하지 않으면 기본적으로 `ENFORCED`로 설정됩니다.

- `DISABLED`: 서버가 TLS 세션 재개 클라이언트 요청을 처리하지 않고 각 요청에 대해 새 TLS 세션을 생성합니다.
- `ENABLED`: 서버가 TLS 세션 재개를 수행하는 클라이언트를 처리하고 수락합니다. 서버는 TLS 세션 재개 클라이언트 처리를 수행하지 않는 클라이언트 데이터 연결을 거부하지 않습니다.
- `ENFORCED`: 서버가 TLS 세션 재개를 수행하는 클라이언트를 처리하고 수락합니다. 서버는 TLS 세션 재개 클라이언트 처리를 수행하지 않는 클라이언트 데이터 연결을 거부합니다. 값을 `ENFORCED`로 설정하기 전에 클라이언트를 테스트합니다.

> **Note**
>
> 모든 FTPS 클라이언트가 TLS 세션 재개를 수행하는 것은 아닙니다. 따라서 TLS 세션 재개를 적용하도록 선택하면 프로토콜 협상을 수행하지 않는 FTPS 클라이언트의 연결을 방지할 수 있습니다. `ENFORCED` 값을 사용할 수 있는지 여부를 결정하려면 클라이언트를 테스트해야 합니다.

타입: 문자열

유효 값: `DISABLED | ENABLED | ENFORCED`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

S3FileLocation

워크플로에서 사용 중인 파일의 파일 위치에 대한 세부 정보를 지정합니다. S3 스토리지를 사용하는 경우에만 해당됩니다.

내용

Bucket

사용 중인 파일이 포함된 S3 버킷을 지정합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 3입니다. 최대 길이는 63입니다.

패턴: `[a-z0-9][\.\-a-z0-9]{1,61}[a-z0-9]`

필수 여부: 아니요

Etag

개체 태그는 객체의 해시입니다. ETag는 객체의 콘텐츠에 대한 변경 사항만 반영하고 메타데이터에 대한 변경을 반영하지 않습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 65536입니다.

패턴: `.+`

필수 여부: 아니요

Key

Amazon S3 생성 시 작업에 할당된 명칭 객체 키를 사용하여 객체를 검색합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 1024입니다.

패턴: `[\P{M}\p{M}]*`

필수 여부: 아니요

VersionId

파일 버전을 지정합니다.

타입: 문자열

길이 제약: 최소 길이 1. 최대 길이는 1024입니다.

패턴: `.+`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

S3InputFileLocation

고객이 입력한 Amazon S3 파일 위치를 지정합니다.
`copyStepDetails.DestinationFileLocation` 내부에서 사용하는 경우 S3 복사 대상이어야 합니다.

버킷과 키를 제공해야 합니다. 키는 경로 또는 파일을 나타낼 수 있습니다. 키 값 끝낼 슬래시 (/) 문자로 끝낼지 여부에 따라 결정됩니다. 마지막 문자가 “/”인 경우, 파일은 폴더에 복사되며 명칭은 변경되지 않습니다. 최종 문자가 영숫자인 경우, 업로드한 파일은 경로 값으로 명칭이 바뀝니다. 이 경우 같은 명칭의 파일이 이미 있으면 덮어씁니다.

예를 들어 경로가 `shared-files/bob/`인 경우 업로드한 파일은 `shared-files/bob/`, 폴더에 복사됩니다. 경로가 `shared-files/today`인 경우, 업로드된 각 파일이 `shared-files` 폴더에 복사되고 명칭이 `today`(으)로 지정됩니다. 각 업로드는 이전 버전의 bob 파일을 덮어씁니다.

내용

Bucket

고객 입력 파일의 S3 버킷을 지정합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 3입니다. 최대 길이는 63입니다.

패턴: `[a-z0-9][\.\-a-z0-9]{1,61}[a-z0-9]`

필수 여부: 아니요

Key

Amazon S3 생성 시 작업에 할당된 명칭 객체 키를 사용하여 객체를 검색합니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 1024입니다.

패턴: `[\P{M}\p{M}]*`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

S3StorageOptions

서버에 구성된 Amazon S3 스토리지 옵션.

내용

DirectoryListingOptimization

Amazon S3 디렉터리의 성능이 최적화되었는지 여부를 지정합니다. 이 옵션은 기본적으로 비활성화되어 있습니다.

기본적으로 홈 디렉터리 매핑에는 a가 있습니다. `TYPE DIRECTORY` 이 옵션을 활성화한 경우 매핑에 파일 대상이 포함되도록 하려면 `HomeDirectoryMapEntry` `Type` 를 `FILE` 명시적으로 설정해야 합니다.

타입: 문자열

유효 값: `ENABLED | DISABLED`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

S3Tag

태깅 단계 실행 중에 파일에 할당되는 키-값 쌍을 지정합니다.

내용

Key

생성한 태그에 할당한 명칭입니다.

타입: 문자열

길이 제약: 최소 길이는 1. 최대 길이 128.

패턴: `([\p{L}\p{Z}\p{N}_.:/=+\-@]*)`

필수 여부: 예

Value

키에 해당하는 값입니다.

타입: 문자열

길이 제약: 최소 길이는 0. 최대 길이는 256입니다.

패턴: `([\p{L}\p{Z}\p{N}_.:/=+\-@]*)`

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

ServiceMetadata

워크플로와 관련된 세션 세부 정보를 위한 컨테이너 객체입니다.

내용

UserDetails

서버 ID(`ServerId`), 세션 ID(`SessionId`) 및 사용자(`UserName`)는 `UserDetails`를(을) 구성합니다.

타입: UserDetails 객체

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

SftpConnectorConfig

SFTP 커넥터 객체에 대한 세부 정보가 들어 있습니다. 커넥터 객체는 파트너 SFTP 서버 사이의 파일 전송에 사용됩니다.

> **Note**
>
> SftpConnectorConfig데이터 유형은 SFTP 커넥터와 해당 매개변수를 생성하고 업데이트하는 데 모두 사용되므로 필수 UserSecretId 사항이 아닌 것으로 표시됩니다. TrustedHostKeys 기존 SFTP 커넥터를 업데이트할 때는 필요하지 않지만 새 SFTP 커넥터를 만들 때는 필요하므로 약간 오해의 소지가 있습니다.

내용

TrustedHostKeys

연결 중인 외부 서버를 식별하는 데 사용되는 호스트 키 또는 키의 공개 부분입니다. SFTP 서버에 대해 ssh-keyscan 명령을 사용하여 필요한 키를 검색할 수 있습니다.

세 가지 표준 SSH 퍼블릭 키 형식 요소는 <key type>, <body base64>, <comment>(옵션)이며 각 요소 사이에 공백이 있습니다. <key type>과 <body base64>만 지정하세요. 키의 <comment> 부분은 입력하지 마세요.

신뢰할 수 있는 호스트 키의 경우 RSA 및 ECDSA 키를 AWS Transfer Family 허용합니다.

- RSA 키의 경우, <key type> 문자열은 ssh-rsa입니다.
- ECDSA 키의 경우, 생성한 키 크기에 따라 <key type> 문자열은 ecdsa-sha2-nistp256, ecdsa-sha2-nistp384 또는 ecdsa-sha2-nistp521입니다.

이 명령을 실행하여 SFTP 서버 이름이 있는 SFTP 서버 호스트 키를 검색하십시오. ftp.host.com

```
ssh-keyscan ftp.host.com
```

그러면 공개 호스트 키가 표준 출력으로 인쇄됩니다.

```
ftp.host.com ssh-rsa AAAAB3Nza...<long-string-for-public-key
```

이 문자열을 복사하여 create-connector 명령 TrustedHostKeys 필드 또는 콘솔의 신뢰할 수 있는 호스트 키 필드에 붙여넣습니다.

유형: 문자열 어레이

배열 멤버: 최소 항목 수는 1개입니다. 최대 항목 수는 10개입니다.

길이 제약 조건: 최소 길이는 1입니다. 최대 길이는 2,048.

필수 여부: 아니요

UserSecretId

SFTP 사용자의 개인 키, 암호 또는 둘 다를 포함하는 (AWS Secrets Manager의) 암호 식별자입니다. 식별자는 암호의 Amazon 리소스 이름(ARN)이어야 합니다.

타입: 문자열

길이 제약: 최소 길이는 1. 최대 길이는 2,048.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

SshPublicKey

특정 File Transfer 프로토콜 지원 서버(`ServerId`로 식별됨)에 대한 Transfer Family 사용자와 연결된 퍼블릭 Secure Shell(SSH) 키에 대한 정보를 제공합니다. 반환된 정보에는 키를 가져온 날짜, 퍼블릭 키 내용 및 퍼블릭 키 ID가 포함됩니다. 사용자는 사용자 이름과 연결된 둘 이상의 SSH 퍼블릭 키를 특정 서버에 저장할 수 있습니다.

내용

DateImported

퍼블릭 키가 Transfer Family 사용자 계정에 추가된 날짜를 지정합니다.

타입: Timestamp

필수 여부: 예

SshPublicKeyBody

`PublicKeyId`로 지정된 SSH 퍼블릭 키의 내용을 지정합니다.

AWS Transfer Family RSA, ECDSA 및 ED25519 키를 사용할 수 있습니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 2048입니다.

필수 여부: 예

SshPublicKeyId

퍼블릭 키의 식별자를 포함하는 `SshPublicKeyId` 파라미터를 지정합니다.

타입: 문자열

길이 제약 조건: 고정 길이는 21입니다.

패턴: `key-[0-9a-f]{17}`

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

Tag

특정 리소스의 키-값 쌍을 생성합니다. 태그는 다양한 목적으로 리소스를 검색하고 그룹화하는 데 사용할 수 있는 메타데이터입니다. 서버, 사용자 및 역할에 태그를 적용할 수 있습니다. 태그는 복수의 값을 가질 수 있습니다. 예를 들어, 계정 관리를 위해 서버를 그룹화하려면 `Group`(이)라는 태그를 만들고 해당 그룹에 값 `Research` 및 `Accounting`를 할당할 수 있습니다.

내용

Key

생성한 태그에 할당한 명칭입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 128입니다.

필수 여부: 예

Value

생성한 키 이름에 할당한 하나 이상의 값을 포함합니다.

타입: 문자열

길이 제약: 최소 길이는 0. 최대 길이는 256.

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

TagStepDetails

각 단계 타입에는 자체의 `StepDetails` 구조가 있습니다.

워크플로 단계를 실행하는 동안 파일에 태그를 지정하는 데 사용되는 키/값 쌍입니다.

내용

Name

단계의 명칭, 식별자로 사용됨.

타입: 문자열

길이 제약 조건: 최소 길이는 0입니다. 최대 길이는 30입니다.

패턴: `[\w-]*`

필수 여부: 아니요

SourceFileLocation

워크플로 단계의 입력으로 사용할 파일(이전 단계의 출력 또는 워크플로에 처음 업로드한 파일)을 지정합니다.

- 이전 파일을 입력으로 사용하려면 `${previous.file}`를 입력합니다. 이 경우 이 워크플로 단계에서는 이전 워크플로 단계의 출력 파일을 입력으로 사용합니다. 이것이 기본값입니다.
- 시초로 업로드된 파일 위치를 이 단계의 입력으로 사용하려면 `${original.file}`를 입력합니다.

타입: 문자열

길이 제약: 최소 길이는 0. 최대 길이는 256입니다.

패턴: `\$\{(\w+.)+\w+\}`

Required: No

Tags

1~10개의 키/값 쌍을 포함하는 배열.

타입: S3Tag 객체 배열

어레이 멤버: 최소 항목 수 1개. 최대 항목 수 10개.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

UserDetails

워크플로에 대한 사용자 이름, 서버 ID 및 세션 ID를 지정합니다.

내용

ServerId

전송 서버 인스턴스에 대한 시스템에서 할당한 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `s-([0-9a-f]{17})`

필수 사항 여부: Yes

UserName

서버와 연결된 Transfer Family 사용자를 식별하는 고유한 문자열입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 3입니다. 최대 길이는 100입니다.

패턴: `[\w][\w@.-]{2,99}`

필수 사항 여부: Yes

SessionId

워크플로에 해당하는 세션에 대해 시스템에서 할당한 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 최소 길이는 3입니다. 최대 길이 32.

패턴: `[\w-]*`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

WorkflowDetail

할당할 워크플로의 워크플로 ID와 워크플로 실행에 사용되는 실행 역할을 지정합니다.

파일이 완전히 업로드될 때 실행되는 워크플로 외에도 `WorkflowDetails`에는 부분 업로드 시 실행할 워크플로의 워크플로 ID와 실행 역할이 포함될 수 있습니다. 파일이 업로드되는 동안 서버 세션 연결이 끊기면 부분 업로드가 수행됩니다.

내용

ExecutionRole

모든 워크플로 단계가 필요한 리소스에서 작동할 수 있도록 Transfer가 맡을 수 있는 S3, EFS 및 Lambda 작업에 필요한 권한을 포함합니다.

타입: 문자열

길이 제약: 최소 길이는 20. 최대 길이는 2,048.

패턴: `arn:.*role/\S+`

필수 여부: 예

WorkflowId

워크플로의 고유 식별자입니다.

타입: 문자열

길이 제약 조건: 고정 길이는 19입니다.

패턴: `w-([a-z0-9]{17})`

필수 여부: 예

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

WorkflowDetails

`WorkflowDetail` 데이터 타입에 대한 컨테이너입니다. 실행을 시작하도록 워크플로를 트리거하는 작업에서 사용됩니다.

내용

OnPartialUpload

파일이 부분적으로만 업로드된 경우 워크플로를 시작하는 트리거입니다. 부분 업로드가 있을 때마다 실행되는 서버에 워크플로를 연결할 수 있습니다.

세션 연결이 끊길 때 파일이 열려 있으면 부분 업로드가 수행됩니다.

> **Note**
>
> `OnPartialUpload`최대 한 개의 `WorkflowDetail` 객체를 포함할 수 있습니다.

유형: WorkflowDetail객체 어레이

어레이 멤버: 최소 항목 수 0개. 최대 항목 수는 1개입니다.

필수 여부: 아니요

OnUpload

워크플로를 시작하는 트리거: 파일이 업로드된 후 워크플로가 실행되기 시작합니다.

서버에서 연결된 워크플로를 제거하려면 다음 예제와 같이 빈 `OnUpload` 객체를 제공하면 됩니다.

```
aws transfer update-server --server-id s-01234567890abcdef --workflow-details '{"OnUpload":[]}'
```

> **Note**
>
> `OnUpload`최대 한 개의 `WorkflowDetail` 개체를 포함할 수 있습니다.

유형: WorkflowDetail객체 어레이

어레이 멤버: 최소 항목 수 0개. 최대 항목 수는 1개입니다.

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

WorkflowStep

워크플로의 기본 구성 요소입니다.

내용

CopyStepDetails

파일 복사를 수행하는 단계에 대한 세부 정보입니다.

다음 값으로 구성됩니다.

- 설명
- 파일 사본의 대상에 대한 Amazon S3 위치입니다.
- 같은 이름의 기존 파일을 덮어쓸지 여부를 나타내는 플래그입니다. 기본값은 FALSE입니다.

타입: CopyStepDetails 객체

필수 항목 여부: 아니요

CustomStepDetails

AWS Lambda 함수를 호출하는 단계에 대한 세부 정보입니다.

Lambda 함수의 이름, 대상 및 제한 시간(초)으로 구성됩니다.

타입: CustomStepDetails 객체

필수 여부: 아니요

DecryptStepDetails

암호화된 파일의 암호를 복호화하는 단계에 대한 세부 정보입니다.

다음 값으로 구성됩니다.

- 설명이 포함된 이름을 지정합니다.
- 복호화할 소스 파일의 Amazon S3 또는 Amazon Elastic File System(Amazon EFS)의 위치입니다.
- 파일 복호화의 대상에 대한 S3 또는 Amazon EFS 위치입니다.
- 같은 명칭의 기존 파일을 덮어쓸지 여부를 나타내는 플래그입니다. 기본값은 FALSE입니다.
- 사용되는 암호화 타입입니다. 현재는 PGP 암호화만 지원됩니다.

타입: DecryptStepDetails 객체

필수 여부: 아니요

DeleteStepDetails

파일을 삭제하는 단계에 대한 세부 정보입니다.

타입: DeleteStepDetails 객체

필수 여부: 아니요

TagStepDetails

하나 이상의 태그를 생성하는 단계에 대한 세부 정보입니다.

태그를 하나 이상 지정합니다. 각 태그에는 키-값 쌍이 포함되어 있습니다.

타입: TagStepDetails 객체

필수 여부: 아니요

Type

현재 다음과 같은 단계 타입이 지원됩니다.

- **COPY** - 다른 위치에 파일을 복사합니다.
- **CUSTOM**- AWS Lambda 함수 대상을 사용하여 사용자 지정 단계를 수행합니다.
- **DECRYPT** - 업로드되기 전에 암호화된 파일을 복호화합니다.
- **DELETE** - 파일을 삭제합니다.
- **TAG** - 파일에 태그를 추가합니다.

타입: 문자열

유효 값: `COPY | CUSTOM | TAG | DELETE | DECRYPT`

필수 여부: 아니요

참고

언어별 AWS SDK 중 하나에서 이 API를 사용하는 방법에 대한 자세한 내용은 다음을 참조하십시오.

- AWS SDK for C++
- AWS Java V2용 SDK
- AWS 루비 V3용 SDK

API 요청 만들기

콘솔의 사용 외에도, AWS Transfer Family API를 사용하여 서버를 프로그래밍 방식으로 구성 및 관리할 수 있습니다. 이 섹션은 AWS Transfer Family 작동, 인증 요청 확인 및 오류 처리를 설명합니다. Transfer Family에서 사용할 수 있는 지역 및 엔드포인트에 대한 자세한 설명은 AWS 일반 참조의 AWS Transfer Family엔드포인트 및 할당량을 참조하세요.

> **Note**
>
> Transfer Family로 애플리케이션을 개발할 때 AWS SDK를 사용할 수도 있습니다. Java, .NET 및 PHP용 AWS SDK는 프로그래밍 작업을 간소화하며 기본 Transfer Family API를 포함합니다. SDK 라이브러리 다운로드에 대한 정보는 샘플 코드 라이브러리 섹션을 참조하세요.

주제

Transfer Family 필수 요청 헤더

이 섹션에서는 AWS Transfer Family에 대한 모든 POST 요청과 함께 전송해야 하는 필수 헤더에 대해 설명합니다. 호출하려는 작업을 포함하는 요청에 대한 핵심 정보, 요청 날짜 및 요청 전송자의 권한을 부여함을 나타내는 정보를 식별할 HTTP 헤더를 포함해야 합니다. 헤더는 대소문자를 구별하고 헤더의 순서는 중요하지 않습니다.

다음은 ListServers 작업에서 사용하는 헤더의 예입니다.

```
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}
```

다음은 Transfer Family에 대한 POST 요청과 함께 포함해야 하는 헤더입니다. "x-amz"로 시작하는 아래의 헤더는 AWS에 고유한 헤더입니다. 나머지 헤더는 HTTP 트랜잭션에 사용되는 공통 헤더입니다.

헤더	설명
`Authorization`	승인 헤더가 필요합니다. 형식은 표준 Sigv4 요청 서명이며, 이는 서명 AWS API 요청에 설명되어 있습니다.
`Content-Type`	Transfer Family에 대한 모든 요청의 콘텐츠 타입으로 `application/x-amz-json-1.1` 를 사용하세요. `Content-Type: application/x-amz-json-1.1`
`Host`	호스트 헤더를 사용하여 요청을 전송하는 Transfer Family 엔드포인트를 지정합니다. 예컨대, `transfer.us-east-1.amazonaws.com` 은 미국 동부(오하이오) 지역의 엔드포인트입니다. Transfer Family에 사용할 수 있는 엔드포인트에 대한 자세한 설명은 AWS 일반 참조의 AWS Transfer Family 엔드포인트 및 할당량을 참조하세요. `Host: transfer. region.amazonaws.com`
`x-amz-date`	HTTP `Date` 헤더 또는 AWS `x-amz-date` 헤더에 타임스탬프를 제공해야 합니다. 일부 HTTP 클라이언트 라이브러리에서는 `Date` 헤더를 설정할 수 없습니다. `x-amz-date` 헤더가 있으면 Transfer Family는 요청 인증 중 모든 `Date` 헤더를 무시합니다. `x-amz-date` 형식은 YYYYMMDD'T'HHMMSS'Z' 형식의 ISO8601이어야 합니다. `x-amz-date: YYYYMMDD'T'HHMMSS'Z'`
`x-amz-target`	이 헤더는 API의 버전과 요청 중인 작업을 지정합니다. 대상 헤더 값은 API 버전을 API 이름과 연결하여 구성하며 형식은 다음과 같습니다.

헤더	설명
	`x-amz-target: TransferService.` *operationName* OperationName 값(예: `ListServers`)은 API 목록인 ListServers에서 찾을 수 있습니다.
`x-amz-security-token`	이 헤더는 요청에 서명하는 데 사용되는 자격 증명이 임시 또는 세션 자격 증명인 경우 필요합니다(자세한 설명은 IAM 사용자 가이드의 AWS 리소스로 임시 자격 증명 사용 참조). 자세한 설명은 Amazon Web Services 일반 참조의 HTTP 요청에 서명 추가를 참조하세요.

Transfer Family 요청 입력 및 서명

모든 요청 입력은 요청 본문의 JSON 페이로드의 일부로 전송되어야 합니다. 모든 요청 필드가 선택 사항(예: `ListServers`)인 작업의 경우 요청 본문에 `{}`와 같이 빈 JSON 객체를 제공해야 합니다. Transfer Family 페이로드 요청/응답의 구조는 기존 API 참조(예: DescribeServer)에 문서화되어 있습니다.

Transfer Family는 AWS 서명 버전 4를 이용한 인증을 지원합니다. 자세한 설명은 AWS API 요청 서명을 참조하세요.

오류 응답

오류가 있는 경우, 응답 헤더 정보에는 다음 내용이 포함됩니다.

- 콘텐츠 타입: `application/x-amz-json-1.1`
- 적절한 `4xx` 또는 `5xx` HTTP 상태 코드

오류 응답의 본문에는 발생한 오류에 대한 정보가 포함됩니다. 다음 샘플 오류 응답은 모든 오류 응답에 공통된 응답 요소의 출력 구문을 나타냅니다.

```
{
    "__type": "String",
    "Message": "String", <!-- Message is lowercase in some instances -->
    "Resource": String,
    "ResourceType": String
```

```
    "RetryAfterSeconds": String
}
```

다음 표는 이전 구문에 표시된 JSON 오류 응답 필드를 설명합니다.

__타입

Transfer Family API 호출의 예외 중 하나입니다.

타입: 문자열

메시지 또는 메시지

작업 오류 코드 메시지 중 하나입니다.

> **Note**
>
> 일부 예외에서 `message`가 사용되고 다른 예외에서 `Message`가 사용됩니다. 인터페이스의 코드를 확인하여 적절한 대/소문자를 확인할 수 있습니다. 또는 각 옵션을 테스트하여 어떤 옵션이 효과가 있는지 확인할 수 있습니다.

타입: 문자열

리소스

오류가 호출된 리소스입니다. 예컨대, 이미 존재하는 사용자를 만들려고 하면 `Resource`가 기존 사용자의 사용자 이름이 됩니다.

타입: 문자열

ResourceType

오류가 호출되는 리소스 타입입니다. 예컨대, 이미 존재하는 사용자를 만들려고 하면 `ResourceType`이 `User`가 됩니다.

타입: 문자열

RetryAfterSeconds

명령을 재시도하기 전에 대기하는 시간(초).

타입: 문자열

오류 응답 예

`DescribeServer` API를 호출하고 존재하지 않는 서버를 지정하면 다음 JSON 본문이 반환됩니다.

```
{
  "__type": "ResourceNotFoundException",
  "Message": "Unknown server",
  "Resource": "s-11112222333344444",
  "ResourceType": "Server"
}
```

API 실행 시 병목 현상이 발생하는 경우 다음 JSON 본문이 반환됩니다.

```
{
   "__type":"ThrottlingException",
   "RetryAfterSeconds":"1"
}
```

`CreateServer` API를 사용하고 Transfer Family 서버를 생성할 수 있는 충분한 권한이 없는 경우 다음 JSON 본문이 반환됩니다.

```
{
  "__type": "AccessDeniedException",
  "Message": "You do not have sufficient access to perform this action."
}
```

`CreateUser` API를 사용하고 이미 존재하는 사용자를 지정하면 다음 JSON 본문이 반환됩니다.

```
{
  "__type": "ResourceExistsException",
  "Message": "User already exists",
  "Resource": "Alejandro-Rosalez",
  "ResourceType": "User"
}
```

사용 가능한 라이브러리

AWS에서는 command-line 도구 및 Query API 대신 언어별로 고유한 API를 사용하여 애플리케이션을 빌드하는 것을 선호하는 소프트웨어 개발자를 위해 라이브러리, 샘플 코드, 자습서 및 기타 리소스를

제공합니다. 이러한 라이브러리는 보다 쉽게 시작하도록 요청 인증, 요청 재시도 및 오류 처리 같은 기본 기능(API에는 포함되지 않음)을 제공합니다. AWS 기반 도구를 참조하세요.

모든 언어의 라이브러리와 샘플 코드는 샘플 코드 및 라이브러리를 참조하세요.

공통 파라미터

다음 목록에는 모든 작업이 쿼리 문자열을 사용하여 Signature Version 4 요청에 서명하는 데 사용하는 파라미터가 포함되어 있습니다. 작업별 파라미터는 그 작업에 대한 항목에 나열되어 있습니다. Signature Version 4에 대한 자세한 내용은 IAM 사용 설명서의 AWS API 요청에 서명을 참조하세요.

Action

수행할 작업입니다.

유형: 문자열

필수 항목 여부: 예

Version

요청이 작성되는 API 버전으로 YYYY-MM-DD 형식으로 표시됩니다.

유형: 문자열

필수 항목 여부: 예

X-Amz-Algorithm

요청 서명을 생성하는 데 사용된 해시 알고리즘입니다.

조건: HTTP 권한 부여 헤더 대신 쿼리 문자열에 인증 정보를 포함하는 경우 이 파라미터를 지정합니다.

유형: 문자열

유효한 값: `AWS4-HMAC-SHA256`

필수 항목 여부: 조건부

X-Amz-Credential

자격 증명 범위 값이며 액세스 키, 날짜, 대상으로 하는 리전, 요청하는 서비스 및 종료 문자열("aws4_request")이 포함된 문자열입니다. 값은 다음 형식으로 표시됩니다. access_key/YYYYMMDD/region/service/aws4_request.

자세한 내용은 IAM 사용 설명서의 서명된 AWS API 요청 생성을 참조하세요.

조건: HTTP 권한 부여 헤더 대신 쿼리 문자열에 인증 정보를 포함하는 경우 이 파라미터를 지정합니다.

유형: 문자열

필수 항목 여부: 조건부

X-Amz-Date

서명을 만드는 데 사용되는 날짜입니다. 형식은 ISO 8601 기본 형식(YYYYMMDD'T'HHMMSS'Z') 이어야 합니다. 예를 들어 다음 날짜 시간은 유효한 X-Amz-Date 값: `20120325T120000Z`.

조건: X-Amz-Date는 모든 요청에서 옵션이지만 서명 요청에 사용되는 날짜보다 우선할 때 사용됩니다. 날짜 헤더가 ISO 8601 기본 형식으로 지정된 경우 X-Amz-Date가 필요하지 않습니다. X-Amz-Date를 사용하는 경우 항상 Date 헤더의 값을 재정의합니다. 자세한 내용은 IAM 사용 설명서의 AWS API 요청 서명의 요소를 참조하세요.

유형: 문자열

필수 항목 여부: 조건부

X-Amz-Security-Token

AWS Security Token Service(AWS STS)에 대한 호출을 통해 받은 임시 보안 토큰입니다. AWS STS의 임시 보안 인증 정보를 지원하는 서비스 목록은 IAM 사용 설명서의 IAM으로 작업하는 AWS 서비스를 참조하세요.

조건: AWS STS의 임시 보안 인증 정보를 사용하는 경우 보안 토큰을 포함시켜야 합니다.

유형: 문자열

필수 항목 여부: 조건부

X-Amz-Signature

서명할 문자열과 파생된 서명 키에서 계산된 16진수로 인코딩된 서명을 지정합니다.

조건: HTTP 권한 부여 헤더 대신 쿼리 문자열에 인증 정보를 포함하는 경우 이 파라미터를 지정합니다.

유형: 문자열

필수 항목 여부: 조건부

X-Amz-SignedHeaders

표준 요청의 일부로 포함된 모든 HTTP 헤더를 지정합니다. 서명된 헤더 지정에 대한 자세한 내용은 IAM 사용 설명서의 서명된 AWS API 요청 생성을 참조하세요.

조건: HTTP 권한 부여 헤더 대신 쿼리 문자열에 인증 정보를 포함하는 경우 이 파라미터를 지정합니다.

유형: 문자열

필수 항목 여부: 조건부

일반적인 오류

이 단원에는 모든 AWS 서비스의 API 작업에 대한 일반 오류가 나와 있습니다. 이 서비스의 API 작업에 대한 오류는 해당 API 작업 항목을 참조하십시오.

AccessDeniedException

이 작업을 수행할 수 있는 충분한 액세스 권한이 없습니다.

HTTP 상태 코드: 400

IncompleteSignature

요청 서명이 AWS 표준을 준수하지 않습니다.

HTTP 상태 코드: 400

InternalFailure

알 수 없는 오류, 예외 또는 장애 때문에 요청 처리가 실패했습니다.

HTTP 상태 코드: 500

InvalidAction

요청된 동작 또는 작업이 유효하지 않습니다. 작업을 올바로 입력했는지 확인합니다.

HTTP 상태 코드: 400

InvalidClientTokenId

제공된 X.509 인증서 또는 AWS 액세스 키 ID가 AWS의 레코드에 존재하지 않습니다.

HTTP 상태 코드: 403

NotAuthorized

이 작업을 수행하려면 권한이 있어야 합니다.

HTTP 상태 코드: 400

OptInRequired

AWS 액세스 키 ID는 서비스에 대한 구독이 필요합니다.

HTTP 상태 코드: 403

RequestExpired

요청이 요청상의 날짜 스탬프로부터 15분 이상, 또는 요청 만료 날짜(예: 미리 서명된 URL)로부터 15분 이상 경과한 후 서비스에 도달했거나, 요청상의 날짜 스탬프가 15분 이상 미래입니다.

HTTP 상태 코드: 400

ServiceUnavailable

서버의 일시적 장애로 인해 요청이 실패하였습니다.

HTTP 상태 코드: 503

ThrottlingException

요청 제한 때문에 요청이 거부되었습니다.

HTTP 상태 코드: 400

ValidationError

입력이 AWS 서비스에서 지정한 제약에 충족되지 않습니다.

HTTP 상태 코드: 400

에 대한 문서 기록 AWS Transfer Family

다음 표에서는 이번 릴리스의 설명서를 설명합니다 AWS Transfer Family.

- API 버전: transfer-2018-11-05
- 최신 설명서 업데이트: 2024년 4월 23일

변경 사항	설명	날짜
SFTP 커넥터가 원격 파일 및 디렉터리를 나열하는 기능	Transfer Family는 고객이 SFTP 커넥터를 사용하여 원격 SFTP 서버에 저장된 파일을 나열할 수 있는 기능을 추가했습니다. 자세한 내용은 원격 디렉터리 콘텐츠 목록을 참조하세요.	2024년 4월 23일
거래 파트너의 자체 서명 TLS 인증서를 AS2 메시지 교환과 함께 사용할 수 있습니다.	AWS Transfer Family 거래 파트너의 자체 서명 공개 TLS 인증서를 가져와 사용하여 HTTPS를 통해 적용 가능성 설명 2 (AS2) 메시지를 서버에 전송하는 옵션이 추가되었습니다.	2024년 4월 12일
SFTP 커넥터에 대한 보안 정책 추가	AWS Transfer Family SFTP 커넥터와 함께 사용하기 위한 보안 정책이 추가되었습니다. 자세한 내용은 SFTP AWS Transfer Family 커넥터에 대한 보안 정책단원을 참조하세요.	2024년 4월 5일
아마존과 통합 EventBridge	AWS Transfer Family 이제 모든 파일 전송 작업에 EventBridge 대한 이벤트를 Amazon에 자동으로 게시합니다. 자세한	2024년 2월 8일

변경 사항	설명	날짜
	내용은 를 사용하여 Transfer Family 이벤트 관리 Amazon EventBridge단원을 참조하세요.	
새 보안 정책 추가	AWS Transfer Family 새로운 FIPS 및 비 FIPS 보안 정책이 추가되었습니다. 또한 서버에 할당되는 기본 보안 정책은 항상 최신 보안 정책입니다. 자세한 내용은 서버 보안 정책 AWS Transfer Family단원을 참조하세요.	2024년 2월 5일
SFTP 커넥터 및 AS2의 고정 IP 주소 지원	이제 Transfer Family는 SFTP 커넥터 및 AS2에 대한 고정 IP 주소를 제공합니다. 이를 통해 IP 허용 목록 제어로 보호되는 원격 SFTP 서버와 연결할 수 있습니다. AS2의 경우 AS2 서버의 비동기 MDN 응답을 위한 고정 IP 주소를 도입합니다.	2024년 1월 16일
사용 설명서가 의 최신 버전에 더 가깝게 조정되도록 재구성되었습니다. AWS Transfer Family	Transfer Family는 안내서가 작성된 이후 여러 기능을 추가했으므로 안내서를 재구성해야 했습니다.	2024년 1월 3일

변경 사항	설명	날짜
논리적 디렉터리 매핑 개선 사항 Amazon S3 목록 성능 최적화	Transfer Family는 이제 최대 2.1MB의 논리적 디렉터리 매핑을 지원합니다. 이제 파일에 대한 사용자 매핑 여부도 선언할 수 있습니다. 자세한 정보는 논리적 디렉터리 사용 규칙을 참조하세요. Amazon S3를 스토리지로 사용하는 서버를 만들거나 업데이트할 때 이제 S3 디렉터리 (또는 폴더) 를 나열하는 성능을 최적화할 수 있습니다. 자세한 정보는 SFTP, FTPS 또는 FTP 서버 엔드포인트 구성을 참조하세요.	2023년 11월 17일
VPC (가상 사설 클라우드) 엔드포인트가 있는 SFTP 서버용 대체 포트	이제 VPC 엔드포인트가 있는 SFTP Transfer Family 서버에 대해 대체 비표준 포트를 활성화할 수 있습니다. 자세한 정보는 Virtual Private Cloud(VPC)에 서버 생성을 참조하세요.	2023년 11월 17일
SFTP 커넥터 지원	SFTP 커넥터는 클라우드 및 온프레미스의 원격 서버와 AWS Transfer Family 통신할 수 있도록 의 기능을 확장합니다. 자세한 정보는 SFTP 커넥터를 사용하여 파일을 보내고 검색합니다.을 참조하세요.	2023년 7월 25일

변경 사항	설명	날짜
AS2 기본 인증 지원	Transfer Family는 이제 적용성 보고서 2(AS2) 프로토콜을 사용하는 서버에 대해 기본 인증 사용을 지원합니다. 자세한 내용은 AS2 커넥터의 기본 인증를 참조하세요.	2023년 6월 30일
정형 JSON 로깅 지원	이제 Transfer Family는 Amazon에 구조화된 JSON 로그를 전송하고 CloudWatch, 로그 스트림을 사용자 지정 로그 그룹으로 그룹화하고, 프로토콜 간에 일반적인 로그 쿼리를 수행하는 것을 지원합니다. 자세한 정보는 아마존 CloudWatch 로깅 대상 AWS Transfer Family을 참조하세요.	2023년 6월 24일
다양한 인증 방법 지원	Transfer Family는 암호, 퍼블릭/프라이빗 키 쌍 또는 둘 다를 사용한 인증을 지원합니다. 이는 SFTP 프로토콜을 사용하는 서버와 사용자 지정 자격 증명 공급자에 사용할 수 있습니다. 자세한 내용은 SFTP 지원 서버 생성를 참조하세요.	2023년 5월 17일

변경 사항	설명	날짜
Transfer Family가 워크플로를 통해 처리하는 파일에 대한 Pretty Good Privacy(PGP) 암호 해독 지원	Transfer Family는 Pretty Good Privacy(PGP) 복호화를 기본적으로 지원합니다. SFTP, FTPS 또는 FTP를 통해 Amazon Simple Storage Service(S3) 또는 Amazon Elastic File System(Amazon EFS)에 업로드된 파일에서 PGP 복호화를 사용할 수 있습니다. 자세한 내용은 PGP 키 생성 및 관리 및 워크플로에서 PGP 복호화 사용를 참조하세요.	2022년 12월 21일
Transfer Family를 통한 적용성 보고서 2(AS2) 파일 전송 프로토콜의 완전관리형 지원	환경 내부 또는 외부의 거래 파트너와 정보를 주고 받는 데 AS2 프로토콜을 사용하는 서버를 만들 수 있습니다. AWS 자세한 정보는 AS2 구성을 참조하세요.	2022년 7월 25일
서버 생성 시 디스플레이 배너 지원	서버를 생성할 때 사용자 지정 메시지를 추가할 수 있습니다. 사전 인증 메시지(모든 프로토콜) 와 사후 인증 메시지(FTP 및 FTPS 서버용)를 표시할 수 있습니다. 자세한 내용은 SFTP 지원 서버 생성, FTPS 지원 서버 생성 또는 FTP 지원 서버 생성를 참조하세요.	2022년 2월 17일

변경 사항	설명	날짜
ID AWS Lambda 제공자로서의 지원	이제 Transfer Family 서버를 AWS Lambda 사용하여 사용자 지정 ID 공급자에 연결할 수 있습니다. 이전에는 사용자 지정 자격 증명 공급자를 통합하려면 Amazon API Gateway URL을 제공해야 했습니다. 자세한 내용은 ID AWS Lambda 제공자를 통합하는 데 사용를 참조하세요.	2021년 11월 16일
관리형 File Transfer 워크플로 지원	관리형 File Transfer 워크플로는 현재 수동으로 수행하는 일반적인 작업에 대한 업로드 후 처리 추상화를 제공합니다. 자세한 내용은 AWS Transfer Family 관리형 워크플로를 참조하세요.	2021년 9월 2일
지원 대상 AWS Directory Service for Microsoft Active Directory	서비스 관리형 및 사용자 지정 ID 제공자 외에도 이제 인증 및 권한 부여를 위한 사용자 액세스를 관리하는 AWS Directory Service for Microsoft Active Directory 데 사용할 수 있습니다. 자세한 정보는 AWS 디렉터리 서비스 ID 제공자 사용을 참조하세요.	2021년 5월 24일

변경 사항	설명	날짜
신규 AWS 리전	AWS Transfer Family 이제 아프리카 (케이프타운) 지역에서 사용할 수 있습니다. Transfer Family 엔드포인트에 대한 자세한 내용은 AWS 일반 참조의 AWS Transfer Family 엔드포인트 및 할당량을 참조하세요.	2021년 2월 24일
신규 AWS 리전	AWS Transfer Family 이제 아시아 태평양 (홍콩) 및 중동 (바레인) 지역에서 사용할 수 있습니다. Transfer Family 엔드포인트에 대한 자세한 내용은 AWS 일반 참조의 AWS Transfer Family 엔드포인트 및 할당량을 참조하세요.	2021년 2월 17일
데이터 스토어로 Amazon EFS 지원	Transfer Family는 이제 Amazon Elastic File System(Amazon EFS) 에서 송수신하는 File Transfer를 지원합니다. Amazon EFS는 단순하고 확장 가능한 완전 관리형 탄력적 NFS 파일 시스템입니다. 자세한 내용은 Amazon EFS 파일 시스템 구성를 참조하세요.	2021년 1월 6일

변경 사항	설명	날짜
지원 대상 AWS WAF	이제 Transfer AWS WAF Family는 웹 애플리케이션과 API 작업을 공격으로부터 보호하는 데 도움이 되는 웹 애플리케이션 방화벽인 웹 애플리케이션 방화벽을 지원합니다. 자세한 정보는 웹 애플리케이션 방화벽 추가을 참조하세요.	2020년 11월 24일
Virtual Private Cloud(VPC)의 여러 보안 그룹 지원	이제 VPC의 서버에 여러 보안 그룹을 연결할 수 있습니다. 자세한 내용은 Virtual Private Cloud(VPC)에 서버 생성를 참조하세요.	2020년 10월 15일
신규 AWS 리전	이제 Transfer Family를 AWS GovCloud (US) 지역에서 이용할 수 있습니다. AWS GovCloud (US) 지역별 Transfer Family 엔드포인트에 대한 자세한 내용은 의 AWS Transfer Family 엔드포인트 및 할당량을 참조하십시오. AWS 일반 참조 AWS GovCloud (US) 지역에서의 Transfer Family 사용에 대한 자세한 내용은 AWS Transfer FamilyAWS GovCloud (US)사용자 안내서의 을 참고하십시오.	2020년 9월 30일

변경 사항	설명	날짜
이제 지원되는 암호화 알고리즘이 포함된 보안 정책을 서버에 연결할 수 있습니다.	이제 지원되는 암호화 알고리즘이 포함된 보안 정책을 서버에 연결할 수 있습니다. 자세한 내용은 서버 보안 정책 AWS Transfer Family를 참조하세요.	2020년 8월 12일
FIPS(Federal Information Processing Standard) 엔드포인트 지원	이제 북미 AWS 리전에서 FIPS 지원 엔드포인트를 사용할 수 있습니다. 사용할 수 있는 리전은 AWS 일반 참조의 AWS Transfer Family 엔드포인트 및 할당량을 참조하세요. SFTP 지원 서버 엔드포인트에 FIPS를 활성화하려면 SFTP 지원 서버 생성를 참조하세요. FTPS 지원 서버 엔드포인트에 FIPS를 활성화하려면 FTPS 지원 서버 생성를 참조하세요. FTP 지원 서버 엔드포인트에 FIPS를 활성화하려면 FTP 지원 서버 생성를 참조하세요.	2020년 8월 12일
사용자 이름 문자 길이 증가 및 추가 허용 문자	이제 사용자 이름에 @ 기호(@)와 마침표(.)를 포함할 수 있으며 최대 길이는 100자일 수 있습니다. 사용자를 추가하려면 서버 엔드포인트의 사용자 관리를 참조하세요.	2020년 8월 12일

변경 사항	설명	날짜
자동 아마존 CloudWatch 로깅 AWS Identity and Access Management (IAM) 역할 생성 지원	이제 Transfer Family는 최종 사용자 활동을 볼 수 있는 CloudWatch 로깅 IAM 역할의 자동 생성을 지원합니다. 자세한 내용은 SFTP 지원 서버 생성, FTPS 지원 서버 생성 또는 FTP 지원 서버 생성를 참조하세요.	2020년 7월 30일
AWS Transfer Family 이제 권한 부여 요소로 소스 IP를 지원합니다.	Transfer Family는 최종 사용자의 소스 IP 주소를 인증 요소로 사용할 수 있는 지원을 추가하여 SFTP(보안 파일 전송 프로토콜), FTPS(SSL을 통한 파일 전송 프로토콜) 또는 FTP(파일 전송 프로토콜)를 통한 액세스를 승인할 때 추가 보안 계층을 적용할 수 있습니다. 자세한 내용은 사용자 지정 자격 증명 공급자와 작업를 참조하세요.	2020년 6월 9일
AWS 이제 SFTP 전송이 가능해졌으며 FTP AWS Transfer Family 및 FTPS에 대한 지원이 추가되었습니다.	이제 사용자의 파일 전송에 FTPS(파일 전송 프로토콜 보안) 와 FTP(파일 전송 프로토콜) 라는 두 가지 추가 프로토콜을 사용할 수 있습니다. 사용자는 기존 보안 파일 전송 프로토콜 (SFTP) 지원 외에도 FTP를 통한 FTP (FTP over SSL) 및 일반 텍스트 FTP 기반 워크플로우를 이동, 실행 AWS, 보호 및 통합할 수 있습니다.	2020년 4월 23일

변경 사항	설명	날짜
Virtual Private Cloud(VPC) 보안 그룹 및 탄력적 IP 주소 지원	이제 보안 그룹을 사용하여 수신 IP 주소에 대한 허용 목록을 생성하여 서버에 추가 보안 계층을 제공할 수 있습니다. 엘라스틱 IP 주소를 서버의 엔드포인트와 연결할 수도 있습니다. 이렇게 하면 방화벽을 사용하는 사용자가 해당 엔드포인트에 대한 액세스를 허용할 수 있습니다. 자세한 내용은 Virtual Private Cloud(VPC)에 서버 생성를 참조하세요.	2020년 1월 10일
VPC에서의 작업 지원	이제 VPC에 서버를 만들 수 있습니다. 퍼블릭 인터넷을 거치지 않고 서버를 사용하여 클라이언트를 통해 Amazon S3 버킷과 데이터를 주고 받을 수 있습니다. 자세한 내용은 Virtual Private Cloud(VPC)에 서버 생성를 참조하세요.	2019년 3월 27일
첫 번째 버전이 출시되었습니다. AWS Transfer Family	이 최초 릴리스는 방향 설정을 포함하며, 시작 방법을 설명하고, 클라이언트 구성과 사용자 구성 및 모니터링 활동 관련 정보를 제공합니다.	2018년 11월 25일

AWS 용어집

최신 AWS 용어는 AWS 용어집 참조서의 AWS 용어집을 참조하세요.

기계 번역으로 제공되는 번역입니다. 제공된 번역과 원본 영어의 내용이 상충하는 경우에는 영어 버전이 우선합니다.