Panduan Pengguna

AWS Transfer Family

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Transfer Family: Panduan Pengguna

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara para pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan properti dari masing-masing pemilik, yang mungkin berafiliasi, terkait dengan, atau disponsori oleh Amazon, atau tidak.

Table of Contents

Apa itu AWS Transfer Family? 1
Bagaimana cara AWS Transfer Family kerja 3
Posting blog yang relevan untuk Transfer Family 5
Prasyarat 7
Wilayah, titik akhir, dan kuota 7
Mendaftar untuk AWS 7
Mengonfigurasi penyimpanan 8
Konfigurasikan bucket Amazon S3 9
Konfigurasikan sistem file Amazon EFS 13
Buat peran dan kebijakan IAM 16
Membuat peran pengguna 18
Bagaimana kebijakan sesi bekerja 21
Contoh kebijakan akses baca/tulis 24
Tutorial Transfer Family 28
Memulai dengan titik akhir server 28
Prasyarat 29
Masuk ke konsol 30
Buat server berkemampuan SFTP 30
Menambahkan pengguna terkelola layanan 31
Transfer file menggunakan klien 33
Buat Alur Kerja dekripsi 35
Langkah 1: Konfigurasikan peran eksekusi 35
Langkah 2: Buat alur kerja terkelola 37
Langkah 3: Tambahkan alur kerja ke server dan buat pengguna 38
Langkah 4: Buat key pair PGP 39
Langkah 5: Simpan kunci pribadi PGP di AWS Secrets Manager 40
Langkah 6: Enkripsi file 42
Langkah 7: Jalankan alur kerja dan lihat hasilnya 42
Buat dan gunakan konektor SFTP 43
Langkah 1: Buat sumber daya pendukung yang diperlukan 44
Langkah 2: Buat dan uji konektor SFTP 49
Langkah 3: Kirim dan ambil file menggunakan konektor SFTP 53
Prosedur untuk membuat server Transfer Family untuk digunakan sebagai server SFTP jarak jauh 56

Menggunakan penyedia identitas kustom 59
Prasyarat 59
Langkah 1: Buat CloudFormation tumpukan 60
Langkah 2: Periksa konfigurasi metode API Gateway untuk server Anda 61
Langkah 3: Lihat detail server Transfer Family 61
Langkah 4: Uji apakah pengguna Anda dapat terhubung ke server 63
Langkah 5: Uji koneksi SFTP dan transfer file 63
Langkah 6: Batasi akses ke ember 64
Perbarui Lambda jika menggunakan Amazon EFS 66
Siapkan konfigurasi AS2 67
Langkah 1: Buat sertifikat untuk AS2 69
Langkah 2: Buat server Transfer Family yang menggunakan protokol AS2 72
Langkah 3: Impor sertifikat sebagai sumber sertifikat Transfer Family 76
Langkah 4: Buat profil untuk Anda dan mitra dagang Anda 77
Langkah 5: Buat kesepakatan antara Anda dan pasangan 78
Langkah 6: Buat konektor antara Anda dan pasangan 79
Langkah 7: Uji pertukaran file melalui AS2 dengan menggunakan Transfer Family 80
Transfer Family untuk SFTP, FTPS, FTP 83
Opsi penyedia identitas 83
AWS Transfer Family matriks tipe titik akhir 85
Mengkonfigurasi titik akhir server Transfer Family 89
Buat server berkemampuan SFTP 91
Buat server berkemampuan FTPS 103
Buat server berkemampuan FTP 115
Buat server di VPC 126
Bekerja dengan nama host khusus 147
Mentransfer file melalui titik akhir server 151
Perintah SFTP/FTPS/FTP yang tersedia 154
Temukan titik akhir Amazon VPC Anda 155
Hindari `setstat` kesalahan 157
Gunakan OpenSSH 34
Gunakan WinSCP 159
Gunakan Cyberduck 33
Gunakan FileZilla 162
Gunakan klien Perl 164
Pemrosesan unggahan pasca 164

Mengelola pengguna 165
Pengguna yang dikelola layanan 167
Pengguna layanan direktori 177
Pengguna penyedia identitas khusus 194
Gunakan direktori logis 224
Aturan untuk menggunakan direktori logis 226
Menerapkan direktori logis dan `chroot` 227
Konfigurasikan contoh direktori logis 230
Konfigurasikan direktori logis untuk Amazon EFS 231
AWS Lambda Tanggapan khusus 231
Konektor SFTP 233
Konfigurasikan konektor SFTP 233
Buat konektor SFTP 234
Simpan rahasia untuk digunakan dengan konektor SFTP 242
Hasilkan dan format kunci pribadi konektor SFTP 243
Uji konektor SFTP 246
Transfer file dengan konektor SFTP 248
Kelola konektor SFTP 250
Perbarui konektor SFTP 250
Lihat detail konektor SFTP 250
Kuota untuk konektor SFTP 252
Transfer Family untuk AS2 254
Kasus penggunaan AS2 255
Konfigurasikan AS2 260
Membuat server AS2 menggunakan konsol Transfer Family 261
Buat server AS2 menggunakan template 264
Konfigurasi AS2 267
Fitur dan kemampuan AS2 273
Konfigurasikan konektor AS2 275
Buat konektor AS2 275
Algoritma konektor AS2 278
Otentikasi dasar untuk konektor AS2 279
Aktifkan otentikasi dasar untuk konektor AS2 281
Lihat detail konektor 285
Kelola mitra AS2 286
Impor sertifikat AS2 286

Rotasi sertifikat AS2 288
Buat profil AS2 290
Buat perjanjian AS2 291
Transfer pesan AS2 292
Kirim pesan AS2 293
Terima pesan AS2 294
Konfigurasikan HTTPS untuk AS2 295
Transfer file dengan konektor AS2 299
Nama dan lokasi file 300
Kode status 302
Contoh file JSON 303
Monitor AS2 305
Kode Status AS2 306
Kode kesalahan AS2 307
Mengelola alur kerja pemrosesan file 321
Buat alur kerja 323
Konfigurasikan dan jalankan alur kerja 324
Lihat detail alur kerja 327
Gunakan langkah-langkah yang telah ditentukan 329
Salin berkas 330
Dekripsi file 335
Berkas tag 341
Hapus berkas 342
Variabel bernama untuk alur kerja 343
Contoh tag dan hapus alur kerja 343
Gunakan langkah-langkah pemrosesan file khusus 348
Menggunakan beberapa fungsi Lambda secara berurutan 350
Mengakses file setelah pemrosesan kustom 350
Contoh peristiwa dikirim ke AWS Lambda saat file upload 351
Contoh fungsi Lambda untuk langkah alur kerja khusus 352
Izin IAM untuk langkah khusus 353
Kebijakan IAM untuk alur kerja 354
Hubungan kepercayaan alur kerja 356
Contoh peran eksekusi: Dekripsi, salin, dan tag 356
Contoh peran eksekusi: Jalankan fungsi dan hapus 358
Penanganan pengecualian untuk alur kerja 359

Pantau eksekusi alur kerja 360
CloudWatch logging untuk alur kerja 360
CloudWatch metrik untuk alur kerja 363
Buat alur kerja dari template 363
Menghapus alur kerja dari server Transfer Family 367
Pembatasan dan batasan 368
Mengelola server 371
Lihat daftar server 371
Hapus server 371
Lihat detail server SFTP 373
Lihat detail server AS2 374
Edit detail server 376
Edit protokol transfer file 379
Edit parameter penyedia identitas khusus 381
Edit titik akhir server 384
Edit pencatatan 385
Edit kebijakan keamanan 385
Mengubah alur kerja terkelola 387
Ubah spanduk tampilan untuk server Anda 388
Menempatkan server Anda secara online atau offline 389
Kelola kunci host server 390
Tambahkan kunci host server tambahan 391
Hapus kunci host server 392
Putar kunci host server 393
Informasi kunci host server tambahan 395
Pantau penggunaan dalam konsol 396
Mengelola kontrol akses 400
Membuat kebijakan akses bucket S3 401
Membuat kebijakan sesi 402
Mencegah pengguna berjalan `mkdir` di bucket S3 406
Logging 407
CloudTrail penebangan 407
Mengaktifkan pencatatan CloudTrail 409
Contoh entri log untuk membuat server 409
CloudWatch penebangan 411
Membuat logging untuk server 412

Mengelola logging untuk alur kerja 420
Mengkonfigurasi peran untuk CloudWatch 423
Melihat aliran log Transfer Family 425
Membuat CloudWatch alarm Amazon 429
Logging panggilan API S3 ke log akses S3 429
Contoh untuk membatasi masalah wakil yang membingungkan 430
CloudWatch struktur log untuk Transfer Family 432
Contoh entri CloudWatch log 437
Menggunakan CloudWatch metrik 441
Notifikasi pengguna 444
Mengelola acara menggunakan EventBridge 445
Transfer Family acara 446
Acara server SFTP, FTPS, dan FTP 446
Acara konektor SFTP 447
Acara A2S 447
Mengirim Transfer Family acara 448
Membuat pola acara 449
Menguji pola acara untuk Transfer Family acara 450
Izin 450
Sumber daya tambahan 451
Referensi detail acara 451
Acara server 452
Acara konektor 456
Acara AS2 461
Keamanan 467
Kebijakan keamanan untuk server 469
Algoritma kriptografi 470
TransferSecurityPolicy-2024-01 478
TransferSecurityPolicy-2023-05 479
TransferSecurityPolicy-2022-03 480
TransferSecurityPolicy-2020-06 481
TransferSecurityPolicy-2018-11 482
TransferSecurityPolicy-FIP-2024-01 483
TransferSecurityPolicy-FIP-2023-05 484
TransferSecurityPolicy-FIP-2020-06 485
Pasca kebijakan keamanan Quantum 487

Kebijakan keamanan untuk konektor SFTP 491
Kebijakan keamanan pasca-Quantum 493
Tentang pertukaran kunci hibrida pasca-kuantum di SSH 495
Cara menggunakannya 495
Bagaimana cara mengujinya 497
Perlindungan data 500
Enkripsi data 501
Manajemen kunci 502
Pengelolaan identitas dan akses 518
Audiens 519
Mengautentikasi dengan identitas 520
Mengelola akses menggunakan kebijakan 523
Bagaimana AWS Transfer Family bekerja dengan IAM 526
Contoh kebijakan berbasis identitas 531
Contoh kebijakan berbasis tanda 534
Pemecahan masalah identitas dan akses 538
Validasi kepatuhan 540
Ketangguhan 541
Keamanan infrastruktur 542
Firewall aplikasi web 542
Pencegahan confused deputy lintas layanan 544
Peran pengguna Transfer Family 545
Peran alur kerja Transfer Family 547
Peran logging/doa Transfer Family 548
AWS kebijakan terkelola 550
AWSTransferConsoleFullAccess 550
AWSTransferFullAccess 552
AWSTransferLoggingAccess 554
AWSTransferReadOnlyAccess 554
Pembaruan kebijakan 555
Pemecahan Masalah Transfer Family 557
Memecahkan masalah pengguna yang dikelola layanan 557
Memecahkan masalah pengguna yang dikelola layanan Amazon EFS 558
Memecahkan masalah badan kunci publik terlalu lama 558
Pemecahan masalah gagal menambahkan kunci publik SSH 559
Memecahkan masalah Amazon API Gateway 559

Terlalu banyak kegagalan otentikasi 559
Koneksi ditutup 561
Memecahkan masalah kebijakan untuk bucket Amazon S3 terenkripsi 561
Memecahkan masalah otentikasi 562
Kegagalan otentikasi—SSH/SFTP 562
Masalah alam tidak cocok AD terkelola 563
Masalah otentikasi lain-lain 563
Memecahkan masalah alur kerja terkelola 564
Memecahkan masalah kesalahan terkait alur kerja menggunakan Amazon CloudWatch 564
Memecahkan masalah kesalahan penyalinan alur kerja 566
Memecahkan masalah dekripsi alur kerja 566
Memecahkan masalah kesalahan untuk file enkripsi yang ditandatangani 567
Memecahkan masalah kesalahan untuk algoritma FIPS 567
Memecahkan masalah Amazon EFS 569
Memecahkan masalah profil POSIX yang hilang 569
Memecahkan masalah direktori logis dengan Amazon EFS 570
Memecahkan masalah pengujian penyedia identitas Anda 571
Memecahkan masalah menambahkan kunci host tepercaya untuk konektor SFTP Anda 571
Memecahkan masalah pengunggahan file 572
Memecahkan masalah kesalahan unggahan file Amazon S3 572
Memecahkan masalah nama file yang tidak dapat dibaca 573
Memecahkan masalah pengecualian `ResourceNotFound` 573
Memecahkan masalah konektor SFTP 574
Negosiasi kunci gagal 574
Masalah konektor SFTP lain-lain 575
Memecahkan masalah AS2 575
Referensi API 576
Selamat datang 576
Tindakan 579
CreateAccess 582
CreateAgreement 589
CreateConnector 595
CreateProfile 603
CreateServer 607
CreateUser 620
CreateWorkflow 629

DeleteAccess 638
DeleteAgreement 641
DeleteCertificate 644
DeleteConnector 646
DeleteHostKey 648
DeleteProfile 651
DeleteServer 653
DeleteSshPublicKey 656
DeleteUser 659
DeleteWorkflow 662
DescribeAccess 664
DescribeAgreement 668
DescribeCertificate 671
DescribeConnector 674
DescribeExecution 677
DescribeHostKey 682
DescribeProfile 685
DescribeSecurityPolicy 688
DescribeServer 692
DescribeUser 697
DescribeWorkflow 702
ImportCertificate 707
ImportHostKey 712
ImportSshPublicKey 716
ListAccesses 721
ListAgreements 725
ListCertificates 729
ListConnectors 733
ListExecutions 736
ListHostKeys 741
ListProfiles 745
ListSecurityPolicies 749
ListServers 753
ListTagsForResource 757
ListUsers 762
ListWorkflows 767

SendWorkflowStepState 770
StartFileTransfer 774
StartServer 780
StopServer 783
TagResource 786
TestConnection 789
TestIdentityProvider 793
UntagResource 800
UpdateAccess 803
UpdateAgreement 810
UpdateCertificate 816
UpdateConnector 820
UpdateHostKey 825
UpdateProfile 829
UpdateServer 832
UpdateUser 845
Tipe Data 852
As2ConnectorConfig 855
CopyStepDetails 859
CustomStepDetails 862
DecryptStepDetails 864
DeleteStepDetails 867
DescribedAccess 869
DescribedAgreement 873
DescribedCertificate 877
DescribedConnector 881
DescribedExecution 885
DescribedHostKey 888
DescribedProfile 891
DescribedSecurityPolicy 894
DescribedServer 897
DescribedUser 906
DescribedWorkflow 910
EfsFileLocation 912
EndpointDetails 914
ExecutionError 918

ExecutionResults 920
ExecutionStepResult 921
FileLocation 923
HomeDirectoryMapEntry 924
IdentityProviderDetails 926
InputFileLocation 929
ListedAccess 930
ListedAgreement 933
ListedCertificate 936
ListedConnector 939
ListedExecution 941
ListedHostKey 943
ListedProfile 945
ListedServer 947
ListedUser 950
ListedWorkflow 953
LoggingConfiguration 955
PosixProfile 957
ProtocolDetails 959
S3FileLocation 963
S3InputFileLocation 965
S3StorageOptions 967
S3Tag 968
ServiceMetadata 969
SftpConnectorConfig 970
SshPublicKey 972
Tag 974
TagStepDetails 975
UserDetails 977
WorkflowDetail 979
WorkflowDetails 981
WorkflowStep 983
Membuat permintaan API 985
Transfer Family membutuhkan header permintaan 985
Transfer Family meminta masukan dan penandatanganan 987
Tanggapan kesalahan 988

Pustaka yang tersedia 990
Parameter Umum 990
Kesalahan Umum 993
Riwayat dokumen 995
Daftar istilah AWS 1009
..... mx

Apa itu AWS Transfer Family?

AWS Transfer Family adalah layanan transfer aman yang memungkinkan Anda mentransfer file masuk dan keluar dari layanan AWS penyimpanan. Transfer Family adalah bagian dari AWS Cloud platform. AWS Transfer Family menawarkan dukungan terkelola penuh untuk transfer file melalui SFTP, AS2, FTPS, dan FTP langsung masuk dan keluar dari Amazon S3 atau Amazon EFS. Anda dapat memigrasi, mengotomatisasi, dan memantau alur kerja transfer file dengan mempertahankan konfigurasi sisi klien yang ada untuk autentikasi, akses, dan firewall—sehingga tidak ada perubahan bagi pelanggan, mitra, dan tim internal Anda, atau aplikasi mereka.

Lihat Memulai AWS untuk mempelajari lebih lanjut dan mulai membangun aplikasi cloud dengan Amazon Web Services.

AWS Transfer Family mendukung transfer data dari atau ke layanan AWS penyimpanan berikut.

- Penyimpanan Amazon Simple Storage Service (Amazon S3). Untuk informasi tentang Amazon S3, lihat Memulai Layanan Penyimpanan Sederhana Amazon.
- Sistem file Sistem File Jaringan Amazon Elastic File System (Amazon EFS) (NFS). Untuk informasi tentang Amazon EFS, lihat Apa itu Amazon Elastic File System?

AWS Transfer Family mendukung transfer data melalui protokol berikut:

- Protokol Transfer File Secure Shell (SSH) (SFTP): versi 3
- Protokol Transfer File Aman (FTPS)
- Protokol Transfer File (FTP)
- Pernyataan Penerapan 2 (AS2)

> **Note**
>
> Untuk koneksi data FTP dan FTPS, rentang port yang digunakan Transfer Family untuk membuat saluran data adalah 8192-8200.

Protokol transfer file digunakan dalam alur kerja pertukaran data di berbagai industri seperti layanan keuangan, perawatan kesehatan, periklanan, dan ritel, antara lain. Transfer Family menyederhanakan migrasi alur kerja transfer file ke. AWS

Berikut ini adalah beberapa kasus penggunaan umum untuk menggunakan Transfer Family dengan Amazon S3:

- Data digunakan AWS untuk diunggah dari pihak ketiga seperti vendor dan mitra.
- Distribusi data berbasis langganan dengan pelanggan Anda.
- Transfer internal dalam organisasi Anda.

Berikut ini adalah beberapa kasus penggunaan umum untuk menggunakan Transfer Family dengan Amazon EFS:

- Distribusi data
- Rantai pasokan
- Manajemen konten
- Aplikasi penyajian web

Berikut ini adalah beberapa kasus penggunaan umum untuk menggunakan Transfer Family dengan AS2:

- Alur kerja dengan persyaratan kepatuhan yang bergantung pada perlindungan data dan fitur keamanan yang dibangun ke dalam protokol
- Logistik rantai pasokan
- Alur kerja pembayaran
- Transaksi B usiness-to-business (B2B)
- Integrasi dengan sistem perencanaan sumber daya perusahaan (ERP) dan manajemen hubungan pelanggan (CRM)

Dengan Transfer Family, Anda mendapatkan akses ke server berkemampuan protokol transfer file AWS tanpa perlu menjalankan infrastruktur server apa pun. Anda dapat menggunakan layanan ini untuk memigrasikan alur kerja berbasis transfer file AWS sambil mempertahankan klien dan konfigurasi pengguna akhir Anda apa adanya. Pertama-tama Anda mengaitkan nama host Anda dengan titik akhir server, lalu menambahkan pengguna Anda dan menyediakannya dengan tingkat akses yang tepat. Setelah Anda melakukan ini, permintaan transfer pengguna Anda dilayani langsung dari titik akhir server Transfer Family Anda.

Transfer Family memberikan manfaat sebagai berikut:

- Layanan terkelola penuh yang menskalakan secara real time untuk memenuhi kebutuhan Anda.
- Anda tidak perlu memodifikasi aplikasi Anda atau menjalankan infrastruktur protokol transfer file apa pun.
- Dengan data Anda dalam penyimpanan Amazon S3 yang tahan lama, Anda dapat menggunakan native Layanan AWS untuk fungsi pemrosesan, analitik, pelaporan, audit, dan arsip.
- Dengan Amazon EFS sebagai penyimpanan data, Anda mendapatkan sistem file elastis yang dikelola sepenuhnya untuk digunakan dengan AWS Cloud layanan dan sumber daya lokal. Amazon EFS dibangun untuk menskalakan sesuai permintaan ke petabyte tanpa mengganggu aplikasi, tumbuh dan menyusut secara otomatis saat Anda menambahkan dan menghapus file. Ini membantu menghilangkan kebutuhan untuk menyediakan dan mengelola kapasitas untuk mengakomodasi pertumbuhan.
- Layanan Alur Kerja Transfer File tanpa server yang dikelola sepenuhnya yang memudahkan untuk mengatur, menjalankan, mengotomatisasi, dan memantau pemrosesan file yang diunggah menggunakan. AWS Transfer Family
- Tidak ada biaya di muka, dan Anda hanya membayar untuk penggunaan layanan.

Di bagian berikut, Anda dapat menemukan deskripsi tentang berbagai fitur Transfer Family, tutorial memulai, petunjuk terperinci tentang cara mengatur berbagai server yang diaktifkan protokol, cara menggunakan berbagai jenis penyedia identitas, dan referensi API layanan.

Untuk memulai Transfer Family, lihat berikut ini:

- Bagaimana cara AWS Transfer Family kerja
- Prasyarat
- Memulai dengan AWS Transfer Family titik akhir server

Bagaimana cara AWS Transfer Family kerja

AWS Transfer Family adalah AWS layanan terkelola penuh yang dapat Anda gunakan untuk mentransfer file masuk dan keluar dari penyimpanan Amazon Simple Storage Service (Amazon S3) atau sistem file Amazon Elastic File System (Amazon EFS) melalui protokol berikut:

- Protokol Transfer File Secure Shell (SSH) (SFTP): versi 3
- Protokol Transfer File Aman (FTPS)
- Protokol Transfer File (FTP)

- Pernyataan Penerapan 2 (AS2)

AWS Transfer Family mendukung hingga 3 Availability Zones dan didukung oleh auto scaling, armada redundan untuk koneksi Anda dan permintaan transfer. Untuk contoh tentang cara membangun redundansi yang lebih tinggi dan meminimalkan latensi jaringan dengan menggunakan perutean berbasis Latensi, lihat posting blog Minimalkan latensi jaringan dengan transfer Anda untuk server SFTP. AWS

Transfer Family Managed File Transfer Workflows (MFTW) adalah layanan Alur Kerja Transfer File tanpa server yang dikelola sepenuhnya yang memudahkan untuk mengatur, menjalankan, mengotomatisasi, dan memantau pemrosesan file yang diunggah menggunakan. AWS Transfer Family Pelanggan dapat menggunakan MFTW untuk mengotomatisasi berbagai langkah pemrosesan seperti menyalin, menandai, memindai, memfilter, mengompres/mendekompresi, dan mengenkripsi/mendekripsi data yang ditransfer menggunakan Transfer Family. Ini memberikan visibilitas ujung ke ujung untuk pelacakan dan auditabilitas. Untuk detail selengkapnya, lihat AWS Transfer Family alur kerja terkelola.

AWS Transfer Family mendukung klien protokol transfer file standar apa pun. Beberapa klien yang umum digunakan adalah sebagai berikut:

- OpenSSH — Utilitas baris perintah Macintosh dan Linux.
- WinSCP - Klien grafis khusus Windows.
- Cyberduck — Klien grafis Linux, Macintosh, dan Microsoft Windows.
- FileZilla— Klien grafis Linux, Macintosh, dan Windows.

AWS menawarkan lokakarya Transfer Family berikut.

- Buat solusi transfer file yang memanfaatkan titik akhir SFTP/FTPS terkelola serta Amazon Cognito dan DynamoDB AWS Transfer Family untuk manajemen pengguna. Anda dapat melihat detail untuk lokakarya ini di sini.
- Buat endpoint Transfer Family dengan AS2 diaktifkan, dan konektor Transfer Family AS2 Anda dapat melihat detail untuk lokakarya ini di sini.
- Buat solusi yang memberikan panduan preskriptif dan lab langsung tentang bagaimana Anda dapat membangun arsitektur transfer file yang terukur dan aman AWS tanpa perlu memodifikasi aplikasi yang ada atau mengelola infrastruktur server. Anda dapat melihat detail untuk lokakarya ini di sini.

Posting blog yang relevan untuk Transfer Family

Tabel berikut mencantumkan posting blog yang berisi informasi berguna bagi pelanggan Transfer Family. Tabel diurutkan dalam urutan kronologis terbalik, sehingga posting terbaru ada di awal tabel.

Judul dan tautan posting blog	Tanggal
Bagaimana Transfer Family dapat membantu Anda membangun solusi transfer file terkelola yang aman dan sesuai	Januari 3, 2024
Mendeteksi ancaman malware menggunakan AWS Transfer Family	Juli 20, 2023
Memperluas beban kerja SAP dengan AWS Transfer Family	13 Juli 2023
Enkripsi dan dekripsi file dengan PGP dan AWS Transfer Family	Juni 21, 2023
Mengautentikasi AWS Transfer Family dengan Azure Active Directory dan AWS Lambda	Desember 15, 2022
Sesuaikan pemberitahuan pengiriman file menggunakan alur kerja AWS Transfer Family terkelola	14 Oktober 2022
Membangun platform transfer file cloud-native menggunakan alur kerja AWS Transfer Family	5 Januari 2022
Mengaktifkan manajemen kunci swalayan pengguna dengan A AWS Transfer Family dan. AWS Lambda	Desember 17, 2021
Tingkatkan kontrol akses data dengan AWS Transfer Family dan Amazon S3	5 Oktober 2021
Meningkatkan throughput untuk penggunaa n AWS Global Accelerator dan AWS Transfer	7 Juni 2021

Judul dan tautan posting blog	Tanggal
Family layanan transfer file yang dihadapi internet	
Mengamankan AWS Transfer Family dengan Firewall Aplikasi AWS Web dan Amazon API Gateway	5 Mei 2021
Mengamankan AWS Transfer Family dengan Firewall Aplikasi AWS Web dan Amazon API Gateway	15 Januari 2021
AWS Transfer Family dukungan untuk Amazon Elastic File System	7 Januari 2021
Aktifkan otentikasi kata sandi untuk menggunakan AWS Transfer FamilyAWS Secrets Manager	5 November 2020
Memusatkan akses data menggunakan AWS Transfer Family dan AWS Storage Gateway	22 Juni 2020
Menggunakan Amazon EFS untuk AWS Lambda aplikasi tanpa server Anda	18 Juni 2020
Gunakan daftar izin IP untuk mengamankan AWS Transfer Family server Anda	8 April 2020
Minimalkan latensi jaringan dengan AWS transfer Anda untuk server SFTP	19 Februari 2020
Angkat dan Pergeseran migrasi server SFTP ke AWS	12 Februari 2020
Sederhanakan Struktur AWS SFTP Anda dengan direktori chroot dan logis	26 September 2019
Menggunakan Okta sebagai penyedia identitas dengan AWS Transfer Family	30 Mei 2019

Prasyarat

Bagian berikut menjelaskan prasyarat yang diperlukan untuk menggunakan layanan ini. AWS Transfer Family Minimal, Anda perlu membuat bucket Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3) dan menyediakan akses ke bucket tersebut melalui AWS Identity and Access Management peran (IAM). Peran Anda juga perlu membangun hubungan kepercayaan. Hubungan kepercayaan ini memungkinkan Transfer Family untuk mengambil peran IAM untuk mengakses bucket Anda sehingga dapat melayani permintaan transfer file pengguna Anda.

Topik

- AWS Wilayah, titik akhir, dan kuota yang didukung
- Mendaftar untuk AWS
- Konfigurasikan penyimpanan untuk digunakan dengan AWS Transfer Family
- Buat peran dan kebijakan IAM

AWS Wilayah, titik akhir, dan kuota yang didukung

Untuk terhubung secara terprogram ke AWS layanan, Anda menggunakan titik akhir. Misalnya, titik akhir untuk pelanggan di wilayah AS Timur (Ohio) (`us-east-2`), adalah. `transfer.us-east-2.amazonaws.com` Kuota layanan, juga disebut sebagai batas, adalah jumlah maksimum sumber daya layanan atau operasi untuk Anda Akun AWS. Dalam panduan ini, Anda dapat menemukan kuota di Kuota AS2 danKuota untuk konektor SFTP.

Untuk informasi selengkapnya tentang AWS Wilayah, titik akhir, dan kuota layanan yang didukung, lihat AWS Transfer Family titik akhir dan kuota di. Referensi Umum Amazon Web

Mendaftar untuk AWS

Saat Anda mendaftar ke Amazon Web Services (AWS), AWS akun Anda secara otomatis mendaftar untuk semua layanan AWS, termasuk AWS Transfer Family. Anda hanya membayar biaya layanan yang Anda gunakan.

Jika Anda sudah memiliki AWS akun, lompat ke tugas berikutnya. Jika Anda belum memiliki akun AWS , gunakan prosedur berikut untuk membuatnya.

Jika Anda tidak memiliki Akun AWS, selesaikan langkah-langkah berikut untuk membuatnya.

Untuk mendaftar untuk Akun AWS

1. Buka https://portal.aws.amazon.com/billing/signup.
2. Ikuti petunjuk secara online.

   Anda akan diminta untuk menerima panggilan telepon dan memasukkan kode verifikasi pada keypad telepon sebagai bagian dari prosedur pendaftaran.

   Saat Anda mendaftar untuk sebuah Akun AWS, sebuah Pengguna root akun AWSdibuat. Pengguna root memiliki akses ke semua Layanan AWS dan sumber daya dalam akun. Sebagai praktik terbaik keamanan, tetapkan akses administratif ke pengguna administratif, dan hanya gunakan pengguna root untuk melakukan tugas yang memerlukan akses pengguna root.

Untuk informasi tentang harga dan penggunaan AWS Pricing Calculator untuk mendapatkan perkiraan biaya penggunaan Transfer Family, lihat AWS Transfer Family harga.

Untuk informasi tentang ketersediaan AWS Wilayah, lihat AWS Transfer Family titik akhir dan kuota di. Referensi Umum AWS

Konfigurasikan penyimpanan untuk digunakan dengan AWS Transfer Family

Topik ini menjelaskan opsi penyimpanan yang dapat Anda gunakan AWS Transfer Family. Anda dapat menggunakan Amazon S3 atau Amazon EFS sebagai penyimpanan untuk server Transfer Family Anda.

Daftar Isi

- Konfigurasikan bucket Amazon S3
  - Titik akses Amazon S3
  - Perilaku Amazon S3 HeadObject
    - Memberikan kemampuan untuk hanya menulis dan daftar file
    - Sejumlah besar objek nol-byte menyebabkan masalah latensi
- Konfigurasikan sistem file Amazon EFS
  - Kepemilikan file Amazon EFS
  - Menyiapkan pengguna Amazon EFS untuk Transfer Family

- Konfigurasikan pengguna Transfer Family di Amazon EFS
- Buat pengguna root Amazon EFS

- Perintah Amazon EFS yang didukung

Konfigurasikan bucket Amazon S3

AWS Transfer Family mengakses bucket Amazon S3 untuk melayani permintaan transfer pengguna, jadi Anda perlu menyediakan bucket Amazon S3 sebagai bagian dari pengaturan server yang mendukung protokol transfer file Anda. Anda dapat menggunakan bucket yang sudah ada, atau Anda dapat membuat yang baru.

> **Note**
>
> Anda tidak harus menggunakan server dan bucket Amazon S3 yang berada di AWS Wilayah yang sama, tetapi kami merekomendasikan ini sebagai praktik terbaik.

Saat Anda mengatur pengguna, Anda menetapkan masing-masing peran IAM kepada mereka. Peran ini menentukan tingkat akses yang mereka miliki ke bucket Amazon S3 Anda.

Untuk informasi tentang cara membuat bucket baru, lihat Bagaimana cara membuat bucket S3? di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

> **Note**
>
> Anda dapat menggunakan Amazon S3 Object Lock untuk mencegah objek ditimpa untuk jangka waktu yang tetap atau tanpa batas waktu. Ini bekerja dengan cara yang sama dengan Transfer Family seperti layanan lainnya. Jika suatu objek ada dan dilindungi, menulis ke file itu atau menghapusnya tidak diperbolehkan. Untuk detail selengkapnya tentang Kunci Objek Amazon S3, lihat Menggunakan Kunci Objek Amazon S3 di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Titik akses Amazon S3

AWS Transfer Family mendukung Poin Akses Amazon S3, fitur Amazon S3 yang memungkinkan Anda mengelola akses granular ke kumpulan data bersama dengan mudah. Anda dapat

menggunakan alias S3 Access Point di mana pun Anda menggunakan nama bucket S3. Anda dapat membuat ratusan titik akses di Amazon S3 untuk pengguna yang memiliki izin berbeda untuk mengakses data bersama di bucket Amazon S3.

Misalnya, Anda dapat menggunakan titik akses untuk memungkinkan tiga tim berbeda memiliki akses ke kumpulan data bersama yang sama di mana satu tim dapat membaca data dari S3, tim kedua dapat menulis data ke S3, dan tim ketiga dapat membaca, menulis, dan menghapus data dari S3. Untuk menerapkan kontrol akses granular seperti yang disebutkan di atas, Anda dapat membuat titik akses S3 yang berisi kebijakan yang memberikan akses asimetris ke tim yang berbeda. Anda dapat menggunakan titik akses S3 dengan server Transfer Family Anda untuk mencapai kontrol akses yang halus, tanpa membuat kebijakan bucket S3 kompleks yang mencakup ratusan kasus penggunaan. Untuk mempelajari lebih lanjut tentang cara menggunakan titik akses S3 dengan server Transfer Family, lihat Tingkatkan kontrol akses data dengan AWS Transfer Family dan posting blog Amazon S3.

> **Note**
>
> AWS Transfer Family saat ini tidak mendukung Titik Akses Multi-Wilayah Amazon S3.

Perilaku Amazon S3 HeadObject

> **Note**
>
> Saat membuat atau memperbarui server Transfer Family, Anda dapat mengoptimalkan kinerja untuk direktori Amazon S3, yang menghilangkan panggilan. `HeadObject`

Di Amazon S3, bucket dan objek adalah sumber daya utama, dan objek disimpan dalam bucket. Amazon S3 dapat meniru sistem file hierarkis, tetapi terkadang dapat berperilaku berbeda dari sistem file biasa. Misalnya, direktori bukan konsep kelas satu di Amazon S3 tetapi didasarkan pada kunci objek. AWS Transfer Family menyimpulkan jalur direktori dengan memisahkan kunci objek dengan karakter garis miring maju (/), memperlakukan elemen terakhir sebagai nama file, lalu mengelompokkan nama file yang memiliki awalan yang sama bersama-sama di bawah jalur yang sama. Objek nol-byte dibuat untuk mewakili jalur folder saat Anda membuat direktori kosong menggunakan `mkdir` atau dengan menggunakan konsol Amazon S3. Kunci untuk benda-benda ini berakhir dengan garis miring ke depan. Objek nol-byte ini dijelaskan dalam Mengatur objek di konsol Amazon S3 menggunakan folder di Panduan Pengguna Amazon S3.

Saat Anda menjalankan `ls` perintah, dan beberapa hasilnya adalah objek zero-byte Amazon S3 (objek ini memiliki kunci yang diakhiri dengan karakter garis miring ke depan), Transfer Family mengeluarkan `HeadObject` permintaan untuk masing-masing objek ini (lihat di Referensi API Layanan Penyimpanan Sederhana Amazon untuk HeadObjectdetailnya). Hal ini dapat mengakibatkan masalah berikut saat menggunakan Amazon S3 sebagai penyimpanan Anda dengan Transfer Family.

Memberikan kemampuan untuk hanya menulis dan daftar file

Dalam beberapa kasus, Anda mungkin hanya ingin menawarkan akses tulis ke objek Amazon S3 Anda. Misalnya, Anda mungkin ingin menyediakan akses untuk menulis (atau mengunggah) dan mencantumkan objek dalam ember, tetapi tidak untuk membaca (mengunduh) objek. Untuk melakukan `ls` dan `mkdir` memerintahkan dengan menggunakan klien transfer file, Anda harus memiliki Amazon S3 `ListObjects` dan `PutObject` izin. Namun, ketika Transfer Family perlu melakukan `HeadObject` panggilan untuk menulis atau membuat daftar file, panggilan gagal dengan kesalahan Akses ditolak, karena panggilan ini memerlukan `GetObject` izin.

> **Note**
>
> Saat membuat atau memperbarui server Transfer Family, Anda dapat mengoptimalkan kinerja untuk direktori Amazon S3, yang menghilangkan panggilan. `HeadObject`

Dalam hal ini, Anda dapat memberikan akses dengan menambahkan kondisi kebijakan AWS Identity and Access Management (IAM) yang menambahkan `GetObject` izin hanya untuk objek yang diakhiri dengan garis miring ()`/`. Kondisi ini mencegah `GetObject` panggilan pada file (sehingga tidak dapat dibaca), tetapi memungkinkan pengguna untuk membuat daftar dan melintasi folder. Kebijakan contoh berikut hanya menawarkan akses tulis dan daftar ke bucket Amazon S3 Anda. Untuk menggunakan kebijakan ini, ganti *`DOC-EXAMPLE-BUCKET`* dengan nama bucket Anda.

```
{
    "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**
>
> Kebijakan ini tidak mengizinkan pengguna untuk menambahkan file. Dengan kata lain, pengguna yang diberi kebijakan ini tidak dapat membuka file untuk menambahkan konten ke dalamnya, atau memodifikasinya. Selain itu, jika kasus penggunaan Anda memerlukan `HeadObject` panggilan sebelum mengunggah file, kebijakan ini tidak akan berfungsi untuk Anda.

Sejumlah besar objek nol-byte menyebabkan masalah latensi

Jika bucket Amazon S3 Anda berisi sejumlah besar objek nol-byte ini, Transfer Family mengeluarkan banyak panggilan, yang dapat mengakibatkan `HeadObject` penundaan pemrosesan.

Salah satu solusi yang mungkin untuk masalah ini adalah menghapus semua objek zero-byte Anda. Perhatikan hal berikut:

- Direktori kosong tidak akan ada lagi. Direktori hanya ada karena nama mereka berada di kunci suatu objek.
- Tidak mencegah seseorang menelepon `mkdir` dan merusak semuanya lagi. Anda dapat mengurangi ini dengan membuat kebijakan yang mencegah pembuatan direktori.
- Beberapa skenario menggunakan objek 0-byte ini. Misalnya, Anda memiliki struktur seperti / inboxes/customer1000 dan direktori kotak masuk dibersihkan setiap hari.

Solusi lain yang mungkin adalah membatasi jumlah objek yang terlihat melalui kondisi kebijakan untuk mengurangi jumlah `HeadObject` panggilan. Agar ini menjadi solusi yang bisa diterapkan, Anda harus menerima bahwa Anda mungkin hanya dapat melihat sekumpulan terbatas semua sub-direktori Anda.

Konfigurasikan sistem file Amazon EFS

AWS Transfer Family mengakses Amazon Elastic File System (Amazon EFS) untuk melayani permintaan transfer pengguna Anda. Jadi, Anda harus menyediakan sistem file Amazon EFS sebagai bagian dari pengaturan server yang mendukung protokol transfer file Anda. Anda dapat menggunakan sistem file yang ada, atau Anda dapat membuat yang baru.

Perhatikan hal berikut:

- Saat Anda menggunakan server Transfer Family dan sistem file Amazon EFS, server dan sistem file harus sama Wilayah AWS.
- Server dan sistem file tidak perlu berada di akun yang sama. Jika server dan sistem file tidak berada dalam akun yang sama, kebijakan sistem file harus memberikan izin eksplisit untuk peran pengguna.

  Untuk informasi tentang cara menyiapkan beberapa akun, lihat Mengelola AWS akun di organisasi Anda di Panduan AWS Organizations Pengguna.
- Saat Anda mengatur pengguna, Anda menetapkan masing-masing peran IAM kepada mereka. Peran ini menentukan tingkat akses yang mereka miliki ke sistem file Amazon EFS Anda.
- Untuk detail tentang pemasangan sistem file Amazon EFS, lihat Memasang sistem file Amazon EFS.

Untuk detail selengkapnya tentang cara AWS Transfer Family dan Amazon EFS bekerja sama, lihat Menggunakan AWS Transfer Family untuk mengakses file di sistem file Amazon EFS Anda di Panduan Pengguna Amazon Elastic File System.

Kepemilikan file Amazon EFS

Amazon EFS menggunakan model izin file Portable Operating System Interface (POSIX) untuk mewakili kepemilikan file.

Di POSIX, pengguna dalam sistem dikategorikan ke dalam tiga kelas izin yang berbeda: Ketika Anda mengizinkan pengguna untuk mengakses file yang disimpan dalam sistem file Amazon EFS menggunakan AWS Transfer Family, Anda harus menetapkan mereka “profil POSIX.” Profil ini digunakan untuk menentukan akses mereka ke file dan direktori di sistem file Amazon EFS.

- User (u): Pemilik file atau direktori. Biasanya, pencipta file atau direktori juga pemiliknya.
- Grup (g): Kumpulan pengguna yang membutuhkan akses identik ke file dan direktori yang mereka bagikan.
- Lainnya (o): Semua pengguna lain yang memiliki akses ke sistem kecuali pemilik dan anggota grup. Kelas izin ini juga disebut sebagai “Publik.”

Dalam model izin POSIX, setiap objek sistem file (file, direktori, tautan simbolis, pipa bernama, dan soket) dikaitkan dengan tiga set izin yang disebutkan sebelumnya. Objek Amazon EFS memiliki mode gaya Unix yang terkait dengannya. Nilai mode ini mendefinisikan izin untuk melakukan tindakan pada objek tersebut.

Selain itu, pada sistem bergaya Unix, pengguna dan grup dipetakan ke pengidentifikasi numerik, yang digunakan Amazon EFS untuk mewakili kepemilikan file. Untuk Amazon EFS, objek dimiliki oleh satu pemilik dan satu grup. Amazon EFS menggunakan ID numerik yang dipetakan untuk memeriksa izin saat pengguna mencoba mengakses objek sistem file.

Menyiapkan pengguna Amazon EFS untuk Transfer Family

Sebelum mengatur pengguna Amazon EFS, Anda dapat melakukan salah satu hal berikut:

- Anda dapat membuat pengguna dan mengatur folder rumah mereka di Amazon EFS. Lihat Konfigurasikan pengguna Transfer Family di Amazon EFS untuk detail.
- Jika Anda merasa nyaman menambahkan pengguna root, Anda bisaBuat pengguna root Amazon EFS.

> **Note**
>
> Server Transfer Family tidak mendukung jalur akses Amazon EFS untuk menetapkan izin POSIX. Profil POSIX pengguna Transfer Family (dijelaskan di bagian sebelumnya) menawarkan kemampuan untuk mengatur izin POSIX. Izin ini ditetapkan pada tingkat pengguna, untuk akses granular, berdasarkan UID, GID, dan GID sekunder.

Konfigurasikan pengguna Transfer Family di Amazon EFS

Transfer Family memetakan pengguna ke UID/GID dan direktori yang Anda tentukan. Jika UID/GID/direktori belum ada di EFS, maka Anda harus membuatnya sebelum menetapkannya di Transfer ke pengguna. Detail untuk membuat pengguna Amazon EFS dijelaskan dalam Bekerja dengan pengguna, grup, dan izin di Tingkat Sistem File Jaringan (NFS) di Panduan Pengguna Amazon Elastic File System.

Langkah-langkah untuk mengatur pengguna Amazon EFS di Transfer Family

1. Petakan EFS UID dan GID untuk pengguna Anda di Transfer Family menggunakan `PosixProfile`bidang.
2. Jika Anda ingin pengguna memulai di folder tertentu saat login, Anda dapat menentukan direktori EFS di bawah `HomeDirectory`bidang.

Anda dapat mengotomatiskan proses, dengan menggunakan CloudWatch aturan dan fungsi Lambda. Misalnya fungsi Lambda yang berinteraksi dengan EFS, lihat Menggunakan Amazon EFS untuk AWS Lambda aplikasi tanpa server Anda.

Selain itu, Anda dapat mengonfigurasi direktori logis untuk pengguna Transfer Family Anda. Untuk detailnya, lihat Konfigurasikan direktori logis untuk Amazon EFS bagian dalam Menggunakan direktori logis untuk menyederhanakan struktur direktori Transfer Family Anda topik.

Buat pengguna root Amazon EFS

Jika organisasi Anda merasa nyaman bagi Anda untuk mengaktifkan akses pengguna root melalui SFTP/FTPS untuk konfigurasi pengguna Anda, Anda dapat membuat pengguna yang UID dan GID adalah 0 (pengguna root), kemudian gunakan pengguna root itu untuk membuat folder dan menetapkan pemilik ID POSIX untuk pengguna lainnya. Keuntungan dari opsi ini adalah tidak perlu memasang sistem file Amazon EFS.

Lakukan langkah-langkah yang dijelaskan dalamMenambahkan pengguna yang dikelola layanan Amazon EFS, dan untuk ID Pengguna dan ID Grup, masukkan 0 (nol).

Perintah Amazon EFS yang didukung

Perintah berikut didukung untuk Amazon EFS untuk AWS Transfer Family.

- `cd`
- `ls/dir`
- `pwd`
- `put`
- `get`
- `rename`
- `chown`: Hanya root (yaitu, pengguna dengan uid=0) yang dapat mengubah kepemilikan dan izin file dan direktori.
- `chmod`: Hanya root yang dapat mengubah kepemilikan dan izin file dan direktori.
- `chgrp`: Didukung baik untuk root atau untuk pemilik file yang hanya dapat mengubah grup file menjadi salah satu grup sekunder mereka.
- `ln -s/symlink`
- `mkdir`
- `rm/delete`
- `rmdir`
- `chmtime`

Buat peran dan kebijakan IAM

Topik ini menjelaskan jenis kebijakan dan peran yang dapat digunakan AWS Transfer Family, dan berjalan melalui proses pembuatan peran pengguna. Ini juga menjelaskan cara kerja kebijakan sesi dan memberikan contoh peran pengguna.

AWS Transfer Family menggunakan jenis peran berikut:

- Peran pengguna — Memungkinkan pengguna yang dikelola layanan mengakses sumber daya Transfer Family yang diperlukan. AWS Transfer Family mengasumsikan peran ini dalam konteks ARN pengguna Transfer Family.

- Peran akses - Menyediakan akses hanya ke file Amazon S3 yang sedang ditransfer. Untuk transfer AS2 masuk, peran akses menggunakan Amazon Resource Name (ARN) untuk perjanjian. Untuk transfer AS2 keluar, peran akses menggunakan ARN untuk konektor.
- Peran pemanggilan — Untuk digunakan dengan Amazon API Gateway sebagai penyedia identitas kustom server. Transfer Family mengasumsikan peran ini dalam konteks ARN server Transfer Family.
- Peran logging - Digunakan untuk log entri ke Amazon CloudWatch. Transfer Family menggunakan peran ini untuk mencatat detail keberhasilan dan kegagalan bersama dengan informasi tentang transfer file. Transfer Family mengasumsikan peran ini dalam konteks ARN server Transfer Family. Untuk transfer AS2 keluar, peran logging menggunakan konektor ARN.
- Peran eksekusi — Memungkinkan pengguna Transfer Family memanggil dan meluncurkan alur kerja. Transfer Family mengasumsikan peran ini dalam konteks alur kerja Transfer Family ARN.

Selain peran ini, Anda juga dapat menggunakan kebijakan sesi. Kebijakan sesi digunakan untuk membatasi akses bila diperlukan. Perhatikan bahwa kebijakan ini berdiri sendiri: artinya, Anda tidak menambahkan kebijakan ini ke peran. Sebaliknya, Anda menambahkan kebijakan sesi langsung ke pengguna Transfer Family.

> **Note**
>
> Saat membuat pengguna Transfer Family yang dikelola layanan, Anda dapat memilih Kebijakan buat otomatis berdasarkan folder beranda. Ini adalah pintasan yang berguna jika Anda ingin membatasi akses pengguna ke folder mereka sendiri. Selain itu, Anda dapat melihat detail tentang kebijakan sesi dan contoh diBagaimana kebijakan sesi bekerja. Anda juga dapat menemukan informasi selengkapnya tentang kebijakan sesi dalam kebijakan Sesi di Panduan Pengguna IAM.

Topik

- Membuat peran pengguna
- Bagaimana kebijakan sesi bekerja
- Contoh kebijakan akses baca/tulis

Membuat peran pengguna

Saat Anda membuat pengguna, Anda membuat sejumlah keputusan tentang akses pengguna. Keputusan ini mencakup bucket Amazon S3 atau sistem file Amazon EFS mana yang dapat diakses pengguna, bagian mana dari setiap bucket Amazon S3 dan file mana dalam sistem file yang dapat diakses, dan izin apa yang dimiliki pengguna (misalnya, atau). `PUT` `GET`

Untuk menetapkan akses, Anda membuat kebijakan dan peran berbasis identitas AWS Identity and Access Management (IAM) yang menyediakan informasi akses tersebut. Sebagai bagian dari proses ini, Anda menyediakan akses bagi pengguna Anda ke bucket Amazon S3 atau sistem file Amazon EFS yang merupakan target atau sumber untuk operasi file. Untuk melakukan ini, ambil langkah-langkah tingkat tinggi berikut, dijelaskan secara rinci nanti:

Membuat peran pengguna

1. Buat kebijakan IAM untuk AWS Transfer Family. Ini dijelaskan dalamUntuk membuat kebijakan IAM untuk AWS Transfer Family.
2. Buat peran IAM dan lampirkan kebijakan IAM baru. Sebagai contoh, lihat Contoh kebijakan akses baca/tulis.
3. Membangun hubungan kepercayaan antara AWS Transfer Family dan peran IAM. Ini dijelaskan dalamUntuk membangun hubungan kepercayaan.

Prosedur berikut menjelaskan cara membuat kebijakan dan peran IAM.

Untuk membuat kebijakan IAM untuk AWS Transfer Family

1. Buka konsol IAM di https://console.aws.amazon.com/iam/.
2. Di panel navigasi, pilih Kebijakan dan kemudian pilih Buat kebijakan.
3. Pada halaman Buat Kebijakan, pilih tab JSON.
4. Di editor yang muncul, ganti isi editor dengan kebijakan IAM yang ingin Anda lampirkan ke peran IAM.

   Anda dapat memberikan akses baca/tulis atau membatasi pengguna ke direktori home mereka. Untuk informasi selengkapnya, lihat Contoh kebijakan akses baca/tulis.
5. Pilih Kebijakan tinjauan dan berikan nama dan deskripsi untuk kebijakan Anda, lalu pilih Buat kebijakan.

Selanjutnya, Anda membuat peran IAM dan melampirkan kebijakan IAM baru ke dalamnya.

Untuk membuat peran IAM untuk AWS Transfer Family

1. Di panel navigasi, pilih Peran, lalu pilih Buat peran.

   Pada halaman Buat peran, pastikan bahwa AWS layanan dipilih.
2. Pilih Transfer dari daftar layanan, lalu pilih Berikutnya: Izin. Ini membangun hubungan kepercayaan antara AWS Transfer Family dan AWS.
3. Di bagian Lampirkan kebijakan izin, cari dan pilih kebijakan yang baru saja Anda buat, lalu pilih Berikutnya: Tag.
4. (Opsional) Masukkan kunci dan nilai untuk tag, dan pilih Berikutnya: Tinjau.
5. Pada halaman Tinjauan, masukkan nama dan deskripsi untuk peran baru Anda, lalu pilih Buat peran.

Selanjutnya, Anda membangun hubungan kepercayaan antara AWS Transfer Family dan AWS.

Untuk membangun hubungan kepercayaan

> **Note**
>
> Dalam contoh kami, kami menggunakan keduanya `ArnLike` dan`ArnEquals`. Mereka identik secara fungsional, dan oleh karena itu Anda dapat menggunakan keduanya ketika Anda membuat kebijakan Anda. Dokumentasi Transfer Family digunakan `ArnLike` ketika kondisi berisi karakter wildcard, dan `ArnEquals` untuk menunjukkan kondisi kecocokan yang tepat.

1. Di konsol IAM, pilih peran yang baru saja Anda buat.
2. Pada halaman Ringkasan, pilih Trust relationship, lalu pilih Edit trust relationship.
3. Di editor Edit Trust Relationship, pastikan layanannya`"transfer.amazonaws.com"`. Kebijakan akses ditampilkan sebagai berikut.

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

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

Kami menyarankan Anda menggunakan kunci `aws:SourceAccount` dan `aws:SourceArn` kondisi untuk melindungi diri Anda dari masalah wakil yang membingungkan. Akun sumber adalah pemilik server dan sumber ARN adalah ARN pengguna. Sebagai contoh:

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

Anda juga dapat menggunakan `ArnLike` kondisi ini jika Anda ingin membatasi ke server tertentu, bukan server apa pun di akun pengguna. Sebagai contoh:

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

> **Note**
>
> Dalam contoh di atas, ganti setiap *placeholder input pengguna dengan informasi* Anda sendiri.

Untuk detail tentang masalah wakil yang membingungkan dan lebih banyak contoh, lihatPencegahan confused deputy lintas layanan.

4. Pilih Perbarui Kebijakan Kepercayaan untuk memperbarui kebijakan akses.

Anda sekarang telah membuat peran IAM yang memungkinkan AWS Transfer Family untuk memanggil AWS layanan atas nama Anda. Anda melampirkan peran kebijakan IAM yang Anda buat untuk memberikan akses ke pengguna Anda. Di Memulai dengan AWS Transfer Family titik akhir server bagian ini, peran dan kebijakan ini ditetapkan untuk pengguna atau pengguna Anda.

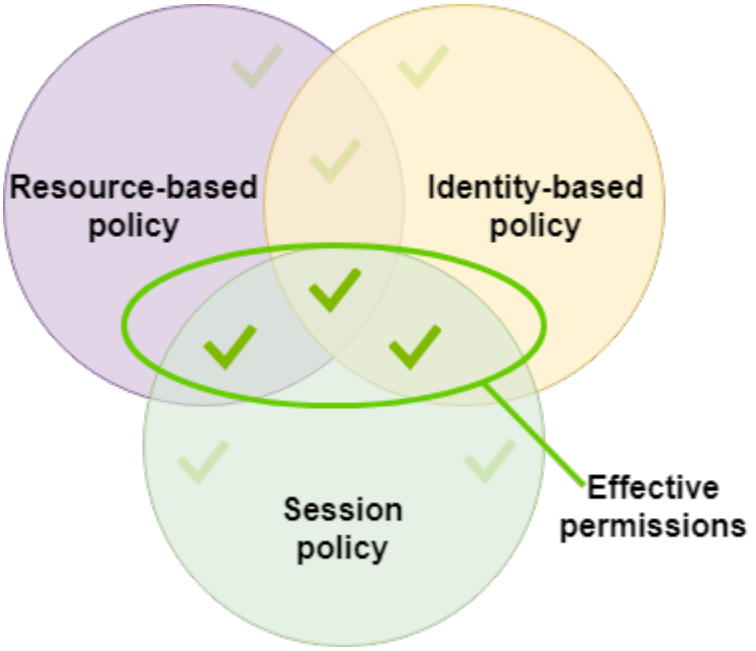
Lihat juga

- Untuk informasi umum selengkapnya tentang peran IAM, lihat Membuat peran untuk mendelegasikan izin ke AWS layanan di Panduan Pengguna IAM.
- Untuk mempelajari selengkapnya tentang kebijakan berbasis identitas untuk sumber daya Amazon S3, lihat Manajemen identitas dan akses di Amazon S3 di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.
- Untuk mempelajari lebih lanjut tentang kebijakan berbasis identitas untuk sumber daya Amazon EFS, lihat Menggunakan IAM untuk mengontrol akses data sistem file di Panduan Pengguna Amazon Elastic File System.

Bagaimana kebijakan sesi bekerja

Saat administrator membuat peran, peran tersebut sering menyertakan izin luas untuk mencakup beberapa kasus penggunaan atau anggota tim. Jika administrator mengonfigurasi URL konsol, administrator dapat mengurangi izin untuk sesi yang dihasilkan dengan menggunakan kebijakan sesi. Misalnya, jika Anda membuat peran dengan akses baca/tulis, Anda dapat mengatur URL yang membatasi akses pengguna hanya ke direktori beranda mereka.

Kebijakan sesi adalah kebijakan lanjutan yang Anda berikan sebagai parameter saat Anda membuat sesi sementara untuk peran atau pengguna secara terprogram. Kebijakan sesi berguna untuk mengunci pengguna sehingga mereka hanya memiliki akses ke bagian bucket Anda di mana awalan objek berisi nama pengguna mereka. Diagram berikut menunjukkan bahwa izin kebijakan sesi adalah persimpangan kebijakan sesi dan kebijakan berbasis sumber daya ditambah persimpangan kebijakan sesi dan kebijakan berbasis identitas.



Untuk detail selengkapnya, lihat Kebijakan sesi di Panduan Pengguna IAM.

Di AWS Transfer Family, kebijakan sesi hanya didukung saat Anda mentransfer ke atau dari Amazon S3. Contoh kebijakan berikut adalah kebijakan sesi yang membatasi akses pengguna ke home direktori mereka saja. Perhatikan hal berikut:

- `PutObjectACL`Pernyataan `GetObjectACL` dan pernyataan hanya diperlukan jika Anda perlu mengaktifkan Akses Lintas Akun. Artinya, server Transfer Family Anda perlu mengakses bucket di akun yang berbeda.
- Panjang maksimum kebijakan sesi adalah 2048 karakter. Untuk detail selengkapnya, lihat parameter Permintaan kebijakan untuk `CreateUser` tindakan dalam referensi API.
- Jika bucket Amazon S3 dienkripsi menggunakan AWS Key Management Service (AWS KMS), Anda harus menentukan izin tambahan dalam kebijakan Anda. Untuk detailnya, lihat Enkripsi data di Amazon S3.

```
{
    "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**
>
> Contoh kebijakan sebelumnya mengasumsikan bahwa pengguna memiliki direktori home mereka diatur untuk menyertakan garis miring, untuk menandakan bahwa itu adalah direktori. Jika, di sisi lain, Anda menetapkan pengguna `HomeDirectory` tanpa garis miring, maka Anda harus memasukkannya sebagai bagian dari kebijakan Anda.

Dalam contoh kebijakan sebelumnya, perhatikan penggunaan parameter`transfer:HomeFolder`,`transfer:HomeBucket`, dan `transfer:HomeDirectory` kebijakan. Parameter ini diatur untuk `HomeDirectory` yang dikonfigurasi untuk pengguna, seperti

yang dijelaskan dalam HomeDirectorydanMenerapkan metode API Gateway. Parameter ini memiliki definisi berikut:

- `transfer:HomeBucket`Parameter diganti dengan komponen pertama dari`HomeDirectory`.
- `transfer:HomeFolder`Parameter diganti dengan bagian `HomeDirectory` parameter yang tersisa.
- `transfer:HomeDirectory`Parameter memiliki garis miring depan (`/`) yang dihapus sehingga dapat digunakan sebagai bagian dari Nama Sumber Daya Amazon S3 (ARN) dalam sebuah pernyataan. `Resource`

> **Note**
>
> Jika Anda menggunakan direktori logik—yaitu, pengguna adalah `LOGICAL` —parameter kebijakan ini (`HomeBucket`,`HomeDirectory`, dan`HomeFolder`) tidak didukung. `homeDirectoryType`

Misalnya, asumsikan bahwa `HomeDirectory` parameter yang dikonfigurasi untuk pengguna Transfer Family adalah`/home/bob/amazon/stuff/`.

- `transfer:HomeBucket`diatur ke`/home`.
- `transfer:HomeFolder`diatur ke`/bob/amazon/stuff/`.
- `transfer:HomeDirectory`menjadi`home/bob/amazon/stuff/`.

Yang pertama `"Sid"` memungkinkan pengguna untuk membuat daftar semua direktori mulai dari`/home/bob/amazon/stuff/`.

Yang kedua `"Sid"` membatasi pengguna `put` dan `get` akses ke jalur yang sama,`/home/bob/amazon/stuff/`.

Contoh kebijakan akses baca/tulis

Berikan akses baca/tulis ke bucket Amazon S3

Contoh kebijakan berikut untuk AWS Transfer Family memberikan akses baca/tulis ke objek di bucket Amazon S3 Anda.

Perhatikan hal berikut:

- Ganti *DOC-CONTOH-BUCKET* dengan nama bucket Amazon S3 Anda.
- `PutObjectACL`Pernyataan `GetObjectACL` dan pernyataan hanya diperlukan jika Anda perlu mengaktifkan Akses Lintas Akun. Artinya, server Transfer Family Anda perlu mengakses bucket di akun yang berbeda.
- `DeleteObjectVersion`Pernyataan `GetObjectVersion` dan hanya diperlukan jika pembuatan versi diaktifkan di bucket Amazon S3 yang sedang diakses.

> **Note**
>
> Jika Anda pernah mengaktifkan pembuatan versi untuk bucket, Anda memerlukan izin ini, karena Anda hanya dapat menangguhkan pembuatan versi di Amazon S3, dan tidak mematikannya sepenuhnya. Untuk detailnya, lihat Bucket yang tidak berversi, berkemampuan versi, dan ditangguhkan versi.

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

Berikan akses sistem file ke file dalam sistem file Amazon EFS

> **Note**
>
> Selain kebijakan, Anda juga harus memastikan izin file POSIX Anda memberikan akses yang sesuai. Untuk informasi selengkapnya, lihat Bekerja dengan pengguna, grup, dan izin di Tingkat Network File System (NFS) dalam Panduan Pengguna Amazon Elastic File System.

Contoh kebijakan berikut memberikan akses sistem file root ke file di sistem file Amazon EFS Anda.

> **Note**
>
> Dalam contoh berikut, ganti *wilayah dengan wilayah* Anda, *account-id* dengan akun tempat file berada, dan *file-system-id*dengan ID Amazon Elastic File System (Amazon EFS) Anda.

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

Contoh kebijakan berikut memberikan akses sistem file pengguna ke file di sistem file Amazon EFS Anda.

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

Tutorial Transfer Family

Panduan AWS Transfer Family pengguna menyediakan penelusuran terperinci untuk beberapa kasus penggunaan.

- Memulai dengan AWS Transfer Family titik akhir server: tutorial ini memandu Anda melalui pembuatan server SFTP Transfer Family dan pengguna yang dikelola layanan, kemudian menunjukkan cara mentransfer file menggunakan klien.
- Menyiapkan dan menggunakan konektor SFTP: tutorial ini menggambarkan cara mengatur konektor SFTP, dan kemudian mentransfer file antara penyimpanan Amazon S3 dan server SFTP.
- Menyiapkan metode Amazon API Gateway sebagai penyedia identitas kustom : tutorial ini menggambarkan cara mengatur metode Amazon API Gateway dan menggunakannya sebagai penyedia identitas khusus untuk mengunggah file ke AWS Transfer Family server.
- Menyiapkan alur kerja terkelola untuk mendekripsi file: tutorial ini menggambarkan cara mengatur alur kerja terkelola yang berisi langkah dekripsi, dan cara mengunggah file terenkripsi ke bucket Amazon S3 dan kemudian melihat file yang didekripsi.
- Menyiapkan konfigurasi AS2: tutorial ini berjalan melalui langkah-langkah yang diperlukan untuk mengkonfigurasi server AS2 Transfer Family. Ada instruksi untuk mengimpor sertifikat, membuat profil dan perjanjian, secara opsional membuat konektor AS2, dan kemudian menguji konfigurasi.

Topik

- Memulai dengan AWS Transfer Family titik akhir server
- Menyiapkan alur kerja terkelola untuk mendekripsi file
- Menyiapkan dan menggunakan konektor SFTP
- Menyiapkan metode Amazon API Gateway sebagai penyedia identitas kustom
- Menyiapkan konfigurasi AS2

Memulai dengan AWS Transfer Family titik akhir server

Gunakan tutorial ini untuk memulai AWS Transfer Family (Transfer Family). Anda akan mempelajari cara membuat server berkemampuan SFTP dengan titik akhir yang dapat diakses publik menggunakan penyimpanan Amazon S3, menambahkan pengguna dengan otentikasi yang dikelola layanan, dan mentransfer file dengan Cyberduck.

Topik

- Prasyarat
- Langkah 1: Masuk ke AWS Transfer Family konsol
- Langkah 2: Buat server berkemampuan SFTP
- Langkah 3: Tambahkan pengguna yang dikelola layanan
- Langkah 4: Transfer file menggunakan klien

Prasyarat

Sebelum Anda mulai, pastikan untuk melengkapi persyaratan diPrasyarat. Sebagai bagian dari penyiapan ini, Anda membuat bucket Amazon Simple Storage Service (Amazon S3) dan AWS Identity and Access Management peran pengguna (IAM).

Ada izin yang diperlukan untuk menggunakan AWS Transfer Family konsol, dan ada izin yang diperlukan untuk mengonfigurasi AWS layanan lain yang digunakan Transfer Family, seperti Amazon Simple Storage Service, Amazon Elastic File System AWS Certificate Manager, dan Amazon Route 53. Misalnya, untuk pengguna yang mentransfer file masuk dan keluar AWS menggunakan Transfer Family, AmazonS3 FullAccess memberikan izin untuk menyiapkan dan menggunakan bucket Amazon S3. Beberapa izin dalam kebijakan ini diperlukan untuk membuat bucket Amazon S3.

Untuk menggunakan konsol Transfer Family, Anda memerlukan yang berikut:

- AWSTransferConsoleFullAccessmemberikan izin bagi pengguna SFTP Anda untuk membuat sumber daya Transfer Family.
- IAM FullAccess (atau khususnya kebijakan yang memungkinkan pembuatan peran IAM) hanya diperlukan jika Anda ingin Transfer Family secara otomatis membuat peran logging untuk server Anda di Amazon CloudWatch Logs atau peran pengguna untuk pengguna yang masuk ke server.
- Untuk membuat dan menghapus jenis server VPC, Anda perlu menambahkan tindakan ec2: CreateVpcEndpoint dan ec2: DeleteVpcEndpoints ke kebijakan Anda.

> **Note**
>
> Kebijakan AmazonS3 FullAccess dan IAM FullAccess sendiri tidak diperlukan untuk penggunaan umum. AWS Transfer Family Mereka disajikan di sini sebagai cara sederhana untuk memastikan bahwa semua izin yang Anda butuhkan tercakup. Selain itu, ini adalah

kebijakan AWS terkelola, yang merupakan kebijakan standar yang tersedia untuk semua AWS pelanggan. Anda dapat melihat izin individual dalam kebijakan ini dan menentukan set minimal yang Anda perlukan untuk tujuan Anda.

Langkah 1: Masuk ke AWS Transfer Family konsol

Untuk masuk ke Transfer Family

1. Masuk ke AWS Management Console dan buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Untuk ID Akun atau alias, masukkan ID untuk Anda Akun AWS.
3. Untuk nama pengguna IAM, masukkan nama peran pengguna yang Anda buat untuk Transfer Family.
4. Untuk Kata Sandi, masukkan kata sandi AWS akun Anda.
5. Pilih Masuk.

Langkah 2: Buat server berkemampuan SFTP

Secure Shell (SSH) File Transfer Protocol (SFTP) adalah protokol jaringan yang digunakan untuk transfer data yang aman melalui internet. Protokol ini mendukung fungsionalitas keamanan dan otentikasi penuh SSH. Ini banyak digunakan untuk bertukar data, termasuk informasi sensitif antara mitra bisnis di berbagai industri seperti layanan keuangan, perawatan kesehatan, ritel, dan periklanan.

Untuk membuat server berkemampuan SFTP

1. Pilih Server dari panel Navigasi lalu pilih Buat server.
2. Di Pilih protokol, pilih SFTP, lalu pilih Berikutnya.
3. Di Pilih penyedia identitas, pilih Layanan yang dikelola untuk menyimpan identitas dan kunci pengguna di Transfer Family, lalu pilih Berikutnya.
4. Di Pilih titik akhir, lakukan hal berikut:
    a. Untuk tipe Endpoint, pilih tipe titik akhir yang dapat diakses publik.
    b. Untuk nama host Kustom, pilih Tidak Ada.
    c. Pilih Berikutnya.

5. Di Pilih domain, pilih Amazon S3.
6. Di Konfigurasikan detail tambahan, lakukan hal berikut:
   a. Untuk CloudWatch logging, pilih Buat peran baru agar Transfer Family membuat peran IAM secara otomatis, selama Anda memiliki izin yang tepat untuk membuat peran baru. Peran IAM yang dibuat disebut`AWSTransferLoggingAccess`.
   b. Untuk opsi algoritma kriptografi, pilih kebijakan keamanan yang berisi algoritma kriptografi yang diaktifkan untuk digunakan oleh server Anda. Kebijakan keamanan default adalah`TransferSecurityPolicy-2020-06`.
   c. Pilih Berikutnya.
7. Di Tinjau dan buat, pilih Buat server. Anda dibawa ke halaman Server.

Diperlukan beberapa menit sebelum status server baru Anda berubah menjadi Online. Pada saat itu, server Anda dapat melakukan operasi file, tetapi Anda harus membuat pengguna terlebih dahulu.

Langkah 3: Tambahkan pengguna yang dikelola layanan

Untuk menambahkan pengguna ke server berkemampuan SFTP

1. Pada halaman Server, pilih kotak centang server yang ingin Anda tambahkan pengguna.
2. Pilih Tambahkan pengguna.
3. Di bagian Konfigurasi pengguna, untuk Nama Pengguna, masukkan nama pengguna. Nama pengguna ini harus minimal 3 dan maksimal 100 karakter. Anda dapat menggunakan karakter berikut dalam nama pengguna: a—z, A-Z, 0-9, garis bawah '_', tanda hubung '-', periode ' . ', dan pada tanda “@”. Nama pengguna tidak dapat dimulai dengan tanda hubung, titik, atau tanda saat.
4. Untuk Access, pilih peran IAM yang sebelumnya Anda buat yang menyediakan akses ke bucket Amazon S3 Anda.

   Anda membuat peran IAM ini menggunakan prosedur diBuat peran dan kebijakan IAM. Peran IAM tersebut mencakup kebijakan IAM yang menyediakan akses ke bucket Amazon S3 Anda. Ini juga mencakup hubungan kepercayaan dengan AWS Transfer Family layanan, yang didefinisikan dalam kebijakan IAM lain.

> **Note**
>
> Peran IAM untuk pengguna yang dikelola layanan harus berisi izin untuk mengakses bucket yang diinginkan. Izin untuk mengakses bucket yang diinginkan tercakup dalam S3 FullAccess yang memberikan izin tingkat administrator ke sumber daya S3.

5. Untuk Kebijakan, pilih Tidak Ada.
6. Untuk direktori Home, pilih bucket Amazon S3 untuk menyimpan data yang akan ditransfer. AWS Transfer Family Masukkan jalur ke home direktori tempat pengguna Anda mendarat saat mereka masuk menggunakan klien mereka.

   Jika Anda membiarkan parameter ini kosong, `root` direktori bucket Amazon S3 Anda akan digunakan. Dalam hal ini, pastikan bahwa peran IAM Anda menyediakan akses ke `root` direktori ini.

   > **Note**
   >
   > Kami menyarankan Anda memilih jalur direktori yang berisi nama pengguna pengguna, yang memungkinkan Anda menggunakan kebijakan sesi secara efektif. Kebijakan sesi membatasi akses pengguna di bucket Amazon S3 ke direktori pengguna tersebut. home

7. Untuk Dibatasi, pilih kotak centang agar pengguna Anda tidak dapat mengakses apa pun di luar folder itu dan tidak dapat melihat bucket atau nama folder Amazon S3.

   > **Note**
   >
   > Saat menugaskan pengguna direktori home dan membatasi pengguna ke direktori home itu, ini harus cukup untuk mengunci akses pengguna ke folder yang ditunjuk. Gunakan kebijakan sesi saat Anda perlu menerapkan kontrol lebih lanjut.

8. Untuk kunci publik SSH, masukkan bagian kunci SSH publik dari key pair SSH.

   Kunci Anda divalidasi oleh layanan sebelum Anda dapat menambahkan pengguna baru Anda.

> **Important**
>
> Format kunci publik SSH adalah`ssh-rsa <string>`. Untuk petunjuk tentang cara membuat key pair SSH, lihatBuat kunci SSH untuk pengguna yang dikelola layanan.

9. (Opsional) Untuk Kunci dan Nilai, masukkan satu atau beberapa tag sebagai pasangan nilai kunci, dan pilih Tambahkan tag.
10. Pilih Tambah untuk menambahkan pengguna baru Anda ke server yang Anda pilih.

    Pengguna baru muncul di bagian Pengguna pada halaman detail Server.

Langkah 4: Transfer file menggunakan klien

Anda mentransfer file melalui AWS Transfer Family layanan dengan menentukan operasi transfer di klien. AWS Transfer Family mendukung beberapa klien. Untuk detailnya, lihat Mentransfer file melalui titik akhir server menggunakan klien

Bagian ini berisi prosedur untuk menggunakan Cyberduck dan OpenSSH.

Topik

- Gunakan Cyberduck
- Gunakan OpenSSH

Gunakan Cyberduck

Untuk mentransfer file AWS Transfer Family menggunakan Cyberduck

1. Buka klien Cyberduck.
2. Pilih Buka Koneksi.
3. Dalam kotak dialog Open Connection, pilih SFTP (SSH File Transfer Protocol).
4. Untuk Server, masukkan endpoint server Anda. Titik akhir server terletak di halaman detail Server, lihatLihat detail server SFTP, FTPS, dan FTP.
5. Untuk nomor Port, masukkan **22** untuk SFTP.
6. Untuk Nama Pengguna, masukkan nama untuk pengguna yang Anda buatMengelola pengguna untuk titik akhir server.

7. Untuk SSH Private Key, pilih atau masukkan kunci pribadi SSH.
8. Pilih Hubungkan.
9. Lakukan transfer file Anda.

   Tergantung di mana file Anda berada, lakukan salah satu hal berikut:

   - Di direktori lokal Anda (sumber), pilih file yang ingin Anda transfer, dan seret dan jatuhkan ke direktori Amazon S3 (target).
   - Di direktori Amazon S3 (sumber), pilih file yang ingin Anda transfer, dan seret dan jatuhkan ke direktori lokal Anda (target).

Gunakan OpenSSH

Gunakan instruksi yang mengikuti untuk mentransfer file dari baris perintah menggunakan OpenSSH.

> **Note**
>
> Klien ini hanya berfungsi dengan server berkemampuan SFTP.

Untuk mentransfer file AWS Transfer Family menggunakan utilitas baris perintah OpenSSH

1. Di Linux atau Macintosh, buka terminal perintah.
2. Pada prompt, masukkan perintah berikut: `% sftp -i transfer-key sftp_user@service_endpoint`

   Pada perintah sebelumnya, `sftp_user` adalah nama pengguna dan `transfer-key` merupakan kunci pribadi SSH. Di sini, `service_endpoint` adalah titik akhir server seperti yang ditunjukkan di AWS Transfer Family konsol untuk server yang dipilih.

   `sftp`Prompt akan muncul.
3. (Opsional) Untuk melihat direktori home pengguna, masukkan perintah berikut pada `sftp` prompt: `sftp> pwd`
4. Pada baris berikutnya, masukkan teks berikut: `sftp> cd /mybucket/home/sftp_user`

   Dalam latihan memulai ini, bucket Amazon S3 ini adalah target transfer file.
5. Pada baris berikutnya, masukkan perintah berikut: `sftp> put filename.txt`

`put`Perintah mentransfer file ke bucket Amazon S3.

Pesan seperti berikut ini muncul, menunjukkan bahwa transfer file sedang berlangsung, atau selesai.

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

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

Menyiapkan alur kerja terkelola untuk mendekripsi file

Tutorial ini menggambarkan cara mengatur alur kerja terkelola yang berisi langkah dekripsi. Tutorial ini juga menunjukkan cara mengunggah file terenkripsi ke bucket Amazon S3 dan kemudian melihat file yang didekripsi di bucket yang sama.

> **Note**
>
> Blog AWS penyimpanan memiliki posting yang menjelaskan cara mengenkripsi dan mendekripsi file, mengenkripsi dan mendekripsi file dengan PGP dan. AWS Transfer Family

Topik

- Langkah 1: Konfigurasikan peran eksekusi
- Langkah 2: Buat alur kerja terkelola
- Langkah 3: Tambahkan alur kerja ke server dan buat pengguna
- Langkah 4: Buat key pair PGP
- Langkah 5: Simpan kunci pribadi PGP di AWS Secrets Manager
- Langkah 6: Enkripsi file
- Langkah 7: Jalankan alur kerja dan lihat hasilnya

Langkah 1: Konfigurasikan peran eksekusi

Buat peran eksekusi AWS Identity and Access Management (IAM) yang dapat digunakan Transfer Family untuk meluncurkan alur kerja. Proses pembuatan peran eksekusi dijelaskan dalamKebijakan IAM untuk alur kerja.

> **Note**
>
> Sebagai bagian dari menciptakan peran eksekusi, pastikan untuk membangun hubungan kepercayaan antara peran eksekusi dan Transfer Family, seperti yang dijelaskan dalamUntuk membangun hubungan kepercayaan.

Kebijakan peran eksekusi berikut berisi semua izin yang diperlukan untuk berhasil menjalankan alur kerja yang akan Anda buat dalam tutorial ini. Untuk menggunakan kebijakan contoh ini, ganti *`user input placeholders`* dengan informasi Anda sendiri. Ganti *`DOC-EXAMPLE-BUCKET`* dengan nama bucket Amazon S3 tempat Anda akan mengunggah file terenkripsi Anda.

> **Note**
>
> Tidak setiap alur kerja memerlukan setiap izin yang tercantum dalam contoh ini. Anda dapat membatasi izin berdasarkan jenis langkah dalam alur kerja spesifik Anda. Izin yang diperlukan untuk setiap jenis langkah yang telah ditentukan dijelaskan dalam. Gunakan langkah-langkah yang telah ditentukan Izin yang diperlukan untuk langkah kustom dijelaskan diIzin IAM untuk langkah khusus.

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

Langkah 2: Buat alur kerja terkelola

Sekarang Anda perlu membuat alur kerja yang berisi langkah dekripsi.

Untuk membuat alur kerja yang berisi langkah dekripsi

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi kiri, pilih Alur kerja, lalu pilih Buat alur kerja.
3. Masukkan detail berikut:
   - Masukkan deskripsi, misalnya**`Decrypt workflow example`**.
   - Di bagian Langkah nominal, pilih Tambah langkah.
4. Untuk Pilih jenis langkah, pilih Dekripsi file, lalu pilih Berikutnya.
5. Dalam kotak dialog Konfigurasi parameter, tentukan yang berikut ini:
   - Masukkan nama langkah deskriptif, misalnya,**`decrypt-step`**. Spasi tidak diperbolehkan dalam nama langkah.
   - Untuk Tujuan untuk file yang didekripsi, pilih Amazon S3.
   - Untuk nama bucket Tujuan, pilih bucket Amazon S3 yang sama dengan yang Anda tentukan seperti *DOC-EXAMPLE-BUCKET* dalam kebijakan IAM yang Anda buat di Langkah 1.

- Untuk awalan kunci Destination, masukkan nama awalan (folder) tempat Anda ingin menyimpan file yang didekripsi di bucket tujuan, misalnya,. **`decrypted-files/`**

  > **Note**
  >
  > Pastikan untuk menambahkan trailing slash (**`/`**) ke awalan Anda.

- Untuk tutorial ini, biarkan Overwrite sudah dihapus. Saat pengaturan ini dihapus, jika Anda mencoba mendekripsi file dengan nama identik dari file yang ada, pemrosesan alur kerja berhenti, dan file baru tidak diproses.

Pilih Berikutnya untuk pindah ke layar ulasan.

6. Tinjau detail untuk langkahnya. Jika semuanya benar, pilih Buat langkah.
7. Alur kerja Anda hanya membutuhkan satu langkah dekripsi, jadi tidak ada langkah tambahan untuk mengonfigurasi. Pilih Buat alur kerja untuk membuat alur kerja baru.

Perhatikan ID alur kerja untuk alur kerja baru Anda. Anda akan memerlukan ID ini untuk langkah selanjutnya. Tutorial ini menggunakan *`w-1234abcd5678efghi`*sebagai contoh ID alur kerja.

Langkah 3: Tambahkan alur kerja ke server dan buat pengguna

Sekarang Anda memiliki alur kerja dengan langkah dekripsi, Anda harus mengaitkannya dengan server Transfer Family. Tutorial ini menunjukkan cara melampirkan alur kerja ke server Transfer Family yang ada. Atau, Anda dapat membuat server baru untuk digunakan dengan alur kerja Anda.

Setelah Anda melampirkan alur kerja ke server, Anda harus membuat pengguna yang dapat SFTP ke server dan memicu alur kerja untuk berjalan.

Untuk mengonfigurasi server Transfer Family untuk menjalankan alur kerja

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi kiri, pilih Server, lalu pilih server dari daftar. Pastikan server ini mendukung protokol SFTP.
3. Pada halaman detail untuk server, gulir ke bawah ke bagian Detail tambahan, lalu pilih Edit.
4. Pada halaman Edit detail tambahan, di bagian Alur kerja terkelola, pilih alur kerja Anda, dan pilih peran eksekusi yang sesuai.

- Untuk Alur Kerja untuk upload file lengkap, pilih alur kerja yang Anda buatLangkah 2: Buat alur kerja terkelola, misalnya,. **`w-1234abcd5678efghi`**
- Untuk peran eksekusi alur kerja terkelola, pilih peran IAM yang Anda buat. Langkah 1: Konfigurasikan peran eksekusi

5. Gulir ke bagian bawah halaman, dan pilih Simpan untuk menyimpan perubahan Anda.

Catat ID untuk server yang Anda gunakan. Nama AWS Secrets Manager rahasia yang Anda gunakan untuk menyimpan kunci PGP Anda sebagian didasarkan pada ID server.

Untuk menambahkan pengguna yang dapat memicu alur kerja

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi kiri, pilih Server, lalu pilih server yang Anda gunakan untuk mendekripsi alur kerja.
3. Pada halaman detail server, gulir ke bawah ke bagian Pengguna, dan pilih Tambah pengguna.
4. Untuk pengguna baru Anda, masukkan detail berikut:
   - Untuk Nama Pengguna, masukkan**`decrypt-user`**.
   - Untuk Peran, pilih peran pengguna yang dapat mengakses server Anda.
   - Untuk direktori Home, pilih bucket Amazon S3 yang Anda gunakan sebelumnya, misalnya,. *`DOC-EXAMPLE-BUCKET`*
   - Untuk kunci publik SSH, tempelkan kunci publik yang sesuai dengan kunci pribadi yang Anda miliki. Untuk detailnya, lihat Buat kunci SSH untuk pengguna yang dikelola layanan.
5. Pilih Tambah untuk menyimpan pengguna baru Anda.

Catat nama pengguna Transfer Family Anda untuk server ini. Rahasianya sebagian didasarkan pada nama pengguna. Untuk kesederhanaan, tutorial ini menggunakan rahasia default yang dapat digunakan oleh setiap pengguna server.

Langkah 4: Buat key pair PGP

Gunakan salah satu klien PGP yang didukung untuk menghasilkan key pair PGP. Proses ini dijelaskan secara rinci dalamHasilkan kunci PGP.

Untuk menghasilkan key pair PGP

1. Untuk tutorial ini, Anda dapat menggunakan gpg (GnuPG) versi 2.0.22 klien untuk menghasilkan key pair PGP yang menggunakan RSA sebagai algoritma enkripsi. Untuk klien ini, jalankan perintah berikut, dan berikan alamat email dan frasa sandi. Anda dapat menggunakan nama atau alamat email apa pun yang Anda suka. Pastikan Anda mengingat nilai yang Anda gunakan, karena Anda harus memasukkannya nanti dalam tutorial.

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

   > **Note**
   >
   > Jika Anda menggunakan GnuPG versi 2.3.0 atau yang lebih baru, Anda harus menjalankannya. `gpg --full-gen-key` Ketika diminta untuk jenis kunci yang akan dibuat, pilih RSA atau ECC. Namun, jika Anda memilih ECC, pastikan untuk memilih salah satu NIST atau BrainPool untuk kurva elips. Jangan memilihCurve 25519.

2. Ekspor kunci pribadi dengan menjalankan perintah berikut. Ganti *user@example.com* dengan alamat email yang Anda gunakan saat membuat kunci.

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

   Perintah ini mengekspor kunci pribadi ke **workflow-tutorial-key.pgp** file. Anda dapat memberi nama file output apa pun yang Anda suka. Anda juga dapat menghapus file kunci pribadi setelah Anda menambahkannya AWS Secrets Manager.

Langkah 5: Simpan kunci pribadi PGP di AWS Secrets Manager

Anda perlu menyimpan kunci pribadi di Secrets Manager, dengan cara yang sangat spesifik, sehingga alur kerja dapat menemukan kunci pribadi saat alur kerja menjalankan langkah dekripsi pada file yang diunggah.

> **Note**
>
> Ketika Anda menyimpan rahasia di Secrets Manager, Anda Akun AWS dikenakan biaya. Untuk informasi tentang harga, lihat AWS Secrets Manager Harga.

Untuk menyimpan kunci pribadi PGP di Secrets Manager

1. Masuk ke AWS Management Console dan buka AWS Secrets Manager konsol di https://console.aws.amazon.com/secretsmanager/.
2. Pada panel navigasi kiri, pilih Rahasia.
3. Pada halaman Rahasia, pilih Simpan rahasia baru.
4. Pada halaman Pilih jenis rahasia, untuk tipe Rahasia, pilih Jenis rahasia lainnya.
5. Di bagian pasangan kunci/Nilai, pilih tab kunci/Nilai.
   - Kunci — Masukkan**`PGPPrivateKey`**.
   - nilai — Tempelkan teks kunci pribadi Anda ke bidang nilai.
6. Pilih Tambah baris, dan di bagian pasangan kunci/Nilai, pilih tab kunci/Nilai.
   - Kunci — Masukkan**`PGPPassphrase`**.
   - value — Masukkan kata sandi yang Anda gunakan saat membuat key pair PGP. Langkah 4: Buat key pair PGP
7. Pilih Berikutnya.
8. Pada halaman Konfigurasi rahasia, masukkan nama dan deskripsi untuk rahasia Anda. Untuk tutorial ini, Anda dapat membuat rahasia default yang dapat digunakan semua pengguna. Dengan asumsi bahwa ID server adalah *`s-11112222333344445`*, beri nama rahasianya**`aws/transfer/`***`s-11112222333344445`***`/@pgp-default`**. Ganti *`s-11112222333344445`* dengan ID server Transfer Family Anda. Masukkan deskripsi untuk rahasia Anda.

   > **Note**
   >
   > Untuk membuat rahasia hanya untuk pengguna yang Anda buat sebelumnya, beri nama rahasianya**`aws/transfer/`***`s-11112222333344445`***`/decrypt-user`**.

9. Pilih Berikutnya, dan kemudian terima default pada halaman Konfigurasi rotasi. Lalu pilih Selanjutnya.
10. Pada halaman Review, pilih Store untuk membuat dan menyimpan rahasia.

Untuk informasi selengkapnya tentang menambahkan kunci pribadi PGP ke Secrets Manager, lihat Menggunakan AWS Secrets Manager untuk menyimpan kunci PGP Anda.

Langkah 6: Enkripsi file

Gunakan gpg program untuk mengenkripsi file untuk digunakan dalam alur kerja Anda. Jalankan perintah berikut untuk mengenkripsi file:

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

Sebelum menjalankan perintah ini, perhatikan hal berikut:

- Untuk `-r` argumennya, ganti *marymajor@example.com* dengan alamat email yang Anda gunakan saat membuat key pair PGP.
- `--openpgp`Bendera adalah opsional. Bendera ini membuat file terenkripsi sesuai dengan standar OpenPGP RFC4880.
- Perintah ini membuat file bernama **testfile.txt.gpg** di lokasi yang sama dengan**testfile.txt**.

Langkah 7: Jalankan alur kerja dan lihat hasilnya

Untuk menjalankan alur kerja, Anda terhubung ke server Transfer Family dengan pengguna yang Anda buat di Langkah 3. Kemudian Anda dapat melihat di bucket Amazon S3 yang Anda tentukan di Langkah 2.5, konfigurasikan parameter tujuan untuk melihat file yang didekripsi.

Untuk menjalankan alur kerja dekripsi

1. Buka terminal perintah.
2. Jalankan perintah berikut, ganti *your-endpoint* dengan endpoint Anda yang sebenarnya, dan *transfer-key* dengan kunci pribadi SSH pengguna Anda:

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

   Misalnya, jika kunci pribadi disimpan`~/.ssh/decrypt-user`, dan titik akhir Anda, perintahnya adalah sebagai berikut: `s-11112222333344445.server.transfer.us-east-2.amazonaws.com`

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

3. Jalankan perintah pwd. Jika berhasil, perintah ini akan mengembalikan yang berikut:

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

   Direktori Anda mencerminkan nama bucket Amazon S3 Anda.

4. Jalankan perintah berikut untuk mengunggah file dan memicu alur kerja untuk dijalankan:

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

5. Untuk tujuan file yang didekripsi, Anda menentukan `decrypted-files/` folder saat Anda membuat alur kerja. Sekarang, Anda dapat menavigasi ke folder itu dan daftar isinya.

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

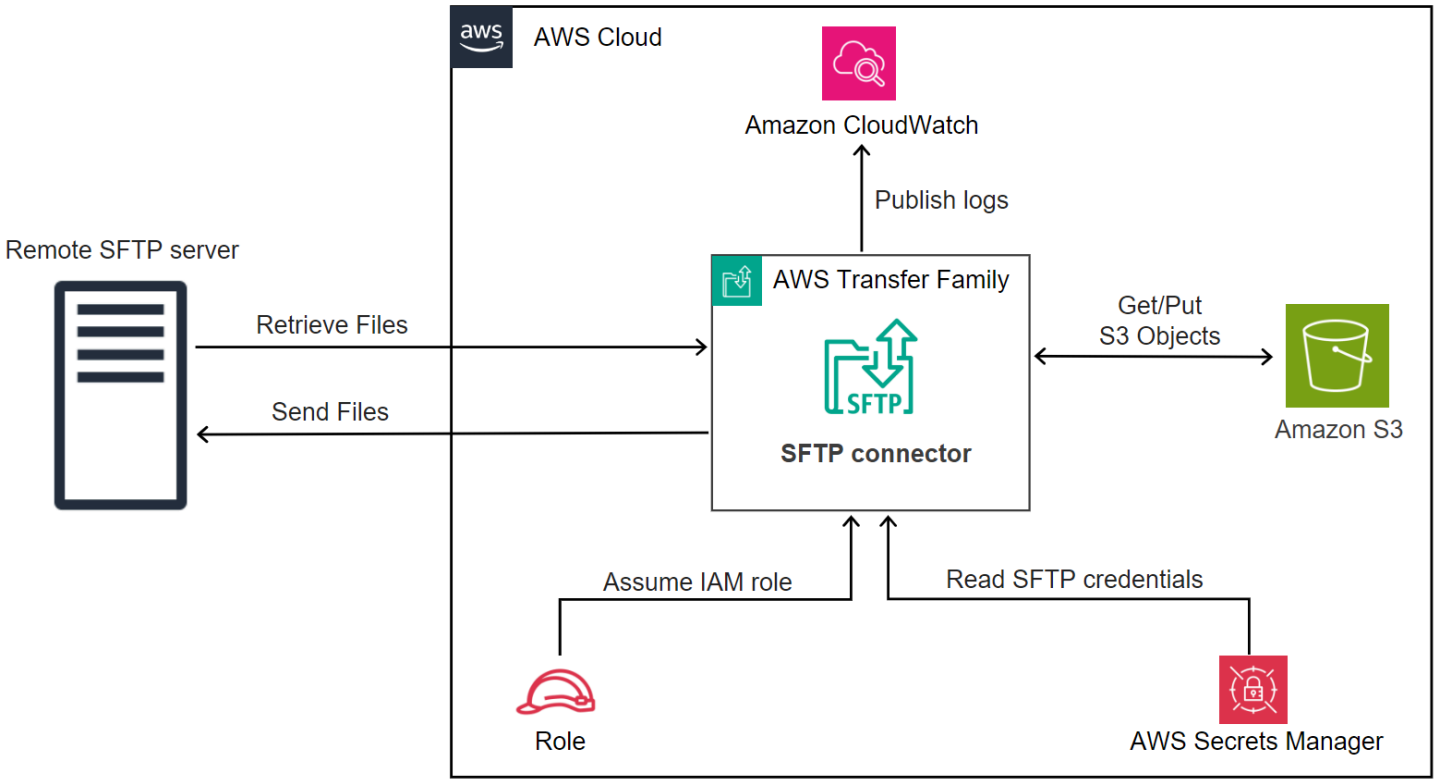
   Jika berhasil, `ls` perintah mencantumkan `testfile.txt` file. Anda dapat mengunduh file ini dan memverifikasi bahwa itu sama dengan file asli yang Anda enkripsi sebelumnya.

Menyiapkan dan menggunakan konektor SFTP

Tujuan dari konektor adalah untuk membangun hubungan antara AWS penyimpanan Anda dan server SFTP mitra. Anda dapat mengirim file dari Amazon S3 ke tujuan eksternal milik mitra. Anda juga dapat menggunakan konektor SFTP untuk mengambil file dari server SFTP mitra.

Tutorial ini menggambarkan cara mengatur konektor SFTP, dan kemudian mentransfer file antara penyimpanan Amazon S3 dan server SFTP.

Konektor SFTP mengambil kredensi SFTP dari AWS Secrets Manager untuk mengautentikasi ke server SFTP jarak jauh dan membuat koneksi. Konektor mengirim file ke atau mengambil file dari server jarak jauh, dan menyimpan file di Amazon S3. Peran IAM digunakan untuk mengizinkan akses ke bucket Amazon S3 dan ke kredensil yang disimpan di Secrets Manager. Dan Anda dapat masuk ke Amazon CloudWatch.



Topik

- Langkah 1: Buat sumber daya pendukung yang diperlukan
- Langkah 2: Buat dan uji konektor SFTP
- Langkah 3: Kirim dan ambil file menggunakan konektor SFTP
- Prosedur untuk membuat server Transfer Family untuk digunakan sebagai server SFTP jarak jauh

Langkah 1: Buat sumber daya pendukung yang diperlukan

Anda dapat menggunakan konektor SFTP untuk menyalin file antara Amazon S3 dan server SFTP jarak jauh apa pun. Untuk tutorial ini, kami menggunakan AWS Transfer Family server sebagai server SFTP jarak jauh kami. Kita perlu membuat dan mengkonfigurasi sumber daya berikut:

- Buat bucket Amazon S3 untuk menyimpan file di AWS lingkungan Anda, dan untuk mengirim dan mengambil file dari server SFTP jarak jauh:. Buat ember Amazon S3
- Buat AWS Identity and Access Management peran untuk mengakses penyimpanan Amazon S3 dan rahasia kami di Secrets Manager:. Buat peran IAM dengan izin yang diperlukan

- Buat server Transfer Family yang menggunakan protokol SFTP, dan pengguna yang dikelola layanan yang menggunakan konektor SFTP untuk mentransfer file ke atau dari server SFTP: Buat server SFTP Transfer Family dan pengguna
- Buat AWS Secrets Manager rahasia yang menyimpan kredensil yang digunakan oleh konektor SFTP untuk masuk ke server SFTP jarak jauh:. Buat dan simpan rahasia di AWS Secrets Manager

Buat ember Amazon S3

Untuk membuat bucket Amazon S3

1. Masuk ke AWS Transfer Family konsol di https://console.aws.amazon.com/s3/.
2. Pilih Wilayah dan masukkan nama.

   Untuk tutorial ini, ember kami ada di**US East (N. Virginia) us-east-1**, dan namanya**sftp-server-storage-east**.
3. Terima defaultnya dan pilih Buat bucket.

Untuk detail selengkapnya tentang membuat bucket Amazon S3, lihat Bagaimana cara membuat bucket S3? di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Buat peran IAM dengan izin yang diperlukan

Untuk peran akses, buat kebijakan dengan izin berikut.

Contoh berikut memberikan izin yang diperlukan untuk mengakses *DOC-EXAMPLE-BUCKET* di Amazon S3, dan rahasia tertentu yang disimpan di Secrets Manager.

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

Ganti item sebagai berikut:

- Untuk *DOC-EXAMPLE-BUCKET, tutorialnya* menggunakan. **s3-storage-east**
- Untuk *wilayah*, tutorial menggunakan**us-east-1**.
- Untuk *account-id*, gunakan ID Anda Akun AWS .
- Untuk *SecretName-6 RandomCharacters*, kami **using sftp-connector1** untuk nama (Anda akan memiliki enam karakter acak Anda sendiri untuk rahasia Anda).

Anda juga harus memastikan bahwa peran ini berisi hubungan kepercayaan yang memungkinkan konektor mengakses sumber daya Anda saat melayani permintaan transfer pengguna Anda. Untuk detail tentang membangun hubungan kepercayaan, lihatUntuk membangun hubungan kepercayaan.

> **Note**
>
> Untuk melihat detail untuk peran yang kita gunakan untuk tutorial, lihatGabungan peran pengguna dan akses.

Buat dan simpan rahasia di AWS Secrets Manager

Kita perlu menyimpan rahasia di Secrets Manager untuk menyimpan kredensil pengguna untuk konektor SFTP Anda. Anda dapat menggunakan kata sandi, kunci pribadi SSH, atau keduanya. Untuk tutorial, kita menggunakan kunci pribadi.

> **Note**
>
> Ketika Anda menyimpan rahasia di Secrets Manager, Anda Akun AWS dikenakan biaya. Untuk informasi tentang harga, lihat AWS Secrets Manager Harga.

Sebelum Anda memulai prosedur untuk menyimpan rahasia, mengambil dan memformat kunci pribadi Anda. Kunci pribadi harus sesuai dengan kunci publik yang dikonfigurasi untuk pengguna di server SFTP jarak jauh. Untuk tutorial kami, kunci pribadi harus sesuai dengan kunci publik yang disimpan untuk pengguna uji kami di server SFTP Transfer Family yang kami gunakan sebagai server jarak jauh.

Untuk melakukan ini, jalankan perintah berikut:

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

Misalnya, jika file kunci pribadi Anda berada di`~/.ssh/sftp-testuser-privatekey`, perintahnya adalah sebagai berikut.

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

Ini menghasilkan kunci dalam format yang benar (dengan karakter baris baru yang disematkan) ke output standar. Salin teks ini di suatu tempat, karena Anda harus menempelkannya dalam prosedur berikut (pada langkah 6).

Untuk menyimpan kredensi pengguna di Secrets Manager untuk konektor SFTP

1. Masuk ke AWS Management Console dan buka AWS Secrets Manager konsol di https://console.aws.amazon.com/secretsmanager/.
2. Pada panel navigasi kiri, pilih Rahasia.
3. Pada halaman Rahasia, pilih Simpan rahasia baru.
4. Pada halaman Pilih jenis rahasia, untuk tipe Rahasia, pilih Jenis rahasia lainnya.
5. Di bagian pasangan kunci/Nilai, pilih tab kunci/Nilai.
   - Kunci — Masukkan**`Username`**.
   - nilai — Masukkan nama pengguna kami,**`sftp-testuser`**.
6. Untuk memasukkan kunci, kami sarankan Anda menggunakan tab Plaintext.
   a. Pilih Tambah baris, lalu masukkan**`PrivateKey`**.
   b. Pilih tab Plaintext. Bidang sekarang berisi teks berikut:

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

   c. Tempelkan teks untuk kunci pribadi Anda (disimpan sebelumnya) di antara tanda kutip ganda kosong (“”).

   Layar Anda akan terlihat sebagai berikut (data kunci berwarna abu-abu).

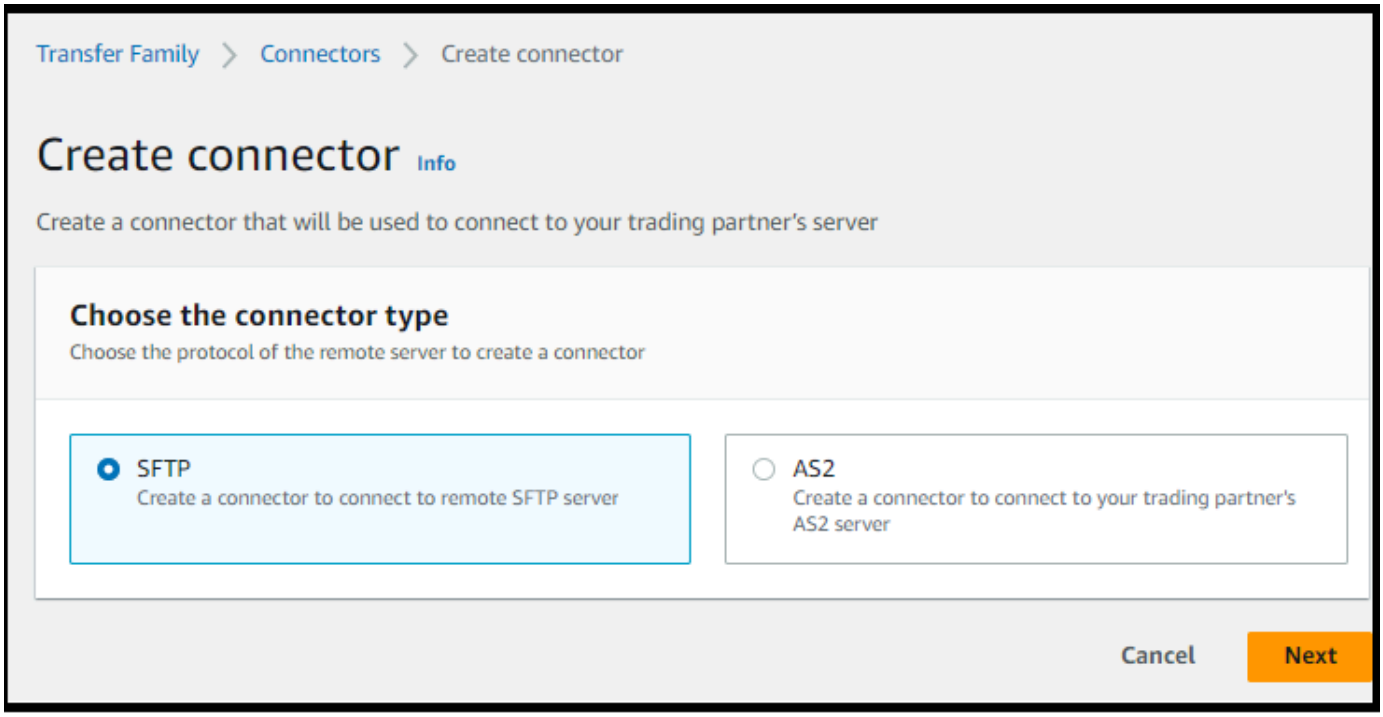
7. Pilih Berikutnya.
8. Pada halaman Konfigurasi rahasia, masukkan nama untuk rahasia Anda. Untuk tutorial ini, kami beri nama rahasianya**`aws/transfer/sftp-connector1`**.
9. Pilih Berikutnya, dan kemudian terima default pada halaman Konfigurasi rotasi. Lalu pilih Selanjutnya.
10. Pada halaman Review, pilih Store untuk membuat dan menyimpan rahasia.

Langkah 2: Buat dan uji konektor SFTP

Di bagian ini, kami membuat konektor SFTP yang menggunakan semua sumber daya yang kami buat sebelumnya. Untuk detail selengkapnya, lihat Konfigurasikan konektor SFTP.

Untuk membuat konektor SFTP

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi kiri, pilih Konektor, lalu pilih Buat konektor.
3. Pilih SFTP untuk jenis konektor untuk membuat konektor SFTP, lalu pilih Berikutnya.



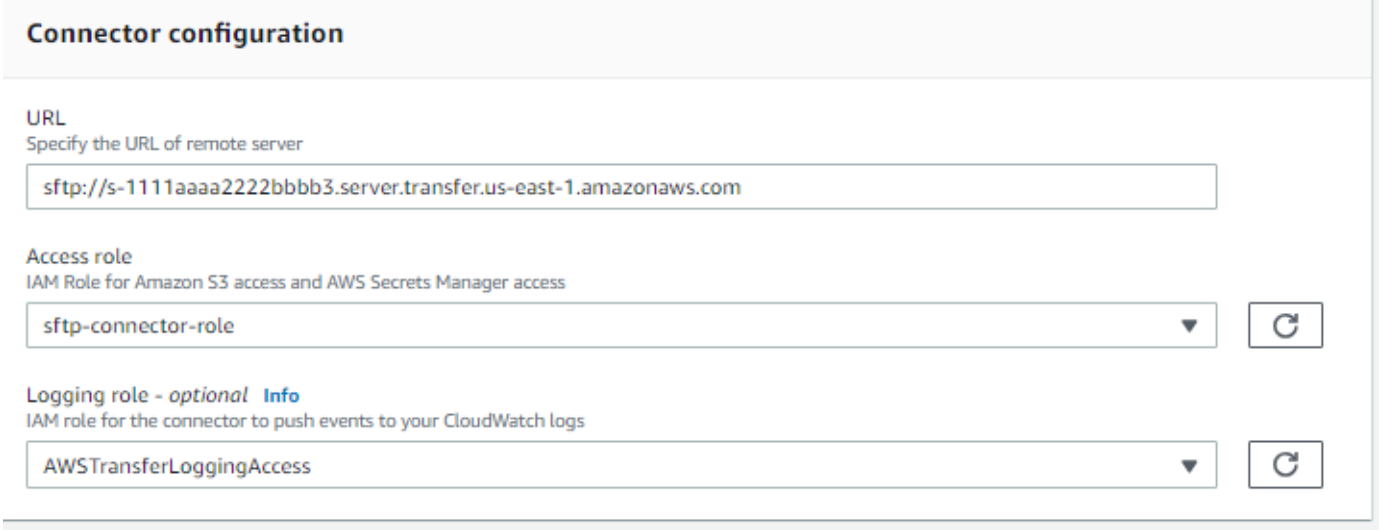
4. Di bagian Konfigurasi konektor, berikan informasi berikut:
   - Untuk URL, masukkan URL server SFTP jarak jauh. Untuk tutorialnya, kami memasukkan URL server Transfer Family yang kami gunakan sebagai server SFTP jarak jauh.

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

Ganti *1111aaaa22bbbb3* dengan ID server Transfer Family Anda.

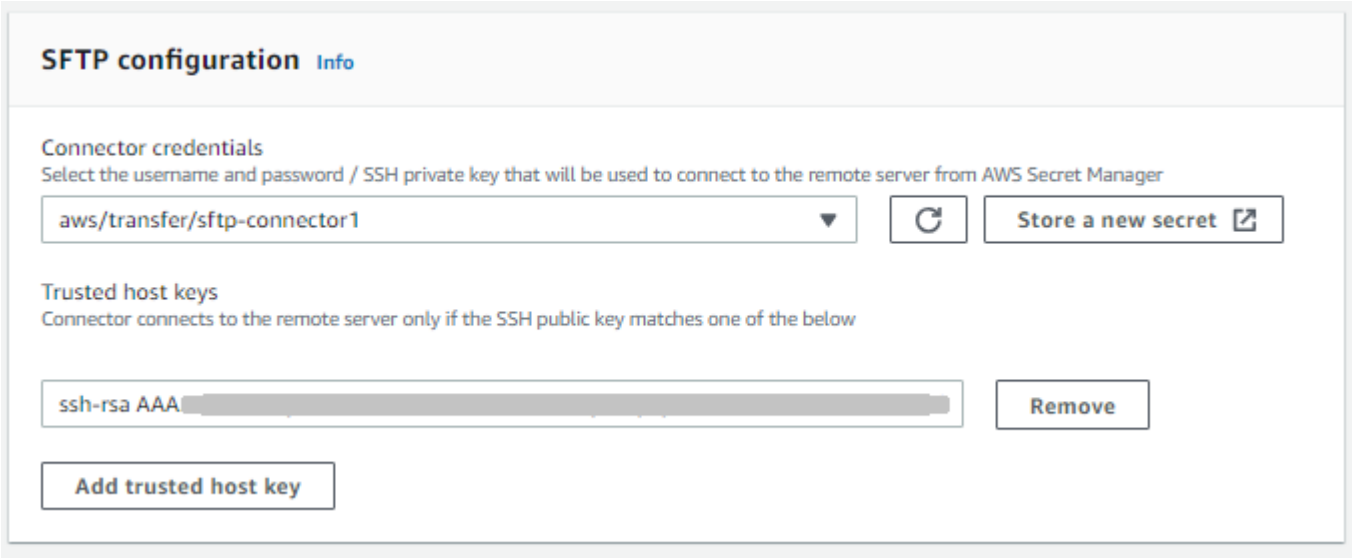
- Untuk peran Access, masukkan peran yang kita buat sebelumnya,**sftp-connector-role**.
- Untuk peran Logging, pilih**AWSTransferLoggingAccess**.

> **Note**
>
> AWSTransferLoggingAccess adalah kebijakan yang AWS dikelola. Kebijakan ini dijelaskan secara rinci dalamAWS kebijakan terkelola: AWSTransferLoggingAccess.



5. Di bagian Konfigurasi SFTP, berikan informasi berikut:

   - Untuk kredensi Connector, pilih nama sumber daya Secrets Manager Anda yang berisi kredenal SFTP. Untuk tutorialnya, pilih**aws/transfer/sftp-connector1**.
   - Untuk kunci host Tepercaya, tempel di bagian publik dari kunci host. Anda dapat mengambil kunci ini dengan menjalankan `ssh-keyscan` server SFTP Anda. Untuk detail tentang cara memformat dan menyimpan kunci host tepercaya, lihat dokumentasi tipe SftpConnectorConfigdata.



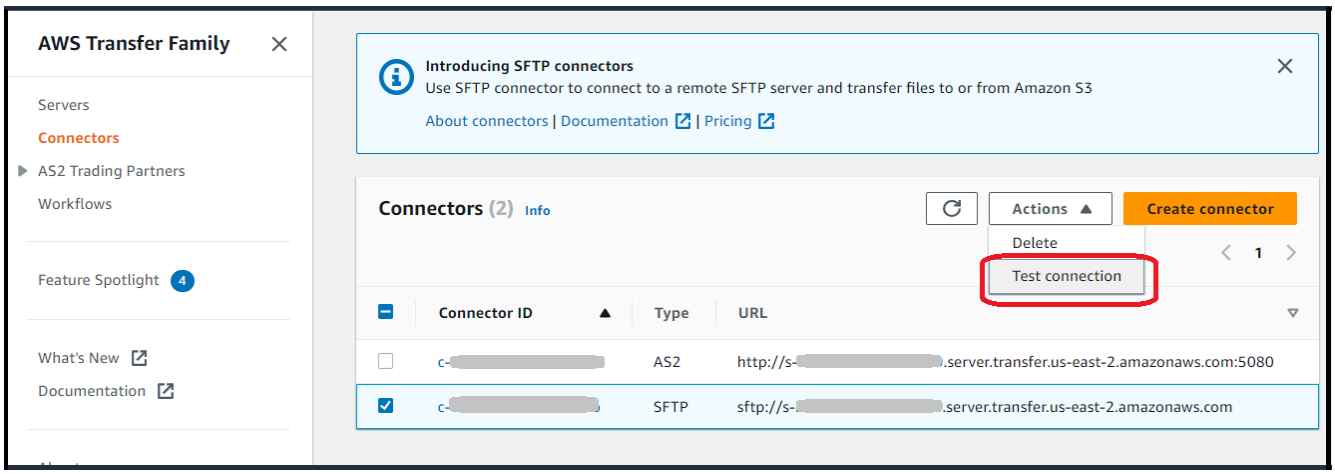
6. Setelah Anda mengkonfirmasi semua pengaturan Anda, pilih Buat konektor untuk membuat konektor SFTP.

Setelah Anda membuat konektor SFTP, kami sarankan Anda mengujinya sebelum mencoba mentransfer file apa pun menggunakan konektor baru Anda.

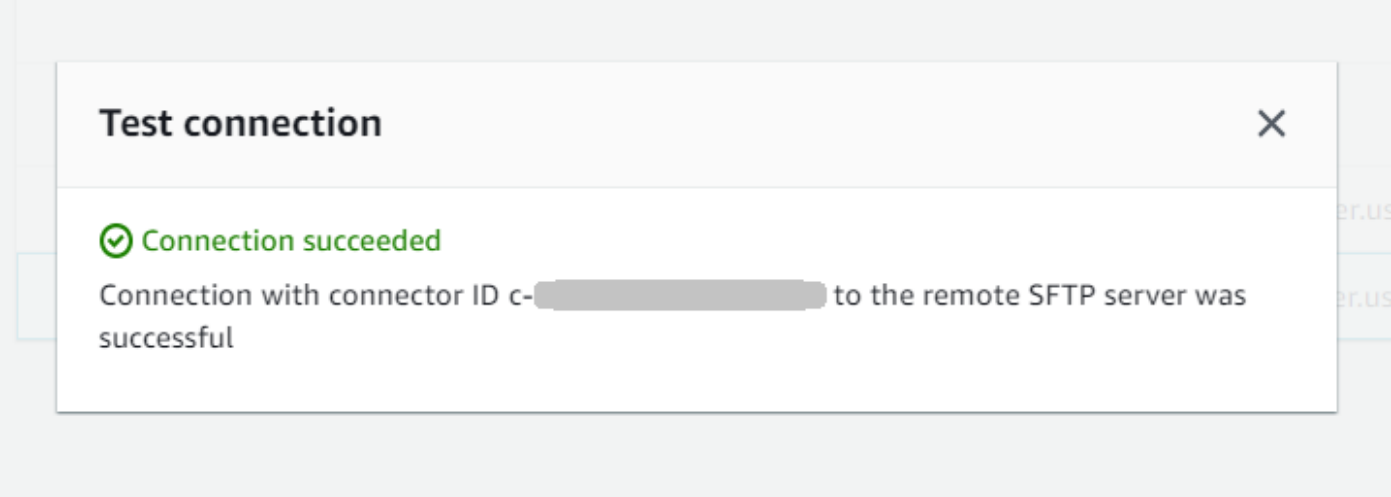
Test a connector using the console

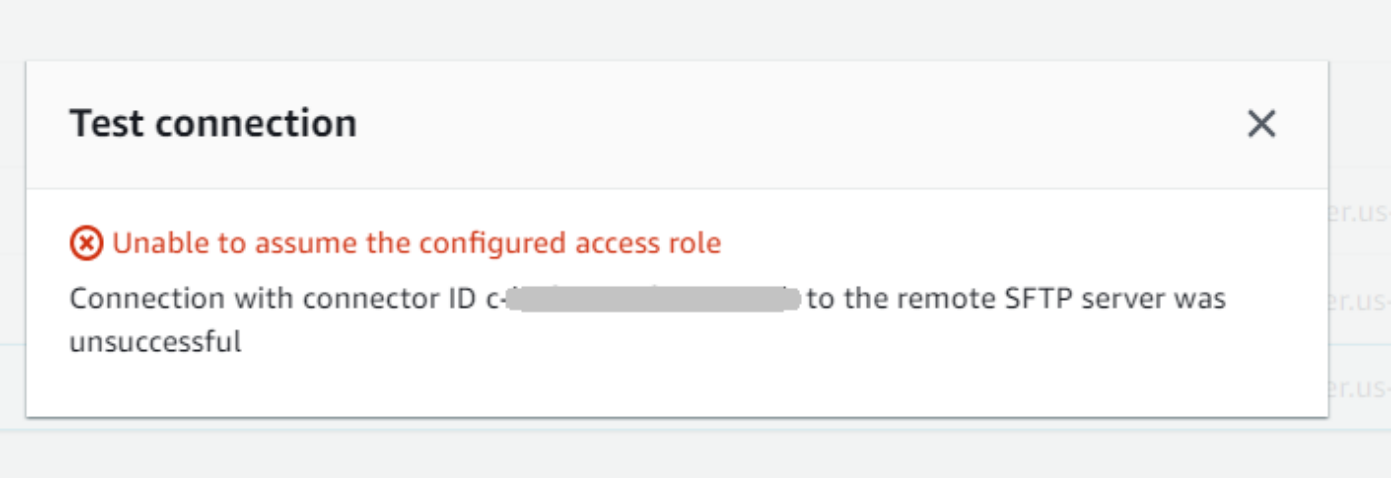
Untuk menguji konektor SFTP

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi kiri, pilih Konektor, dan pilih konektor.
3. Dari menu Tindakan, pilih Uji koneksi.



Sistem mengembalikan pesan, menunjukkan apakah tes lulus atau gagal. Jika tes gagal, sistem memberikan pesan kesalahan berdasarkan alasan pengujian gagal.





Test a connector using the CLI

Untuk menguji konektor menggunakan AWS Command Line Interface, jalankan perintah berikut pada prompt perintah (ganti *connector-id dengan ID* konektor Anda yang sebenarnya):

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

Jika tes berhasil, baris berikut dikembalikan:

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

Jika tes tidak berhasil, Anda menerima pesan kesalahan deskriptif, misalnya:

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

Langkah 3: Kirim dan ambil file menggunakan konektor SFTP

Untuk mempermudah, kami berasumsi bahwa Anda sudah memiliki file di bucket Amazon S3 Anda.

> **Note**
>
> Tutorial ini menggunakan bucket Amazon S3 untuk lokasi penyimpanan sumber dan tujuan. Jika server SFTP Anda tidak menggunakan penyimpanan Amazon S3, maka di mana pun Anda `sftp-server-storage-east` melihat dalam perintah berikut, Anda dapat mengganti jalur dengan jalur ke lokasi file yang dapat diakses dari server SFTP Anda.

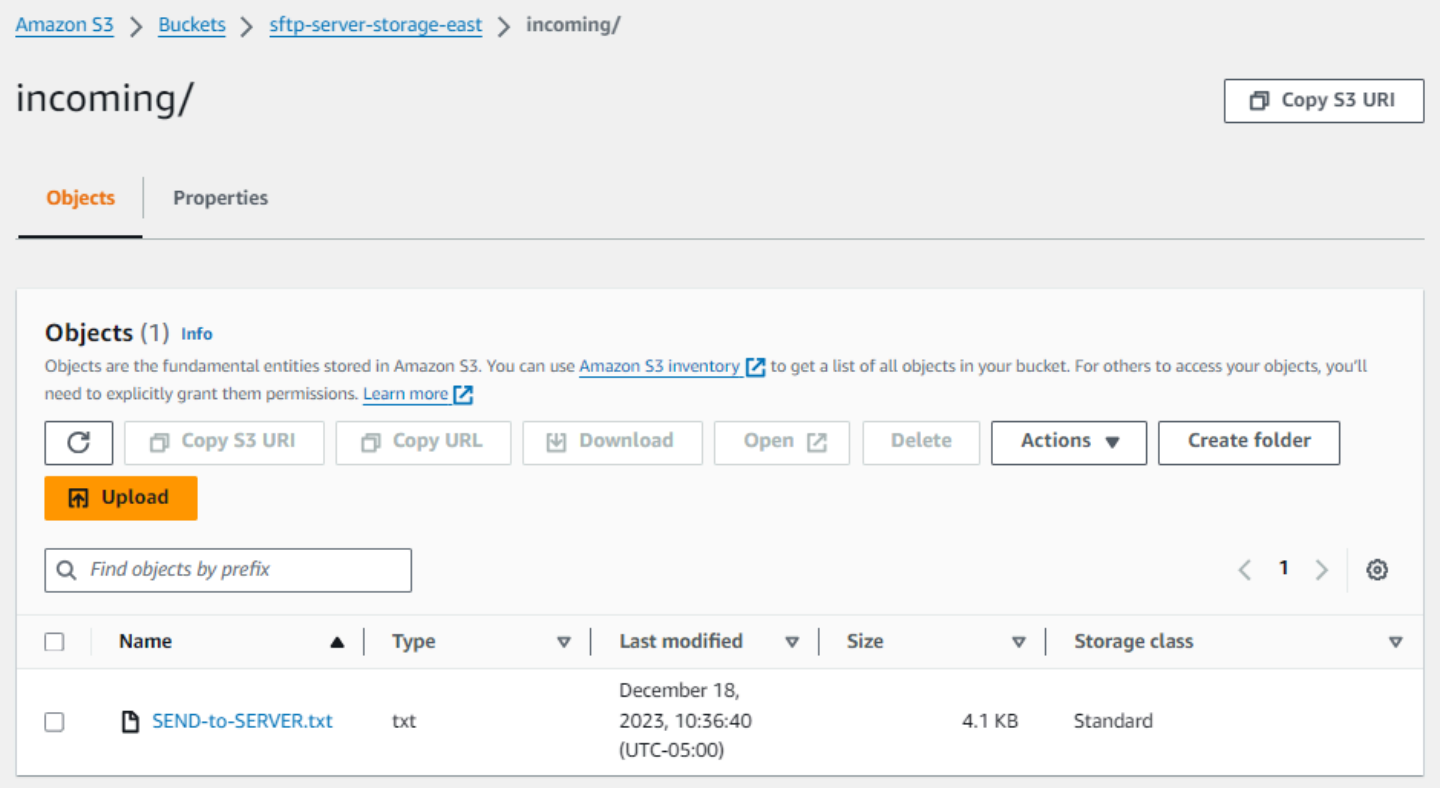
- Kami mengirim file bernama `SEND-to-SERVER.txt` dari penyimpanan Amazon S3 ke server SFTP.
- Kami mengambil file bernama `RETRIEVE-to-S3.txt` dari server SFTP ke penyimpanan Amazon S3.

> **Note**
>
> Dalam perintah berikut, ganti *`connector-id dengan ID`* konektor Anda.

Pertama, kami mengirim file dari bucket Amazon S3 kami ke server SFTP jarak jauh. Dari command prompt, jalankan perintah berikut:

```
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"
```

`sftp-server-storage-east`Ember Anda sekarang akan terlihat seperti ini.



Jika Anda tidak melihat file seperti yang diharapkan, periksa CloudWatch log Anda.

Untuk memeriksa CloudWatch log Anda

1. Buka CloudWatch konsol Amazon di https://console.aws.amazon.com/cloudwatch/
2. Pilih Grup log dari menu navigasi kiri.
3. Masukkan ID konektor Anda di bilah pencarian untuk menemukan log Anda.
4. Pilih aliran Log yang dikembalikan dari pencarian.
5. Perluas entri log terbaru.

Jika berhasil, entri log terlihat seperti berikut:

```
{
    "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",
```

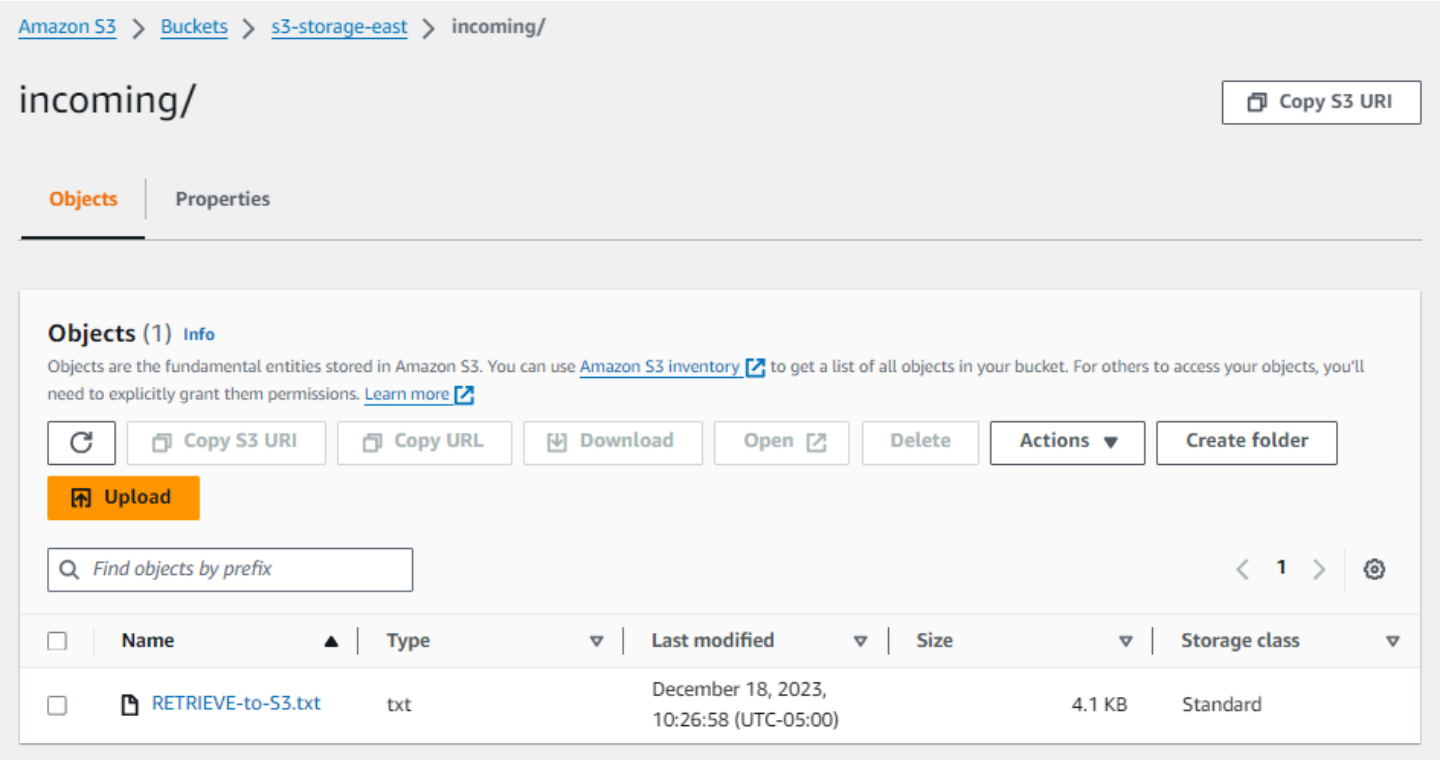
```
    "status-code": "COMPLETED",
    "start-time": "2023-12-18T15:26:56.915864Z",
    "end-time": "2023-12-18T15:26:57.298122Z",
    "account-id": "500655546075",
    "connector-arn": "arn:aws:transfer:us-east-1:500655546075:connector/connector-id",
    "remote-directory-path": "/sftp-server-storage-east/incoming"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

Jika transfer file gagal, entri log berisi pesan kesalahan yang menentukan masalah. Penyebab umum kesalahan adalah masalah dengan izin IAM dan jalur file yang salah.

Selanjutnya, kami mengambil file dari server SFTP ke bucket Amazon S3. Dari command prompt, jalankan perintah berikut:

```
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"
```

Jika transfer berhasil, bucket Amazon S3 Anda berisi file yang ditransfer, seperti yang ditunjukkan di sini.



Jika berhasil, entri log terlihat seperti berikut:

```
{
    "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"
}
```

Prosedur untuk membuat server Transfer Family untuk digunakan sebagai server SFTP jarak jauh

Berikut ini, kami menguraikan langkah-langkah untuk membuat server Transfer Family yang berfungsi sebagai server SFTP jarak jauh Anda untuk tutorial ini. Perhatikan hal berikut:

- Kami menggunakan server Transfer Family untuk mewakili server SFTP jarak jauh. Pengguna konektor SFTP yang khas memiliki server SFTP jarak jauh mereka sendiri. Lihat Buat server SFTP Transfer Family dan pengguna.
- Karena kami menggunakan server Transfer Family, kami juga menggunakan pengguna SFTP yang dikelola layanan. Dan, untuk kesederhanaan, kami menggabungkan izin yang dibutuhkan pengguna ini untuk mengakses server Transfer Family dengan izin yang mereka butuhkan untuk menggunakan konektor kami. Sekali lagi, sebagian besar kasus penggunaan konektor SFTP memiliki pengguna SFTP terpisah yang tidak terkait dengan server Transfer Family. Lihat Buat server SFTP Transfer Family dan pengguna.
- Untuk tutorial, karena kita menggunakan penyimpanan Amazon S3 untuk server SFTP jarak jauh kita, kita perlu membuat ember kedua,**`s3-storage-east`**, sehingga kita dapat mentransfer file dari satu ember ke ember lainnya.

Buat server SFTP Transfer Family dan pengguna

Sebagian besar pengguna tidak perlu membuat server SFTP Transfer Family dan pengguna, karena Anda sudah memiliki server SFTP dengan pengguna, dan Anda dapat menggunakan server ini untuk mentransfer file ke dan dari. Namun, untuk tutorial ini, untuk kesederhanaan, kami menggunakan server Transfer Family untuk berfungsi sebagai server SFTP jarak jauh.

Ikuti prosedur yang dijelaskan dalam Buat server berkemampuan SFTP untuk membuat server, dan Langkah 3: Tambahkan pengguna yang dikelola layanan untuk menambahkan pengguna. Ini adalah detail pengguna yang kami gunakan untuk tutorial:

- Buat pengguna yang dikelola layanan Anda,. `sftp-testuser`
  - Mengatur direktori home ke `/sftp-server-storage-east/sftp-testuser`
  - Saat Anda membuat pengguna, Anda menyimpan kunci publik. Kemudian, ketika Anda membuat rahasia di Secrets Manager, Anda perlu memberikan kunci pribadi yang sesuai.
- Peran:`sftp-connector-role`. Untuk tutorial, kami menggunakan peran IAM yang sama untuk pengguna SFTP kami dan untuk mengakses konektor SFTP. Saat membuat konektor untuk organisasi, Anda mungkin memiliki peran pengguna dan akses yang terpisah.
- Kunci host server: Anda perlu menggunakan kunci host server saat Anda membuat konektor. Anda dapat mengambil kunci ini dengan menjalankan `ssh-keyscan` server Anda. Misalnya, jika ID server Anda`s-1111aaaa2222bbbb3`, dan titik akhirnya masuk`us-east-1`, perintah berikut mengambil kunci host server:

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

  Salin teks ini di suatu tempat, karena Anda harus menempelkannya dalam Langkah 2: Buat dan uji konektor SFTP prosedur.

Gabungan peran pengguna dan akses

Untuk tutorial, kita menggunakan peran tunggal gabungan. Kami menggunakan peran ini baik untuk pengguna SFTP kami, maupun untuk akses ke konektor. Contoh berikut berisi rincian untuk peran ini, jika Anda ingin melakukan tugas-tugas dalam tutorial.

Contoh berikut memberikan izin yang diperlukan untuk mengakses dua bucket kami di Amazon S3, dan rahasia bernama disimpan di `aws/transfer/sftp-connector1` Secrets Manager. Untuk tutorial, peran ini diberi nama`sftp-connector-role`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowListingOfUserFolder",
            "Action": [
                "s3:ListBucket",
                "s3:GetBucketLocation"
            ],
            "Effect": "Allow",
            "Resource": [
                "arn:aws:s3:::sftp-server-storage-east",
                "arn:aws:s3:::s3-storage-east"
            ]
        },
        {
            "Sid": "HomeDirObjectAccess",
            "Effect": "Allow",
            "Action": [
                "s3:PutObject",
                "s3:GetObject",
                "s3:DeleteObject",
                "s3:DeleteObjectVersion",
                "s3:GetObjectVersion",
                "s3:GetObjectACL",
                "s3:PutObjectACL"
            ],
            "Resource": [
                "arn:aws:s3:::sftp-server-storage-east/*",
                "arn:aws:s3:::s3-storage-east/*"
            ]
        },
        {
            "Sid": "GetConnectorSecretValue",
            "Effect": "Allow",
            "Action": [
                "secretsmanager:GetSecretValue"
            ],
            "Resource": "arn:aws:secretsmanager:us-east-1:500655546075:secret:aws/
transfer/sftp-connector1-6RandomCharacters"
        }
    ]
}
```

Untuk detail lengkap tentang membuat peran untuk Transfer Family, ikuti prosedur yang dijelaskan Membuat peran pengguna untuk membuat peran.

Menyiapkan metode Amazon API Gateway sebagai penyedia identitas kustom

Tutorial ini menggambarkan cara menyiapkan metode Amazon API Gateway dan menggunakannya sebagai penyedia identitas khusus untuk mengunggah file ke AWS Transfer Family server. Tutorial ini menggunakan template stack Dasar, dan fungsi dasar lainnya sebagai contoh saja.

Topik

- Prasyarat
- Langkah 1: Buat CloudFormation tumpukan
- Langkah 2: Periksa konfigurasi metode API Gateway untuk server Anda
- Langkah 3: Lihat detail server Transfer Family
- Langkah 4: Uji apakah pengguna Anda dapat terhubung ke server
- Langkah 5: Uji koneksi SFTP dan transfer file
- Langkah 6: Batasi akses ke ember
- Perbarui Lambda jika menggunakan Amazon EFS

Prasyarat

Sebelum Anda membuat resource Transfer Family diAWS CloudFormation, buat penyimpanan dan peran pengguna Anda.

Untuk menentukan penyimpanan dan membuat peran pengguna

1. Bergantung pada penyimpanan yang Anda gunakan, lihat dokumentasi berikut:
   - Untuk membuat bucket Amazon S3, lihat Bagaimana cara membuat bucket S3? di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.
   - Untuk membuat sistem file Amazon EFS, lihatKonfigurasikan sistem file Amazon EFS.
2. Untuk membuat peran pengguna, lihat Buat peran dan kebijakan IAM

Anda memasukkan detail untuk penyimpanan Anda dan peran pengguna Anda ketika Anda membuat AWS CloudFormation tumpukan Anda di bagian berikutnya.

Langkah 1: Buat CloudFormation tumpukan

Untuk membuat AWS CloudFormation tumpukan dari template yang disediakan

1. Buka AWS CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
2. Pilih Buat tumpukan, dan pilih Dengan sumber daya baru (standar).
3. Di panel Prasyarat - Siapkan templat, pilih Template sudah siap.
4. Salin tautan ini, template tumpukan dasar, dan tempel ke bidang URL Amazon S3.
5. Klik Berikutnya.
6. Tentukan parameter, termasuk nama untuk tumpukan Anda. Pastikan untuk melakukan hal berikut:
   - Ganti nilai default untuk UserNamedan UserPassword.
   - Untuk UserHomeDirectory, masukkan detail penyimpanan (baik bucket Amazon S3 atau sistem file Amazon EFS) yang Anda buat sebelumnya.
   - Ganti default UserRoleArndengan peran pengguna yang Anda buat sebelumnya. Peran AWS Identity and Access Management (IAM) harus memiliki izin yang sesuai. Untuk contoh peran IAM dan kebijakan bucket, lihatLangkah 6: Batasi akses ke ember.
   - Jika Anda ingin mengautentikasi menggunakan kunci publik alih-alih kata sandi, masukkan kunci publik Anda di bidang UserPublicKey1. Pertama kali Anda terhubung ke server menggunakan SFTP, Anda kemudian memberikan kunci pribadi alih-alih kata sandi.
7. Pilih Berikutnya, lalu pilih Berikutnya lagi di halaman Configure stack options.
8. Tinjau detail tumpukan yang Anda buat, lalu pilih Buat tumpukan.

> **Note**
>
> Di bagian bawah halaman, di bawah Kemampuan, Anda harus mengakui bahwa AWS CloudFormation mungkin membuat sumber daya IAM.

Langkah 2: Periksa konfigurasi metode API Gateway untuk server Anda

> **Note**
>
> Untuk meningkatkan keamanan, Anda dapat mengkonfigurasi firewall aplikasi web. AWS WAFadalah firewall aplikasi web yang memungkinkan Anda memantau permintaan HTTP dan HTTPS yang diteruskan ke Amazon API Gateway. Untuk detailnya, lihat Tambahkan firewall aplikasi web.

Untuk memeriksa konfigurasi metode API Gateway untuk server Anda dan menerapkannya

1. Buka konsol API Gateway di https://console.aws.amazon.com/apigateway/.
2. Pilih API template dasar Transfer Custom Identity Provider yang dihasilkan AWS CloudFormation template.
3. Di panel Resources, pilih GET, lalu pilih Method Request.
4. Untuk Tindakan, pilih Deploy API. Untuk tahap Deployment, pilih prod, lalu pilih Deploy.

   Setelah metode API Gateway berhasil diterapkan, lihat kinerjanya di bagian Editor Panggung.

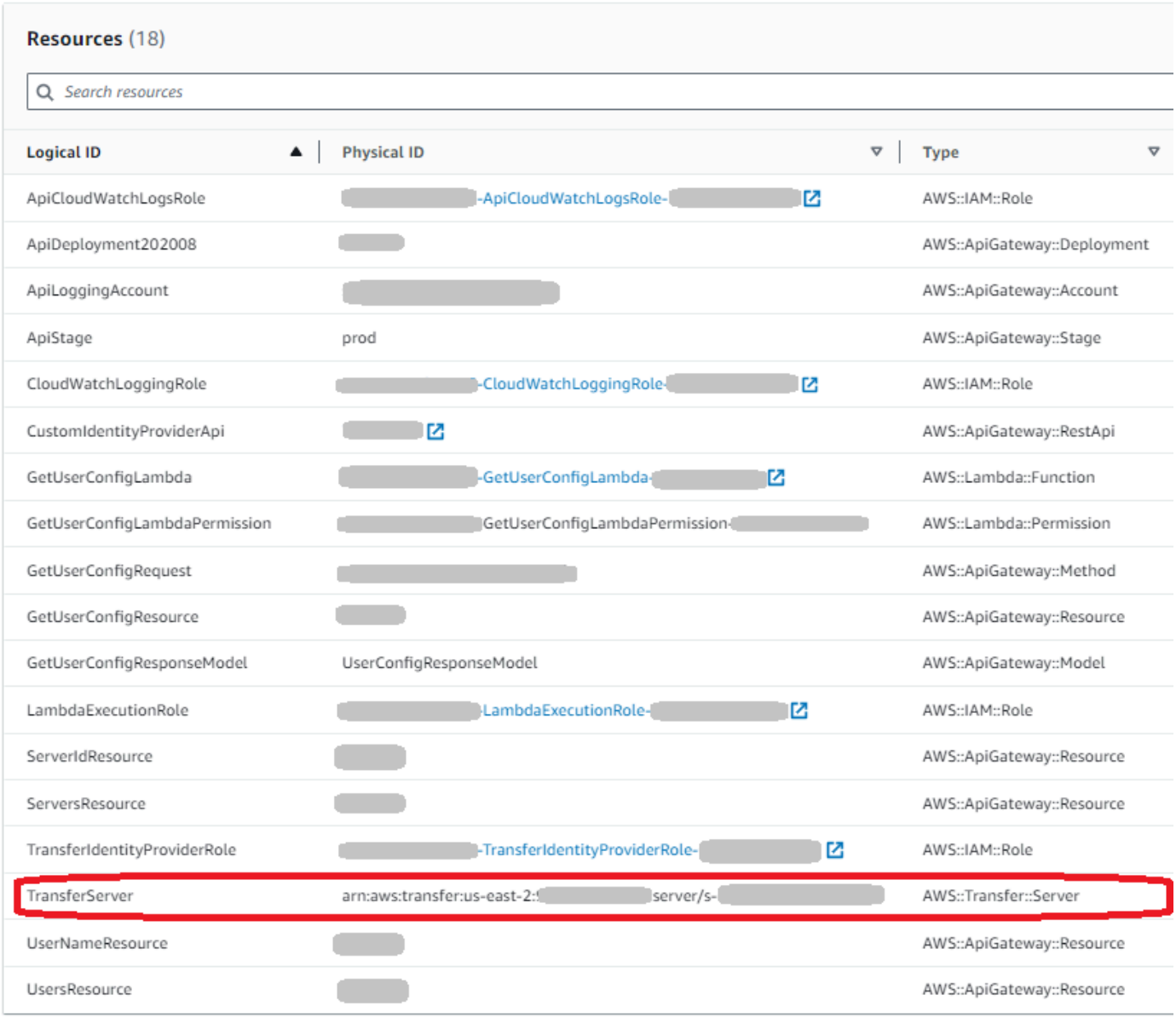
   > **Note**
   >
   > Salin alamat URL Invoke yang muncul di bagian atas halaman. Anda akan membutuhkannya untuk langkah selanjutnya.

Langkah 3: Lihat detail server Transfer Family

Saat Anda menggunakan template untuk membuat AWS CloudFormation tumpukan, server Transfer Family dibuat secara otomatis.

Untuk melihat detail server Transfer Family

1. Buka AWS CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
2. Pilih tumpukan yang Anda buat.
3. Pilih tab Sumber Daya.



Server ARN ditampilkan di kolom Physical ID untuk baris. TransferServer ID server terkandung dalam ARN, misalnya s-11112222333344445.

4. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/, dan di halaman Server, pilih server baru.

   ID server cocok dengan ID yang ditampilkan untuk TransferServersumber daya diAWS CloudFormation.

Langkah 4: Uji apakah pengguna Anda dapat terhubung ke server

Untuk menguji apakah pengguna Anda dapat terhubung ke server, menggunakan konsol Transfer Family

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Pada halaman Server, pilih server baru Anda, pilih Tindakan, lalu pilih Uji.
3. Masukkan teks untuk kredensi login Anda ke dalam bidang Nama Pengguna, dan ke bidang Kata Sandi. Ini adalah nilai yang Anda tetapkan saat Anda menerapkan AWS CloudFormation tumpukan.
4. Untuk Protokol Server, pilih SFTP, dan untuk IP Sumber, masukkan. **`127.0.0.1`**
5. Pilih Uji.

   Jika otentikasi pengguna berhasil, pengujian mengembalikan respons `StatusCode: 200` HTML dan objek JSON yang berisi rincian peran dan izin pengguna. Sebagai contoh:

   ```
   {
       "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"
   }
   ```

   Jika pengujian gagal, tambahkan salah satu kebijakan yang AWS dikelola API Gateway ke peran yang Anda gunakan untuk API Anda.

Langkah 5: Uji koneksi SFTP dan transfer file

Untuk menguji koneksi SFTP

1. Pada perangkat Linux atau macOS, buka terminal perintah.
2. Masukkan salah satu perintah berikut, tergantung pada apakah Anda menggunakan kata sandi atau key pair untuk otentikasi.

   - Jika Anda menggunakan kata sandi, masukkan perintah ini:

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

Jika diminta, masukkan kata sandi Anda.

- Jika Anda menggunakan key pair, masukkan perintah ini:

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

> **Note**
>
> Untuk `sftp` perintah ini, masukkan kode Wilayah AWS tempat server Transfer Family Anda berada. Misalnya, jika server Anda berada di AS Timur (Ohio), masukkan**`us-east-2`**.

3. Pada `sftp>` prompt, pastikan bahwa Anda dapat meng-upload (`put`), download (`get`), dan melihat direktori dan file (`pwd`dan`ls`).

Langkah 6: Batasi akses ke ember

Anda dapat membatasi siapa yang dapat mengakses bucket Amazon S3 tertentu. Contoh berikut menunjukkan setelan yang akan digunakan di CloudFormation tumpukan Anda dan dalam kebijakan yang Anda pilih untuk pengguna Anda.

Dalam contoh ini, kami menetapkan parameter berikut untuk AWS CloudFormation tumpukan:

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

> **Important**
>
> Ini adalah contoh kata sandi. Saat mengonfigurasi metode API Gateway, pastikan Anda memasukkan kata sandi yang kuat.

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

UserPublicKey1 adalah kunci publik yang telah Anda hasilkan sebagai bagian dari public/private key pair.

*role-id*Ini unik untuk peran pengguna yang Anda buat. Kebijakan yang dilampirkan `myuser-api-gateway-role` adalah sebagai berikut:

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

Untuk terhubung ke server menggunakan SFTP, masukkan salah satu perintah berikut pada prompt.

- Jika Anda menggunakan kata sandi untuk mengautentikasi, jalankan perintah berikut:

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

Jika diminta, masukkan kata sandi Anda.

- Jika Anda menggunakan key pair untuk mengautentikasi, jalankan perintah berikut:

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

> **Note**
>
> Untuk `sftp` perintah ini, gunakan ID Wilayah AWS tempat server Transfer Family Anda berada. Misalnya, jika server Anda berada di AS Timur (Ohio), gunakan`us-east-2`.

Pada `sftp` prompt, Anda diarahkan ke direktori home Anda, yang dapat Anda lihat dengan menjalankan pwd perintah. Sebagai contoh:

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

Pengguna tidak dapat melihat direktori apa pun di atas direktori home. Sebagai contoh:

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

Perbarui Lambda jika menggunakan Amazon EFS

Jika Anda memilih Amazon EFS sebagai opsi penyimpanan untuk server Transfer Family, Anda perlu mengedit fungsi lambda untuk tumpukan Anda.

Untuk menambahkan profil posix ke fungsi Lambda Anda

1. Buka konsol Lambda di https://console.aws.amazon.com/lambda/.
2. Pilih fungsi Lambda yang Anda buat sebelumnya. *Fungsi Lambda memiliki format* ***stack-name - GetUserConfigLambda - lambda-identifier, di mana stack-name adalah nama tumpukan dan lambda-identifier*** *adalah pengidentifikasi untuk fungsi tersebut CloudFormation .*

3. Di tab Kode, pilih index.js untuk menampilkan kode untuk fungsi tersebut.
4. Dalam`response`, tambahkan baris berikut antara `Policy` dan`HomeDirectory`:

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

   Dimana nilai *`uid dan nilai gid`* adalah bilangan bulat, 0 atau lebih besar, yang mewakili ID Pengguna dan ID Grup masing-masing.

   Misalnya, setelah Anda menambahkan profil Posix, bidang respons mungkin terlihat seperti berikut:

   ```
   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 '/'
      };
   ```

Menyiapkan konfigurasi AS2

Tutorial ini membahas cara mengatur konfigurasi Applicability Statement 2 (AS2) dengan. AWS Transfer Family Setelah Anda menyelesaikan langkah-langkah yang dijelaskan di sini, Anda akan memiliki server berkemampuan AS2 yang siap menerima pesan AS2 dari mitra dagang sampel. Anda juga akan memiliki konektor yang dapat digunakan untuk mengirim pesan AS2 ke mitra dagang sampel.

> **Note**
>
> Beberapa bagian dari pengaturan contoh menggunakan AWS Command Line Interface (AWS CLI). Jika Anda belum menginstal AWS CLI, lihat Menginstal atau memperbarui versi terbaru dari AWS CLI Panduan AWS Command Line Interface Pengguna.

1. Buat sertifikat untuk diri sendiri dan mitra dagang Anda. Jika Anda memiliki sertifikat yang dapat Anda gunakan, Anda dapat melewati bagian ini.

   Proses ini dijelaskan di Langkah 1: Buat sertifikat untuk AS2.

2. Buat AWS Transfer Family server yang menggunakan protokol AS2. Secara opsional, Anda dapat menambahkan alamat IP Elastis ke server untuk membuatnya menghadap ke internet.

   Proses ini dijelaskan di Langkah 2: Buat server Transfer Family yang menggunakan protokol AS2.

   > **Note**
   >
   > Anda harus membuat server Transfer Family hanya untuk transfer masuk. Jika Anda hanya melakukan transfer keluar, Anda tidak memerlukan server Transfer Family.

3. Impor sertifikat yang Anda buat di langkah 1.

   Proses ini dijelaskan di Langkah 3: Impor sertifikat sebagai sumber sertifikat Transfer Family.

4. Untuk mengatur mitra dagang Anda, buat profil lokal dan profil mitra.

   Proses ini dijelaskan di Langkah 4: Buat profil untuk Anda dan mitra dagang Anda.

5. Buat perjanjian antara Anda dan mitra dagang Anda.

   Proses ini dijelaskan di Langkah 5: Buat kesepakatan antara Anda dan pasangan.

   > **Note**
   >
   > Anda harus membuat perjanjian untuk transfer masuk saja. Jika Anda hanya melakukan transfer keluar, Anda tidak memerlukan perjanjian.

6. Buat konektor antara Anda dan mitra dagang Anda.

   Proses ini dijelaskan di Langkah 6: Buat konektor antara Anda dan pasangan.

   > **Note**
   >
   > Anda harus membuat konektor untuk transfer keluar saja. Jika Anda hanya melakukan transfer masuk, Anda tidak memerlukan konektor.

7. Uji pertukaran file AS2.

   Proses ini dijelaskan di Langkah 7: Uji pertukaran file melalui AS2 dengan menggunakan Transfer Family.

Setelah Anda menyelesaikan langkah-langkah ini, Anda dapat melakukan hal berikut:

- Kirim file ke server mitra berkemampuan AS2 jarak jauh dengan perintah Transfer Family `start-file-transfer` AWS Command Line Interface (AWS CLI).
- Terima file dari server mitra berkemampuan AS2 jarak jauh di port 5080 melalui titik akhir virtual private cloud (VPC) Anda.

Langkah 1: Buat sertifikat untuk AS2

Kedua belah pihak dalam pertukaran AS2 membutuhkan sertifikat X.509. Anda dapat membuat sertifikat ini dengan cara apa pun yang Anda suka. Topik ini menjelaskan cara menggunakan OpenSSL dari baris perintah untuk membuat sertifikat root, dan kemudian menandatangani sertifikat bawahan. Kedua belah pihak harus membuat sertifikat mereka sendiri.

> **Note**
>
> Panjang kunci untuk sertifikat AS2 harus minimal 2048 bit, dan paling banyak 4096.

Untuk mentransfer file dengan mitra, perhatikan hal-hal berikut:

- Anda dapat melampirkan sertifikat ke profil. Sertifikat berisi kunci publik atau pribadi.
- Mitra dagang Anda mengirimi Anda kunci publik mereka, dan Anda mengirimkannya milik Anda.
- Mitra dagang Anda mengenkripsi pesan dengan kunci publik Anda dan menandatanganinya dengan kunci pribadi mereka. Sebaliknya, Anda mengenkripsi pesan dengan kunci publik mitra Anda dan menandatanganinya dengan kunci pribadi Anda.

> **Note**
>
> Jika Anda lebih suka mengelola kunci dengan GUI, Portecleadalah salah satu opsi yang dapat Anda gunakan.

Untuk menghasilkan contoh sertifikat

> **Important**
>
> Jangan kirimkan kunci pribadi Anda kepada pasangan Anda. Dalam contoh ini, Anda membuat satu set kunci publik dan pribadi yang ditandatangani sendiri untuk satu pihak. Jika Anda akan bertindak sebagai mitra dagang untuk tujuan pengujian, Anda dapat mengulangi instruksi ini untuk menghasilkan dua set kunci: satu untuk setiap mitra dagang. Dalam hal ini, Anda tidak perlu menghasilkan dua otoritas sertifikat root (CA).

1. Jalankan perintah berikut untuk menghasilkan kunci pribadi RSA dengan modulus 2048-bit-long.

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

2. Jalankan perintah berikut untuk membuat sertifikat yang ditandatangani sendiri dengan `root-ca-key.pem` file Anda.

```
/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`Argumen terdiri dari nilai-nilai berikut.

	Nama	Penjelasan
`C`	Kode negara	Kode dua huruf untuk negara tempat organisasi Anda berada.
`ST`	Negara bagian, wilayah, atau provinsi	Negara bagian, wilayah, atau provinsi tempat organisasi Anda berada. (Dalam hal ini, wilayah tidak mengacu pada Anda Wilayah AWS.)

	Nama	Penjelasan
L	Nama lokal	Kota tempat organisasi Anda berada.
O	Nama Organisasi	Nama resmi lengkap organisasi Anda, termasuk sufiks, seperti LLC, Corp, dan sebagainya.
OU	Nama unit organisasi	Divisi dalam organisasi Anda yang berhubungan dengan sertifikat ini.
CN	Nama umum atau nama domain yang sepenuhnya memenuhi syarat (FQDN)	Dalam hal ini, kami membuat sertifikat root, jadi nilainya`ROOTCA`. Dalam contoh-contoh ini, kami menggunakan `CN` untuk menggambarkan tujuan sertifikat.

3. Buat kunci penandatanganan dan kunci enkripsi untuk profil lokal Anda.

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

> **Note**
>
> Beberapa server yang mendukung AS2, seperti OpenAS2, mengharuskan Anda menggunakan sertifikat yang sama untuk penandatanganan dan enkripsi. Dalam hal ini, Anda dapat mengimpor kunci pribadi dan sertifikat yang sama untuk kedua tujuan. Untuk melakukannya, jalankan perintah ini alih-alih dua perintah sebelumnya:
>
> ```
> /usr/bin/openssl genrsa -out signing-and-encryption-key.pem 2048
> ```

4. Jalankan perintah berikut untuk membuat Permintaan Penandatanganan Sertifikat (CSR) agar kunci root ditandatangani.

```
/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. Selanjutnya, Anda harus membuat `signing-cert.conf` file dan `encryption-cert.conf` file.

   - Gunakan editor teks untuk membuat `signing-cert.conf` file dengan konten berikut:

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

   - Gunakan editor teks untuk membuat `encryption-cert.conf` file dengan konten berikut:

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

6. Terakhir, Anda membuat sertifikat yang ditandatangani dengan menjalankan perintah berikut.

```
/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
```

Langkah 2: Buat server Transfer Family yang menggunakan protokol AS2

Prosedur ini menjelaskan cara membuat server berkemampuan AS2 dengan menggunakan Transfer Family. AWS CLI

> **Note**
>
> Banyak contoh langkah menggunakan perintah yang memuat parameter dari file. Untuk detail selengkapnya tentang menggunakan file untuk memuat parameter, lihat Cara memuat parameter dari file.

Jika Anda ingin menggunakan konsol sebagai gantinya, lihatMembuat server AS2 menggunakan konsol Transfer Family.

Mirip dengan cara Anda membuat server SFTP atau FTPS, Anda membuat AWS Transfer Family server berkemampuan AS2 dengan menggunakan parameter perintah. `--protocols AS2` `create-server` AWS CLI Saat ini, Transfer Family hanya mendukung tipe titik akhir VPC dan penyimpanan Amazon S3 dengan protokol AS2.

Saat Anda membuat server berkemampuan AS2 untuk Transfer Family dengan menggunakan `create-server` perintah, titik akhir VPC secara otomatis dibuat untuk Anda. Endpoint ini mengekspos port TCP 5080 sehingga dapat menerima pesan AS2.

Jika Anda ingin mengekspos titik akhir VPC Anda secara publik ke internet, Anda dapat mengaitkan alamat IP Elastis dengan titik akhir VPC Anda.

Untuk menggunakan petunjuk ini, Anda memerlukan yang berikut:

- ID VPC Anda (misalnya, vpc-abcdef01).
- ID subnet VPC Anda (misalnya, subnet-abcdef01, 01, subnet-021345ab). subnet-subnet-abcdef
- Satu atau lebih ID grup keamanan yang memungkinkan lalu lintas masuk pada port TCP 5080 dari mitra dagang Anda (misalnya, sg-1234567890abcdef0 dan sg-abcdef01234567890).
- (Opsional) Alamat IP Elastis yang ingin Anda kaitkan dengan titik akhir VPC Anda.
- Jika mitra dagang Anda tidak terhubung ke VPC Anda melalui VPN, Anda memerlukan gateway internet. Untuk informasi selengkapnya, lihat Hubungkan ke internet menggunakan gateway internet di Panduan Pengguna Amazon VPC.

Untuk membuat server berkemampuan AS2

1. Jalankan perintah berikut. Ganti masing-masing *`user input placeholder`* dengan informasi Anda sendiri.

```
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. (Opsional) Anda dapat membuat titik akhir VPC menjadi publik. Anda dapat melampirkan alamat IP Elastis ke server Transfer Family hanya melalui `update-server` operasi. Perintah berikut menghentikan server, memperbaruinya dengan alamat IP Elastis, dan kemudian mulai lagi.

```
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
```

`start-server`Perintah ini secara otomatis membuat catatan DNS untuk Anda yang berisi alamat IP publik untuk server Anda. Untuk memberi mitra dagang Anda akses ke server, Anda memberi mereka informasi berikut. Dalam hal ini, *`your-region`* mengacu pada Anda Wilayah AWS.

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

URL lengkap yang Anda berikan kepada mitra dagang Anda adalah sebagai berikut:

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

3. Untuk menguji apakah server AS2 Anda dapat diakses, gunakan perintah berikut. Pastikan server Anda dapat diakses baik melalui alamat DNS pribadi titik akhir VPC Anda, atau melalui titik akhir publik Anda (jika Anda mengaitkan alamat IP Elastis dengan titik akhir Anda).

   Jika server Anda dikonfigurasi dengan benar, koneksi akan berhasil. Namun, Anda akan menerima respons kode status HTTP 400 (Permintaan Buruk) karena Anda tidak mengirim pesan AS2 yang valid.

- Untuk titik akhir publik (jika Anda mengaitkan alamat IP Elastis pada langkah sebelumnya), jalankan perintah berikut, ganti ID server dan Wilayah Anda.

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

- Jika Anda terhubung dalam VPC Anda, cari nama DNS pribadi titik akhir VPC Anda dengan menjalankan perintah berikut.

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

`describe-server`Perintah ini mengembalikan ID titik akhir VPC Anda dalam parameter. `VpcEndpointId` Gunakan nilai ini untuk menjalankan perintah berikut.

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

`describe-vpc-endpoints`Perintah ini mengembalikan `DNSEntries` array, dengan beberapa `DnsName` parameter. Gunakan nama DNS Regional (yang tidak menyertakan Availability Zone) dalam perintah berikut.

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

Misalnya, perintah berikut menunjukkan nilai sampel untuk placeholder di perintah sebelumnya.

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

4. (Opsional) Konfigurasikan peran logging. Transfer Family mencatat status pesan yang dikirim dan diterima dalam format JSON terstruktur ke CloudWatch log Amazon. Untuk memberi Transfer Family akses ke CloudWatch log di akun Anda, Anda harus mengonfigurasi peran pencatatan di server Anda.

   Buat peran AWS Identity and Access Management (IAM) yang dipercaya`transfer.amazonaws.com`, dan lampirkan kebijakan `AWSTransferLoggingAccess` terkelola. Untuk detailnya, lihat Buat peran dan kebijakan IAM.

Perhatikan Nama Sumber Daya Amazon (ARN) dari peran IAM yang baru saja Anda buat, dan kaitkan dengan server dengan menjalankan perintah berikut: `update-server`

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

> **Note**
>
> Meskipun peran logging bersifat opsional, kami sangat menyarankan untuk mengaturnya sehingga Anda dapat melihat status pesan Anda dan memecahkan masalah konfigurasi.

Langkah 3: Impor sertifikat sebagai sumber sertifikat Transfer Family

Prosedur ini menjelaskan cara mengimpor sertifikat dengan menggunakan AWS CLI. Jika Anda ingin menggunakan konsol Transfer Family sebagai gantinya, lihatthe section called “Impor sertifikat AS2”.

Untuk mengimpor sertifikat penandatanganan dan enkripsi yang Anda buat di langkah 1, jalankan `import-certificate` perintah berikut. Jika Anda menggunakan sertifikat yang sama untuk enkripsi dan penandatanganan, impor sertifikat yang sama dua kali (sekali dengan `SIGNING` penggunaan dan sekali lagi dengan `ENCRYPTION` penggunaan).

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

Perintah ini mengembalikan penandatanganan Anda`CertificateId`. Pada bagian selanjutnya, ID sertifikat ini disebut sebagai*my-signing-cert-id*.

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

Perintah ini mengembalikan enkripsi Anda`CertificateId`. Pada bagian selanjutnya, ID sertifikat ini disebut sebagai*my-encrypt-cert-id*.

Selanjutnya, impor enkripsi mitra Anda dan tandatangani sertifikat dengan menjalankan perintah berikut.

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

Perintah ini mengembalikan enkripsi pasangan Anda`CertificateId`. Pada bagian selanjutnya, ID sertifikat ini disebut sebagai*partner-encrypt-cert-id*.

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

Perintah ini mengembalikan penandatanganan pasangan Anda`CertificateId`. Pada bagian selanjutnya, ID sertifikat ini disebut sebagai*partner-signing-cert-id*.

Langkah 4: Buat profil untuk Anda dan mitra dagang Anda

Prosedur ini menjelaskan cara membuat profil AS2 dengan menggunakan AWS CLI. Jika Anda ingin menggunakan konsol Transfer Family sebagai gantinya, lihatthe section called “Buat profil AS2”.

Buat profil AS2 lokal Anda dengan menjalankan perintah berikut. Perintah ini mereferensikan sertifikat yang berisi kunci publik dan pribadi Anda.

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

Perintah ini mengembalikan ID profil Anda. Pada bagian selanjutnya, ID ini disebut sebagai*my-profile-id*.

Sekarang buat profil mitra dengan menjalankan perintah berikut. Perintah ini hanya menggunakan sertifikat kunci publik mitra Anda. Untuk menggunakan perintah ini, ganti *user input placeholders* dengan informasi Anda sendiri; misalnya, nama AS2 mitra Anda dan ID sertifikat.

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

Perintah ini mengembalikan ID profil mitra Anda. Pada bagian selanjutnya, ID ini disebut sebagai*partner-profile-id*.

> **Note**
>
> Pada perintah sebelumnya, ganti *MYCORP* dengan nama organisasi Anda, dan *PARTNER-COMPANY dengan nama organisasi mitra* dagang Anda.

Langkah 5: Buat kesepakatan antara Anda dan pasangan

Prosedur ini menjelaskan cara membuat perjanjian AS2 dengan menggunakan. AWS CLI Jika Anda ingin menggunakan konsol Transfer Family sebagai gantinya, lihatthe section called “Buat perjanjian AS2”.

Perjanjian menyatukan dua profil (lokal dan mitra), sertifikat mereka, dan konfigurasi server yang memungkinkan transfer AS2 masuk antara dua pihak. Anda dapat membuat daftar item Anda dengan menjalankan perintah berikut.

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

Langkah ini memerlukan bucket Amazon S3 dan peran IAM dengan akses baca/tulis ke dan dari bucket. Instruksi untuk membuat peran ini sama dengan protokol Transfer Family SFTP, FTP, dan FTPS dan tersedia di. Buat peran dan kebijakan IAM

Untuk membuat perjanjian, Anda memerlukan item berikut:

- Nama bucket Amazon S3 (dan awalan objek, jika ditentukan)
- ARN dari peran IAM dengan akses ke bucket
- ID server Transfer Family Anda
- ID profil Anda dan ID profil mitra Anda

Buat perjanjian dengan menjalankan perintah berikut.

```
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
```

Jika berhasil, perintah ini mengembalikan ID untuk perjanjian. Anda kemudian dapat melihat detail perjanjian dengan perintah berikut.

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

Langkah 6: Buat konektor antara Anda dan pasangan

Prosedur ini menjelaskan cara membuat konektor AS2 dengan menggunakan. AWS CLI Jika Anda ingin menggunakan konsol Transfer Family sebagai gantinya, lihatthe section called “Konfigurasikan konektor AS2”.

Anda dapat menggunakan operasi `StartFileTransfer` API untuk mengirim file yang disimpan di Amazon S3 ke titik akhir AS2 mitra dagang Anda dengan menggunakan konektor. Anda dapat menemukan profil yang Anda buat sebelumnya dengan menjalankan perintah berikut.

```
aws transfer list-profiles
```

Saat Anda membuat konektor, Anda harus memberikan URL server AS2 mitra Anda. Salin teks berikut ke file bernama`testAS2Config.json`.

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

> **Note**
>
> Untuk`EncryptionAlgorithm`, jangan tentukan `DES_EDE3_CBC` algoritme kecuali Anda harus mendukung klien lama yang membutuhkannya, karena ini adalah algoritma enkripsi yang lemah.

Kemudian jalankan perintah berikut untuk membuat konektor.

```
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
```

Langkah 7: Uji pertukaran file melalui AS2 dengan menggunakan Transfer Family

Menerima file dari mitra dagang Anda

Jika Anda mengaitkan alamat IP Elastis publik dengan titik akhir VPC Anda, Transfer Family secara otomatis membuat nama DNS yang berisi alamat IP publik Anda. Subdomain adalah ID AWS Transfer Family server Anda (dari formats-1234567890abcdef0). Berikan URL server Anda kepada mitra dagang Anda dalam format berikut.

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

Jika Anda tidak mengaitkan alamat IP Elastis publik dengan titik akhir VPC Anda, cari nama host dari titik akhir VPC yang dapat menerima pesan AS2 melalui HTTP POST dari mitra dagang Anda di port 5080. Untuk mengambil detail titik akhir VPC, gunakan perintah berikut.

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

Misalnya, asumsikan perintah sebelumnya mengembalikan ID titik akhir VPC dari. vpce-1234abcd5678efghi Kemudian, Anda akan menggunakan perintah berikut untuk mengambil nama DNS.

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

Perintah ini mengembalikan semua detail untuk titik akhir VPC yang Anda butuhkan untuk menjalankan perintah berikut.

Nama DNS tercantum dalam DnsEntries array. Mitra dagang Anda harus berada dalam VPC Anda untuk mengakses titik akhir VPC Anda (misalnya melalui AWS PrivateLink atau VPN). Berikan URL titik akhir VPC Anda kepada mitra Anda dalam format berikut.

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

Misalnya, URL berikut menunjukkan nilai sampel untuk placeholder di perintah sebelumnya.

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

Dalam contoh ini, transfer yang berhasil disimpan di lokasi yang ditentukan dalam `base-directory` parameter yang Anda tentukanLangkah 5: Buat kesepakatan antara Anda dan pasangan. Jika kami berhasil menerima file bernama `myfile1.txt` dan`myfile2.txt`, file disimpan sebagai/*`path-defined-in-the-agreement`*/`processed/`*`original_filename.messageId.original_extension`*. Di sini, file disimpan sebagai /*`DOC-EXAMPLE-DESTINATION-BUCKET`*/`AS2-inbox/processed/myfile1.`*`messageId`*`.txt` dan/*`DOC-EXAMPLE-DESTINATION-BUCKET`*/`AS2-inbox/processed/myfile2.`*`messageId`*`.txt`.

Jika Anda mengonfigurasi peran logging saat membuat server Transfer Family, Anda juga dapat memeriksa CloudWatch log untuk status pesan AS2.

Kirim file ke mitra dagang Anda

Anda dapat menggunakan Transfer Family untuk mengirim pesan AS2 dengan mereferensikan ID konektor dan jalur ke file, seperti yang diilustrasikan dalam perintah `start-file-transfer` AWS Command Line Interface ()AWS CLI berikut:

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

Untuk mendapatkan detail konektor Anda, jalankan perintah berikut:

```
aws transfer list-connectors
```

`list-connectors`Perintah mengembalikan ID konektor, URL, dan Nama Sumber Daya Amazon (ARN) untuk konektor Anda.

Untuk mengembalikan properti konektor tertentu, jalankan perintah berikut dengan ID yang ingin Anda gunakan:

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

`describe-connector`Perintah mengembalikan semua properti untuk konektor, termasuk URL, peran, profil, Pemberitahuan Disposisi Pesan (mDNS), tag, dan metrik pemantauan.

Anda dapat mengonfirmasi bahwa mitra berhasil menerima file dengan melihat file JSON dan MDN. File-file ini diberi nama sesuai dengan konvensi yang dijelaskan dalamNama dan lokasi file. Jika Anda mengonfigurasi peran logging saat membuat konektor, Anda juga dapat memeriksa CloudWatch log Anda untuk status pesan AS2.

Mengkonfigurasi titik akhir server SFTP, FTPS, atau FTP

Topik ini memberikan detail untuk membuat dan menggunakan titik akhir AWS Transfer Family server yang menggunakan satu atau lebih protokol SFTP, FTPS, dan FTP.

Topik

- Opsi penyedia identitas
- AWS Transfer Family matriks tipe titik akhir
- Mengkonfigurasi titik akhir server SFTP, FTPS, atau FTP
- Mentransfer file melalui titik akhir server menggunakan klien
- Mengelola pengguna untuk titik akhir server
- Menggunakan direktori logis untuk menyederhanakan struktur direktori Transfer Family Anda

Opsi penyedia identitas

AWS Transfer Family menyediakan beberapa metode untuk mengautentikasi dan mengelola pengguna. Tabel berikut membandingkan penyedia identitas yang tersedia yang dapat Anda gunakan dengan Transfer Family.

Tindakan	AWS Transfer Family layanan dikelola	AWS Managed Microsoft AD	Amazon API Gateway	AWS Lambda
Protokol yang didukung	SFTP	SFTP, FTPS, FTP	SFTP, FTPS, FTP	SFTP, FTPS, FTP
Otentikasi berbasis kunci	Ya	Tidak	Ya	Ya
Autentikasi kata sandi	Tidak	Ya	Ya	Ya
AWS Identity and Access Managemen	Ya	Ya	Ya	Ya

Tindakan	AWS Transfer Family layanan dikelola	AWS Managed Microsoft AD	Amazon API Gateway	AWS Lambda
t (IAM) dan POSIX				
Direktori home logis	Ya	Ya	Ya	Ya
Akses parameter (berbasis nama pengguna)	Ya	Ya	Ya	Ya
Struktur akses ad hoc	Ya	Tidak	Ya	Ya
AWS WAF	Tidak	Tidak	Ya	Tidak

Catatan:

- IAM digunakan untuk mengontrol akses untuk penyimpanan dukungan Amazon S3, dan POSIX digunakan untuk Amazon EFS.
- Ad hoc mengacu pada kemampuan untuk mengirim profil pengguna saat runtime. Misalnya, Anda dapat mendaratkan pengguna di direktori home mereka dengan meneruskan nama pengguna sebagai variabel.
- Untuk detailnya AWS WAF, lihatTambahkan firewall aplikasi web.
- Ada posting blog yang menjelaskan penggunaan fungsi Lambda yang terintegrasi dengan Microsoft Azure AD sebagai penyedia identitas Transfer Family Anda. Untuk detailnya, lihat Mengautentikasi AWS Transfer Family dengan Azure Active Directory dan. AWS Lambda
- Kami menyediakan beberapa AWS CloudFormation template untuk membantu Anda dengan cepat menyebarkan server Transfer Family yang menggunakan penyedia identitas khusus. Untuk detailnya, lihat Template fungsi Lambda.

Dalam prosedur berikut, Anda dapat membuat server berkemampuan SFTP, server berkemampuan FTPS, server berkemampuan FTP, atau server yang mendukung AS2.

Langkah selanjutnya

- Buat server berkemampuan SFTP
- Buat server berkemampuan FTPS
- Buat server berkemampuan FTP
- Mengkonfigurasi AS2

AWS Transfer Family matriks tipe titik akhir

Saat membuat server Transfer Family, Anda memilih jenis endpoint yang akan digunakan. Tabel berikut menjelaskan karakteristik untuk setiap jenis titik akhir.

Matriks tipe titik akhir

Karakteristik	Publik	VPC - Internet	VPC - Internal	VPC_Endpoint (usang)
Protokol yang didukung	SFTP	SFTP, FTPS, AS2	SFTP, FTP, FTPS, AS2	SFTP
Akses	Dari internet. Jenis titik akhir ini tidak memerlukan konfigurasi khusus apa pun di VPC Anda.	Melalui internet dan dari dalam lingkungan yang terhubung dengan VPC dan VPC, seperti pusat data lokal di atas atau VPN. AWS Direct Connect	Dari dalam lingkungan yang terhubung dengan VPC dan VPC, seperti pusat data lokal di atas atau VPN. AWS Direct Connect	Dari dalam lingkungan yang terhubung dengan VPC dan VPC, seperti pusat data lokal di atas atau VPN. AWS Direct Connect
Alamat IP statis	Anda tidak dapat melampirkan alamat IP statis. AWS menyediakan alamat IP yang dapat berubah.	Anda dapat melampirkan alamat IP Elastis ke titik akhir. Ini bisa berupa alamat AWS IP milik atau alamat	Alamat IP pribadi yang dilampirkan ke titik akhir tidak berubah.	Alamat IP pribadi yang dilampirkan ke titik akhir tidak berubah.

Karakteristik	Publik	VPC - Internet	VPC - Internal	VPC_Endpoint (usang)
		IP Anda sendiri (Bawa alamat IP Anda sendiri). Alamat IP elastis yang dilampirkan ke titik akhir tidak berubah. Alamat IP pribadi yang dilampirkan ke server juga tidak berubah.		

Karakteristik	Publik	VPC - Internet	VPC - Internal	VPC_Endpoint (usang)
Daftar izin IP sumber	Jenis titik akhir ini tidak mendukung daftar izin berdasarkan alamat IP sumber. Titik akhir dapat diakses publik dan mendengarkan lalu lintas melalui port 22. **Note** Untuk titik akhir yang dihosting VPC, server SFTP Transfer Family dapat beroperasi melalui port 22 (default) atau port 2222.	Untuk mengizinkan akses berdasarkan alamat IP sumber, Anda dapat menggunakan grup keamanan yang dilampirkan ke titik akhir server dan ACL jaringan yang dilampirkan ke subnet tempat titik akhir berada.	Untuk mengizinkan akses berdasarkan alamat IP sumber, Anda dapat menggunakan grup keamanan yang dilampirkan ke titik akhir server dan daftar kontrol akses jaringan (ACL jaringan) yang dilampirkan ke subnet tempat titik akhir berada.	Untuk mengizinkan akses berdasarkan alamat IP sumber, Anda dapat menggunakan grup keamanan yang dilampirkan ke titik akhir server dan ACL jaringan yang dilampirkan ke subnet tempat titik akhir berada.

Karakteristik	Publik	VPC - Internet	VPC - Internal	VPC_Endpoint (usang)
Daftar izin firewall klien	Anda harus mengizinkan nama DNS server. Karena alamat IP dapat berubah, hindari menggunak an alamat IP untuk daftar izin firewall klien Anda.	Anda dapat mengizinkan nama DNS server atau alamat IP Elastis yang dilampirkan ke server.	Anda dapat mengizinkan alamat IP pribadi atau nama DNS dari titik akhir.	Anda dapat mengizinkan alamat IP pribadi atau nama DNS dari titik akhir.

> **Note**
>
> Jenis `VPC_ENDPOINT` endpoint sekarang sudah usang dan tidak dapat digunakan untuk membuat server baru. Alih-alih menggunakan`EndpointType=VPC_ENDPOINT`, gunakan tipe titik akhir VPC baru (`EndpointType=VPC`), yang dapat Anda gunakan sebagai Internal atau Internet Facing, seperti yang dijelaskan dalam tabel sebelumnya. Untuk detailnya, lihat Menghentikan penggunaan VPC_ENDPOINT.

Pertimbangkan opsi berikut untuk meningkatkan postur keamanan AWS Transfer Family server Anda:

- Gunakan titik akhir VPC dengan akses internal, sehingga server hanya dapat diakses oleh klien dalam lingkungan yang terhubung dengan VPC atau VPC Anda seperti pusat data lokal di atas atau VPN. AWS Direct Connect
- Untuk memungkinkan klien mengakses titik akhir melalui internet dan melindungi server Anda, gunakan titik akhir VPC dengan akses yang menghadap ke internet. Kemudian, ubah grup keamanan VPC untuk mengizinkan lalu lintas hanya dari alamat IP tertentu yang meng-host klien pengguna Anda.

- Jika Anda memerlukan otentikasi berbasis kata sandi dan Anda menggunakan penyedia identitas khusus dengan server Anda, itu adalah praktik terbaik bahwa kebijakan kata sandi Anda mencegah pengguna membuat kata sandi yang lemah dan membatasi jumlah upaya login yang gagal.
- AWS Transfer Family adalah layanan terkelola, sehingga tidak menyediakan akses shell. Anda tidak dapat langsung mengakses server SFTP yang mendasarinya untuk menjalankan perintah asli OS di server Transfer Family.
- Gunakan Network Load Balancer di depan titik akhir VPC dengan akses internal. Ubah port listener pada penyeimbang beban dari port 22 ke port yang berbeda. Ini dapat mengurangi, tetapi tidak menghilangkan, risiko pemindai port dan bot yang menyelidiki server Anda, karena port 22 paling sering digunakan untuk pemindaian. Untuk detailnya, lihat posting blog Network Load Balancers sekarang mendukung grup Keamanan.

> **Note**
>
> Jika Anda menggunakan Network Load Balancer, AWS Transfer Family CloudWatch log menampilkan alamat IP untuk NLB, bukan alamat IP klien yang sebenarnya.

Mengkonfigurasi titik akhir server SFTP, FTPS, atau FTP

Anda dapat membuat server transfer file dengan menggunakan AWS Transfer Family layanan ini. Protokol transfer file berikut tersedia:

- Secure Shell (SSH) File Transfer Protocol (SFTP) — Transfer file melalui SSH. Untuk detailnya, lihat the section called “Buat server berkemampuan SFTP”.

> **Note**
>
> Kami memberikan AWS CDK contoh untuk membuat server SFTP Transfer Family. Contoh menggunakan TypeScript, dan tersedia di GitHub sini.

- File Transfer Protocol Secure (FTPS) — Transfer file dengan enkripsi TLS. Untuk detailnya, lihat the section called “Buat server berkemampuan FTPS”.
- Protokol Transfer File (FTP) — Transfer file tidak terenkripsi. Untuk detailnya, lihat the section called “Buat server berkemampuan FTP”.
- Pernyataan Penerapan 2 (AS2) — Transfer file untuk mengangkut data terstruktur. business-to-business Untuk detailnya, lihat the section called “Konfigurasikan AS2”. Untuk AS2, Anda dapat

dengan cepat membuat AWS CloudFormation tumpukan untuk tujuan demonstrasi. Prosedur ini dijelaskan dalamGunakan template untuk membuat demo Transfer Family AS2 stack.

Anda dapat membuat server dengan beberapa protokol.

> **Note**
>
> Jika Anda memiliki beberapa protokol yang diaktifkan untuk titik akhir server yang sama dan Anda ingin memberikan akses dengan menggunakan nama pengguna yang sama melalui beberapa protokol, Anda dapat melakukannya selama kredensyal khusus untuk protokol telah diatur di penyedia identitas Anda. Untuk FTP, kami sarankan untuk mempertahankan kredensyal terpisah dari SFTP dan FTPS. Ini karena, tidak seperti SFTP dan FTPS, FTP mentransmisikan kredensyal dalam teks yang jelas. Dengan mengisolasi kredensyal FTP dari SFTP atau FTPS, jika kredensyal FTP dibagikan atau diekspos, beban kerja Anda menggunakan SFTP atau FTPS tetap aman.

Saat Anda membuat server, Anda memilih spesifik Wilayah AWS untuk melakukan permintaan operasi file pengguna yang ditugaskan ke server tersebut. Seiring dengan menetapkan server satu atau beberapa protokol, Anda juga menetapkan salah satu jenis penyedia identitas berikut:

- Layanan dikelola dengan menggunakan kunci SSH. Untuk detailnya, lihat Bekerja dengan pengguna yang dikelola layanan.
- AWS Directory Service for Microsoft Active Directory (AWS Managed Microsoft AD). Metode ini memungkinkan Anda mengintegrasikan grup Microsoft Active Directory untuk menyediakan akses ke server Transfer Family Anda. Untuk detailnya, lihat Menggunakan penyedia identitas AWS Directory Service.
- Metode khusus. Metode penyedia identitas kustom menggunakan AWS Lambda atau Amazon API Gateway dan memungkinkan Anda mengintegrasikan layanan direktori untuk mengautentikasi dan mengotorisasi pengguna Anda. Layanan secara otomatis menetapkan pengenal yang secara unik mengidentifikasi server Anda. Untuk detailnya, lihat Bekerja dengan penyedia identitas khusus. Transfer Family menyediakan AWS CloudFormation template yang dapat Anda gunakan untuk menyebarkan server dengan cepat yang menggunakan penyedia identitas khusus.
  - Fungsi Lambda untuk otentikasimenjelaskan CloudFormation template yang menggunakan fungsi Lambda untuk otentikasi.

- Mengautentikasi menggunakan metode API Gatewaymenjelaskan CloudFormation template yang menggunakan metode Amazon API Gateway untuk autentikasi.

Anda juga menetapkan server jenis titik akhir (dapat diakses publik atau dihosting VPC) dan nama host dengan menggunakan titik akhir server default, atau nama host khusus dengan menggunakan layanan Amazon Route 53 atau dengan menggunakan layanan Sistem Nama Domain (DNS) pilihan Anda. Nama host server harus unik di Wilayah AWS tempat pembuatannya.

Selain itu, Anda dapat menetapkan peran CloudWatch pencatatan Amazon untuk mendorong peristiwa ke CloudWatch log Anda, memilih kebijakan keamanan yang berisi algoritme kriptografi yang diaktifkan untuk digunakan oleh server Anda, dan menambahkan metadata ke server dalam bentuk tag yang merupakan pasangan nilai kunci.

> **Important**
>
> Anda dikenakan biaya untuk server instantiated dan untuk transfer data. Untuk informasi tentang harga dan penggunaan AWS Pricing Calculator untuk mendapatkan perkiraan biaya penggunaan Transfer Family, lihat AWS Transfer Family harga.

Buat server berkemampuan SFTP

Secure Shell (SSH) File Transfer Protocol (SFTP) adalah protokol jaringan yang digunakan untuk transfer data yang aman melalui internet. Protokol ini mendukung fungsionalitas keamanan dan otentikasi penuh SSH. Ini banyak digunakan untuk bertukar data, termasuk informasi sensitif antara mitra bisnis di berbagai industri seperti layanan keuangan, perawatan kesehatan, ritel, dan periklanan.

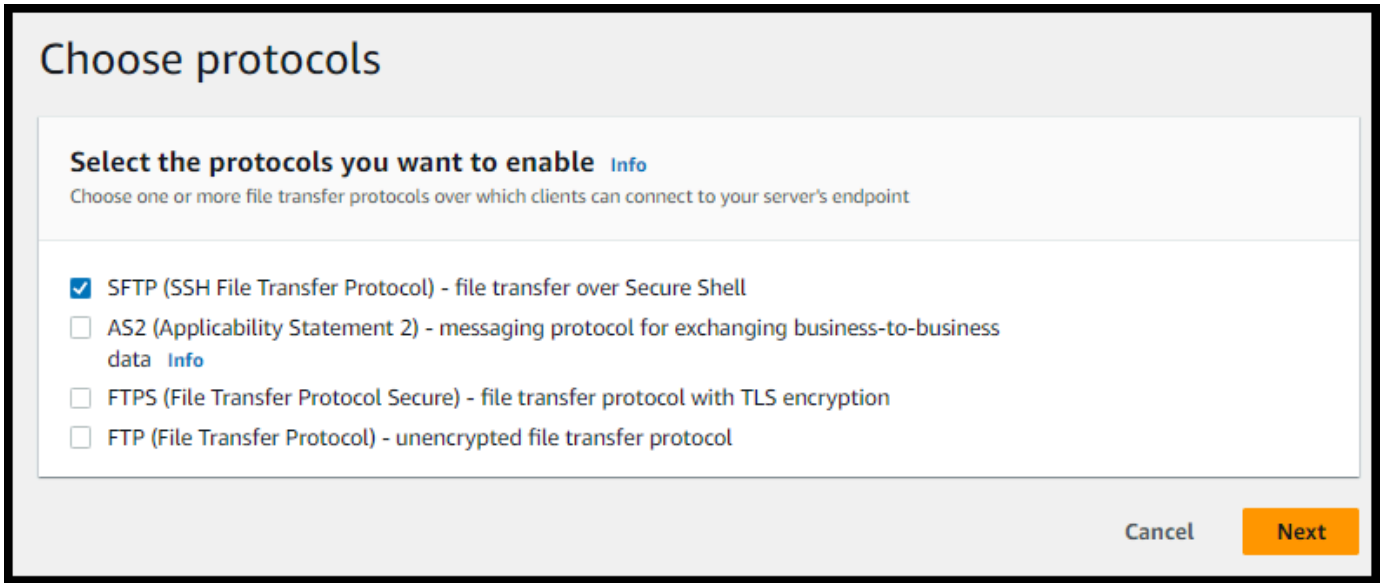
> **Note**
>
> Server SFTP untuk Transfer Family beroperasi melalui port 22. Untuk titik akhir yang dihosting VPC, server SFTP Transfer Family juga dapat beroperasi melalui port 2222. Untuk rincian selengkapnya, lihat Buat server di cloud pribadi virtual.

Lihat juga

- Kami memberikan AWS CDK contoh untuk membuat server SFTP Transfer Family. Contoh menggunakan TypeScript, dan tersedia di GitHub sini.
- Untuk panduan tentang cara menerapkan server Transfer Family di dalam VPC, lihat Gunakan daftar izin IP untuk mengamankan server Anda. AWS Transfer Family

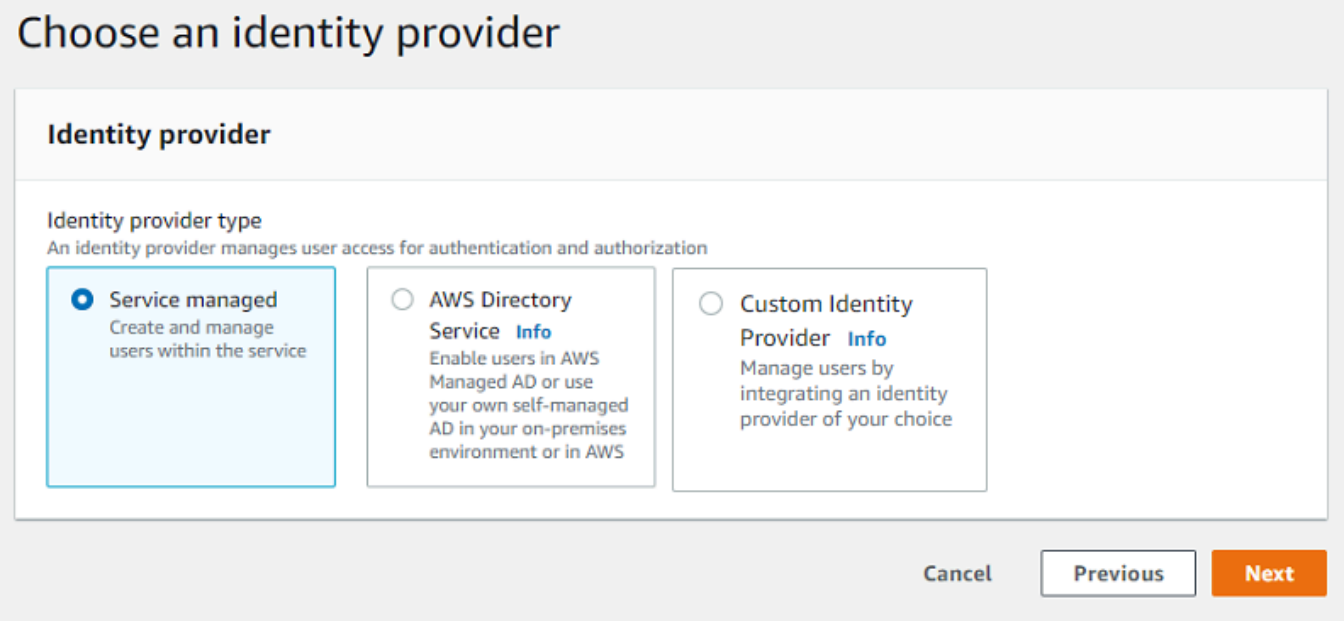
Untuk membuat server berkemampuan SFTP

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/ dan pilih Server dari panel navigasi, lalu pilih Buat server.
2. Di Pilih protokol, pilih SFTP, lalu pilih Berikutnya.



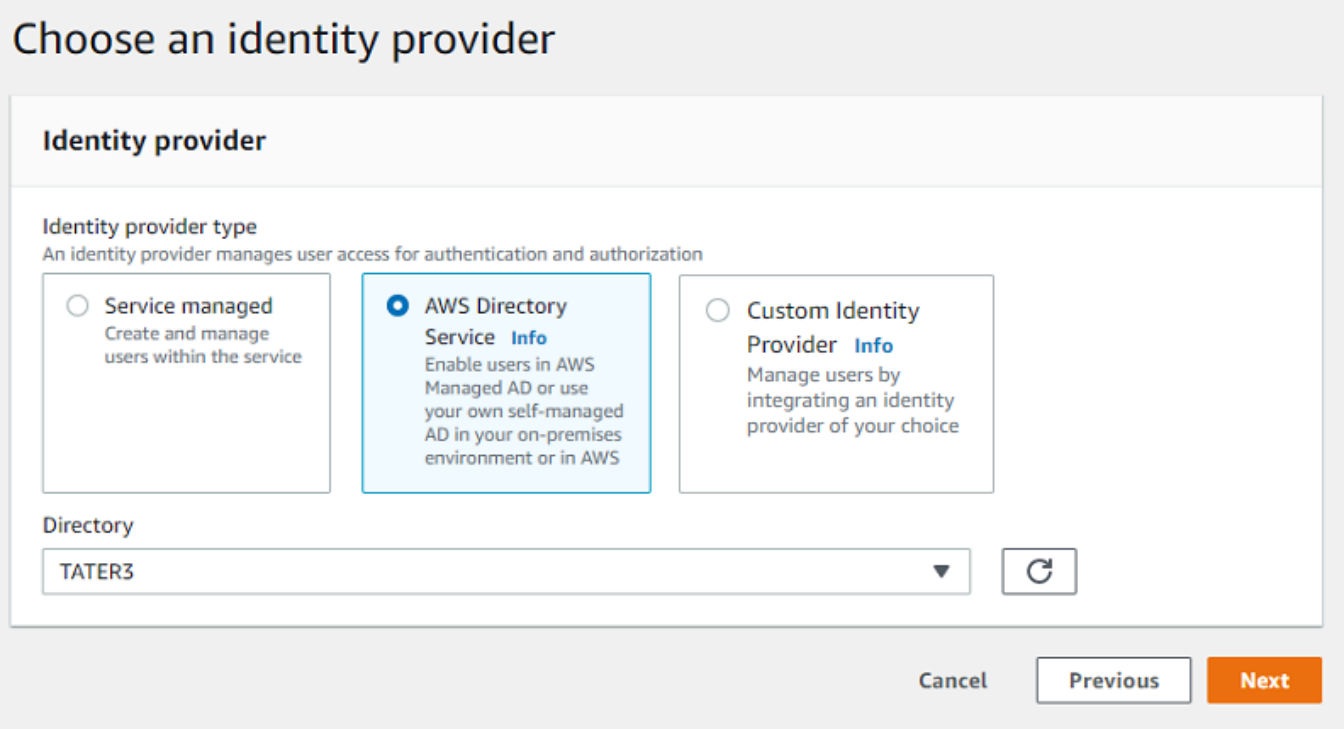
3. Di Pilih penyedia identitas, pilih penyedia identitas yang ingin Anda gunakan untuk mengelola akses pengguna. Anda memiliki opsi berikut:

   - Layanan dikelola - Anda menyimpan identitas dan kunci pengguna. AWS Transfer Family



- AWS Directory Service for Microsoft Active Directory— Anda menyediakan AWS Directory Service direktori untuk mengakses titik akhir. Dengan demikian, Anda dapat menggunakan kredensyal yang disimpan di Active Directory untuk mengautentikasi pengguna Anda. Untuk mempelajari lebih lanjut tentang bekerja dengan penyedia AWS Managed Microsoft AD identitas, lihatMenggunakan penyedia identitas AWS Directory Service.

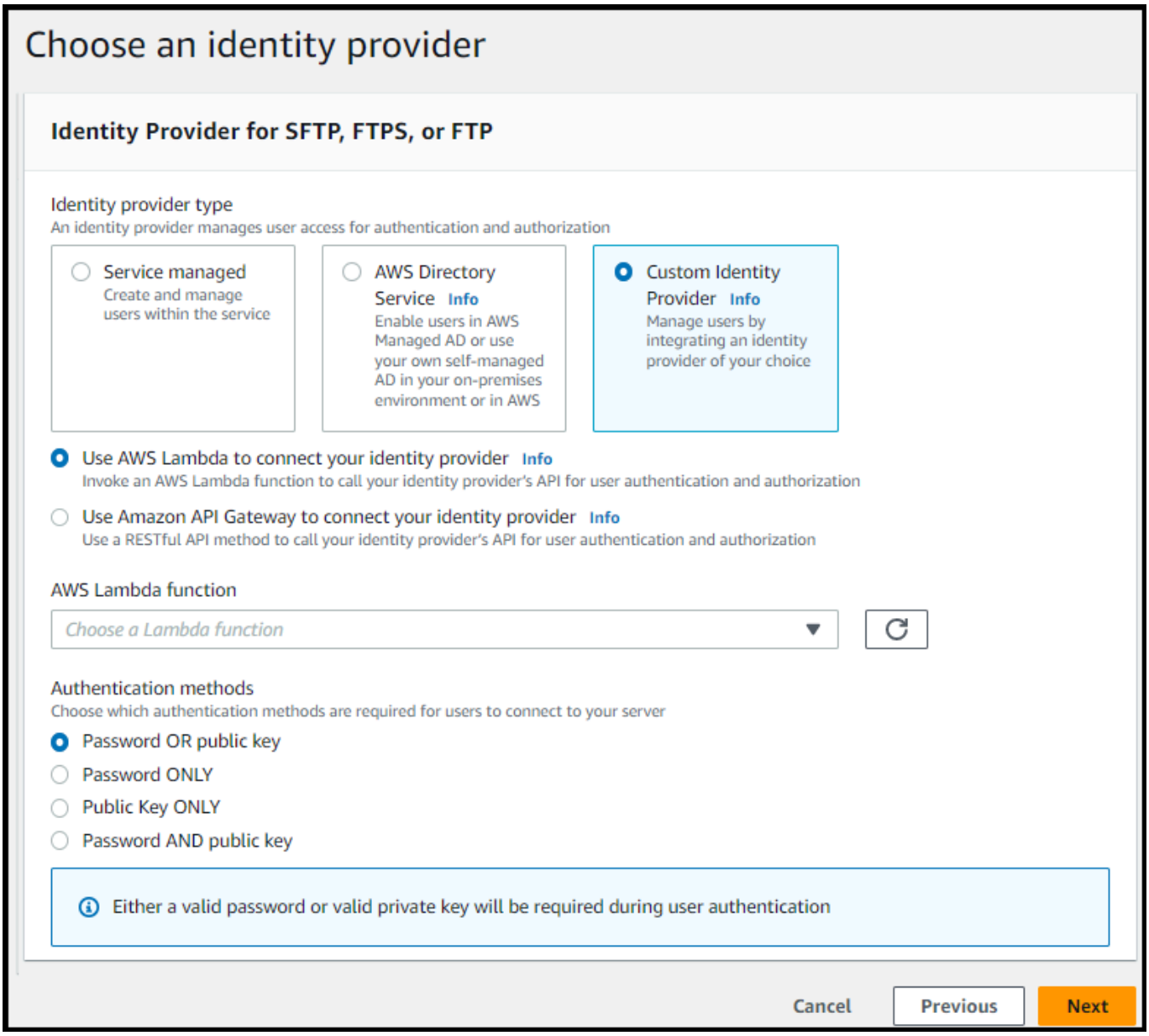
> **Note**
>
> - Direktori Cross-Account dan Shared tidak didukung untuk. AWS Managed Microsoft AD
> - Untuk menyiapkan server dengan Directory Service sebagai penyedia identitas Anda, Anda perlu menambahkan beberapa AWS Directory Service izin. Untuk detailnya, lihat Sebelum Anda mulai menggunakan AWS Directory Service for Microsoft Active Directory.



- Penyedia identitas khusus - Pilih salah satu opsi berikut:
  - Gunakan AWS Lambda untuk menghubungkan penyedia identitas Anda — Anda dapat menggunakan penyedia identitas yang ada, didukung oleh fungsi Lambda. Anda memberikan nama fungsi Lambda. Untuk informasi selengkapnya, lihat Menggunakan AWS Lambda untuk mengintegrasikan penyedia identitas Anda.
  - Gunakan Amazon API Gateway untuk menghubungkan penyedia identitas Anda — Anda dapat membuat metode API Gateway yang didukung oleh fungsi Lambda untuk digunakan sebagai penyedia identitas. Anda menyediakan URL Amazon API Gateway dan peran pemanggilan. Untuk informasi selengkapnya, lihat Menggunakan Amazon API Gateway untuk mengintegrasikan penyedia identitas Anda.

  Untuk salah satu opsi, Anda juga dapat menentukan cara mengautentikasi.
  - Kata Sandi ATAU Kunci — pengguna dapat mengautentikasi dengan kata sandi atau kunci mereka. Ini adalah nilai default.
  - Hanya kata sandi — pengguna harus memberikan kata sandi mereka untuk terhubung.
  - Hanya kunci — pengguna harus menyediakan kunci pribadi mereka untuk terhubung.
  - Kata Sandi dan Kunci — pengguna harus memberikan kunci pribadi dan kata sandi mereka untuk terhubung. Server memeriksa kunci terlebih dahulu, dan kemudian jika kuncinya valid,

sistem meminta kata sandi. Jika kunci pribadi yang diberikan tidak cocok dengan kunci publik yang disimpan, otentikasi gagal.



4. Pilih Berikutnya.
5. Di Pilih titik akhir, lakukan hal berikut:

   a. Untuk tipe Endpoint, pilih tipe titik akhir yang dapat diakses publik. Untuk titik akhir yang dihosting VPC, lihat. Buat server di cloud pribadi virtual

   b. (Opsional) Untuk nama host Kustom, pilih Tidak Ada.

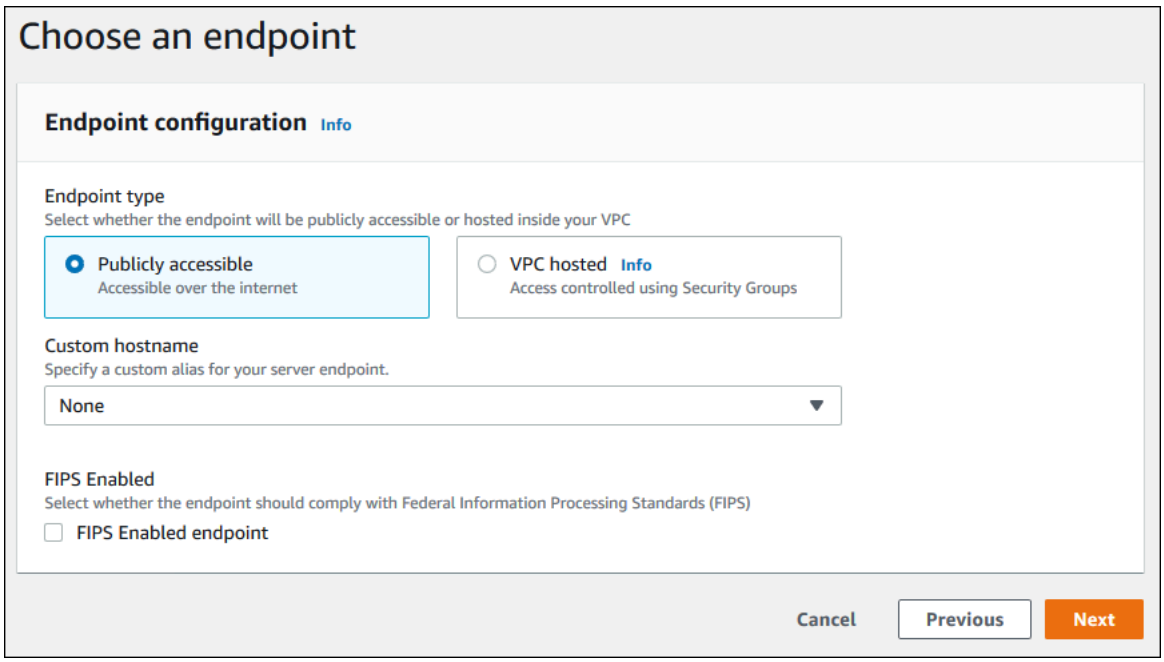
Anda mendapatkan nama host server yang disediakan oleh AWS Transfer Family. Nama host server mengambil formulir`serverId`.server.transfer.`regionId`.amazonaws.com.

Untuk nama host kustom, Anda menentukan alias kustom untuk endpoint server Anda. Untuk mempelajari lebih lanjut tentang bekerja dengan nama host kustom, lihatBekerja dengan nama host khusus.

c. (Opsional) Untuk FIPS Diaktifkan, pilih kotak centang titik akhir Diaktifkan FIPS untuk memastikan bahwa titik akhir sesuai dengan Standar Pemrosesan Informasi Federal (FIPS).

> **Note**
>
> Titik akhir berkemampuan FIPS hanya tersedia di Wilayah Amerika Utara. AWS Untuk Wilayah yang tersedia, lihat AWS Transfer Family titik akhir dan kuota di. Referensi Umum AWS Untuk informasi lebih lanjut tentang FIPS, lihat Federal Information Processing Standard (FIPS) 140-2.

d. Pilih Berikutnya.



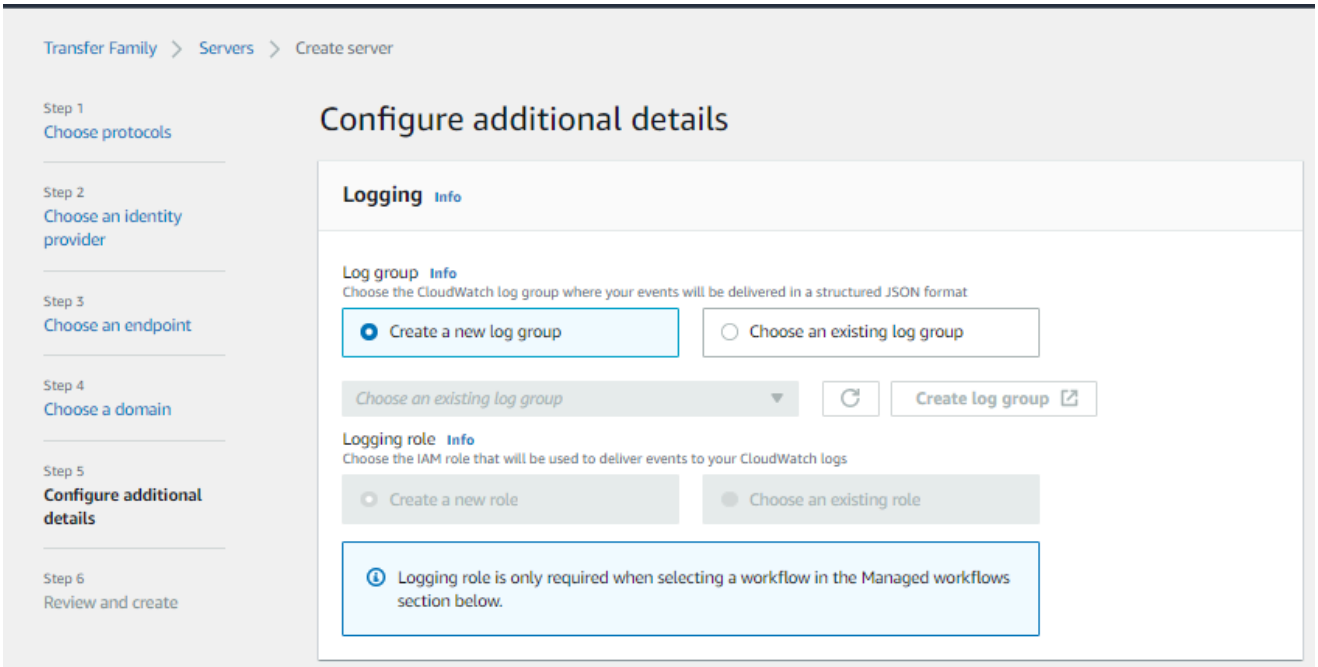
6. Pada halaman Pilih domain, pilih layanan AWS penyimpanan yang ingin Anda gunakan untuk menyimpan dan mengakses data Anda melalui protokol yang dipilih:

- Pilih Amazon S3 untuk menyimpan dan mengakses file Anda sebagai objek di atas protokol yang dipilih.
- Pilih Amazon EFS untuk menyimpan dan mengakses file Anda di sistem file Amazon EFS Anda melalui protokol yang dipilih.

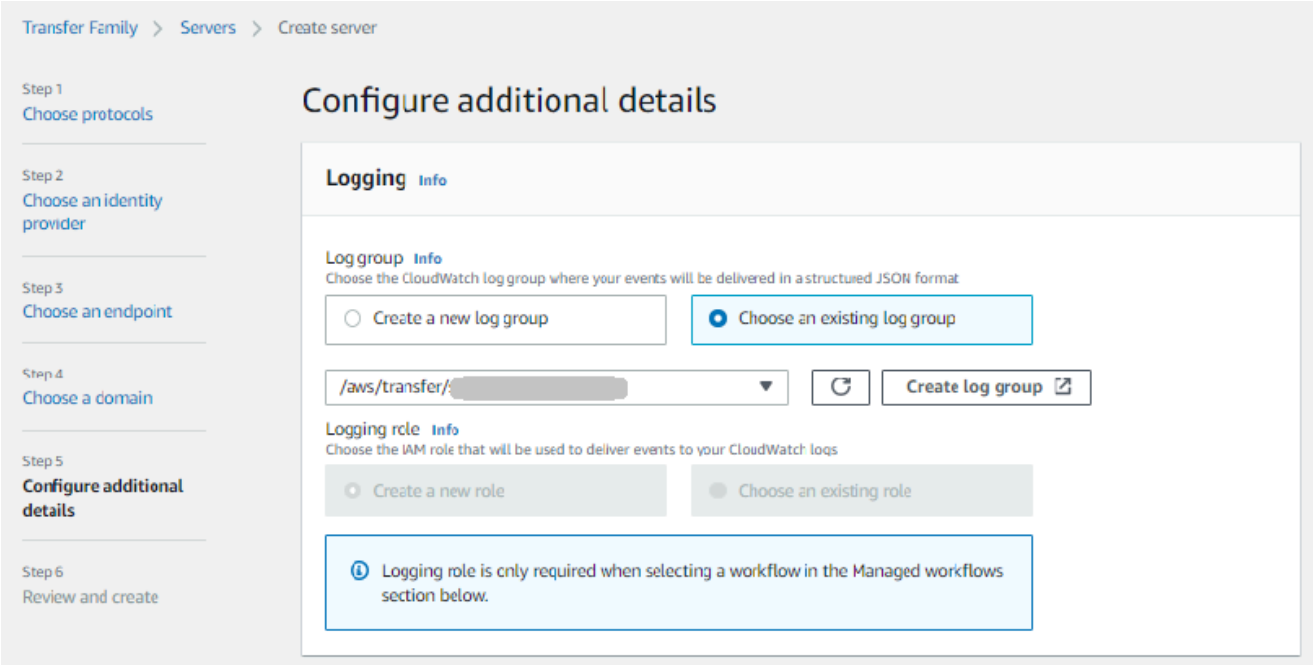
Pilih Berikutnya.

7. Di Konfigurasikan detail tambahan, lakukan hal berikut:

   a. Untuk logging, tentukan grup log yang ada atau buat yang baru (opsi default).

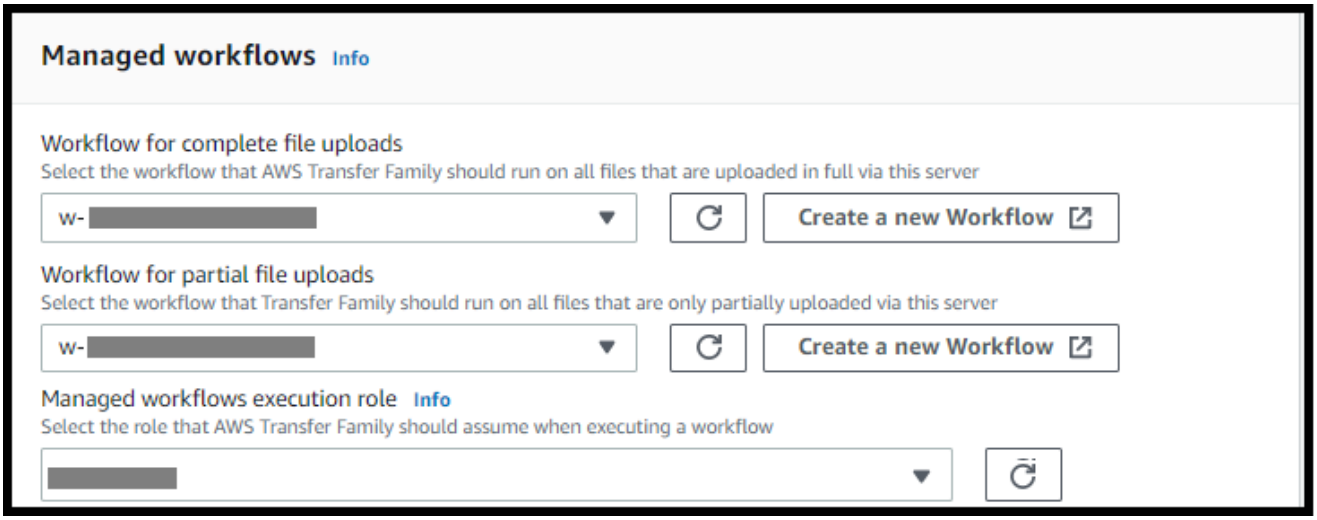


   Jika Anda memilih grup log yang ada, Anda harus memilih salah satu yang terkait dengan Anda Akun AWS.



Jika Anda memilih Buat grup log, CloudWatch konsol (https://console.aws.amazon.com/cloudwatch/) terbuka ke halaman Buat grup log. Untuk detailnya, lihat Membuat grup log di CloudWatch Log.

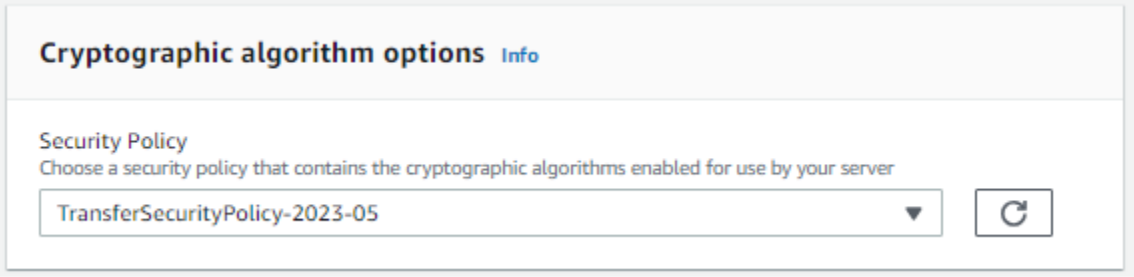
b. (Opsional) Untuk alur kerja Terkelola, pilih ID alur kerja (dan peran terkait) yang harus diasumsikan oleh Transfer Family saat menjalankan alur kerja. Anda dapat memilih satu alur kerja untuk dieksekusi setelah unggahan lengkap, dan satu lagi untuk mengeksekusi pada unggahan sebagian. Untuk mempelajari lebih lanjut tentang memproses file Anda menggunakan alur kerja terkelola, lihatAWS Transfer Family alur kerja terkelola.



c. Untuk opsi algoritma kriptografi, pilih kebijakan keamanan yang berisi algoritma kriptografi yang diaktifkan untuk digunakan oleh server Anda.

> **Note**
>
> Secara default:
>
> - Jika titik akhir FIPS Enabled tidak dipilih, kebijakan `TransferSecurityPolicy-2020-06` keamanan dilampirkan ke server Anda.
> - Jika titik akhir FIPS Enabled dipilih, kebijakan `TransferSecurityPolicy-FIPS-2020-06` keamanan dilampirkan ke server Anda.

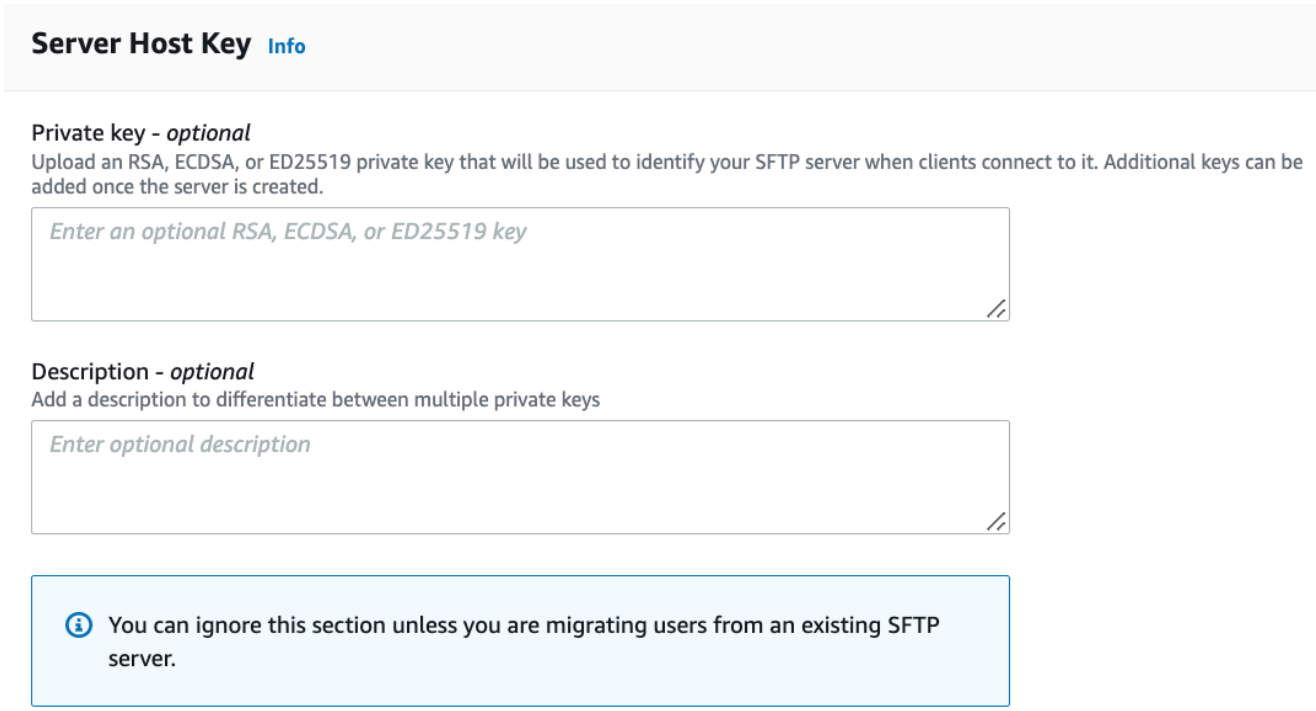
Untuk informasi selengkapnya tentang kebijakan keamanan, lihatKebijakan keamanan untuk AWS Transfer Family server.



d. (Opsional) Untuk Server Host Key, masukkan kunci pribadi RSA, ED25519, atau ECDSA yang akan digunakan untuk mengidentifikasi server Anda ketika klien terhubung ke sana melalui SFTP. Anda juga dapat menambahkan deskripsi untuk membedakan antara beberapa kunci host.

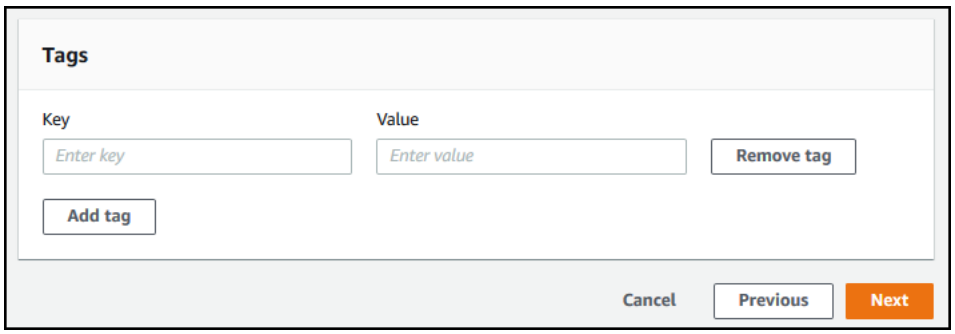
   Setelah Anda membuat server Anda, Anda dapat menambahkan kunci host tambahan. Memiliki beberapa kunci host berguna jika Anda ingin memutar tombol atau jika Anda ingin memiliki berbagai jenis kunci, seperti kunci RSA dan juga kunci ECDSA.

   > **Note**
   >
   > Bagian Server Host Key hanya digunakan untuk memigrasikan pengguna dari server berkemampuan SFTP yang ada.

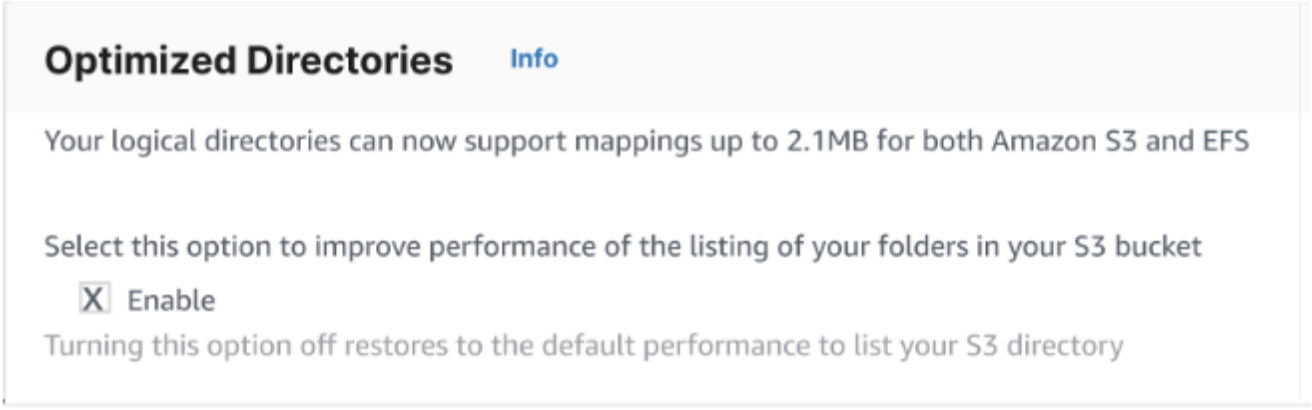


e. (Opsional) Untuk Tag, untuk Kunci dan Nilai, masukkan satu atau beberapa tag sebagai pasangan nilai kunci, lalu pilih Tambahkan tag.

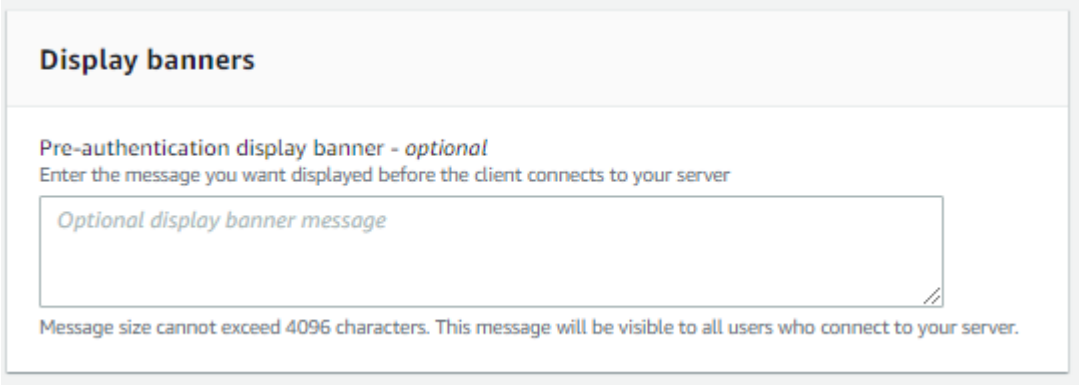
f. Pilih Berikutnya.



g. Anda dapat mengoptimalkan kinerja untuk direktori Amazon S3 Anda. Misalnya, Anda masuk ke direktori home Anda, dan Anda memiliki 10.000 subdirektori. Dengan kata lain, bucket S3 Anda memiliki 10.000 folder. Dalam skenario ini, jika Anda menjalankan perintah `ls` (daftar), operasi daftar memakan waktu antara enam dan delapan menit. Namun, jika Anda mengoptimalkan direktori Anda, operasi ini hanya membutuhkan beberapa detik.

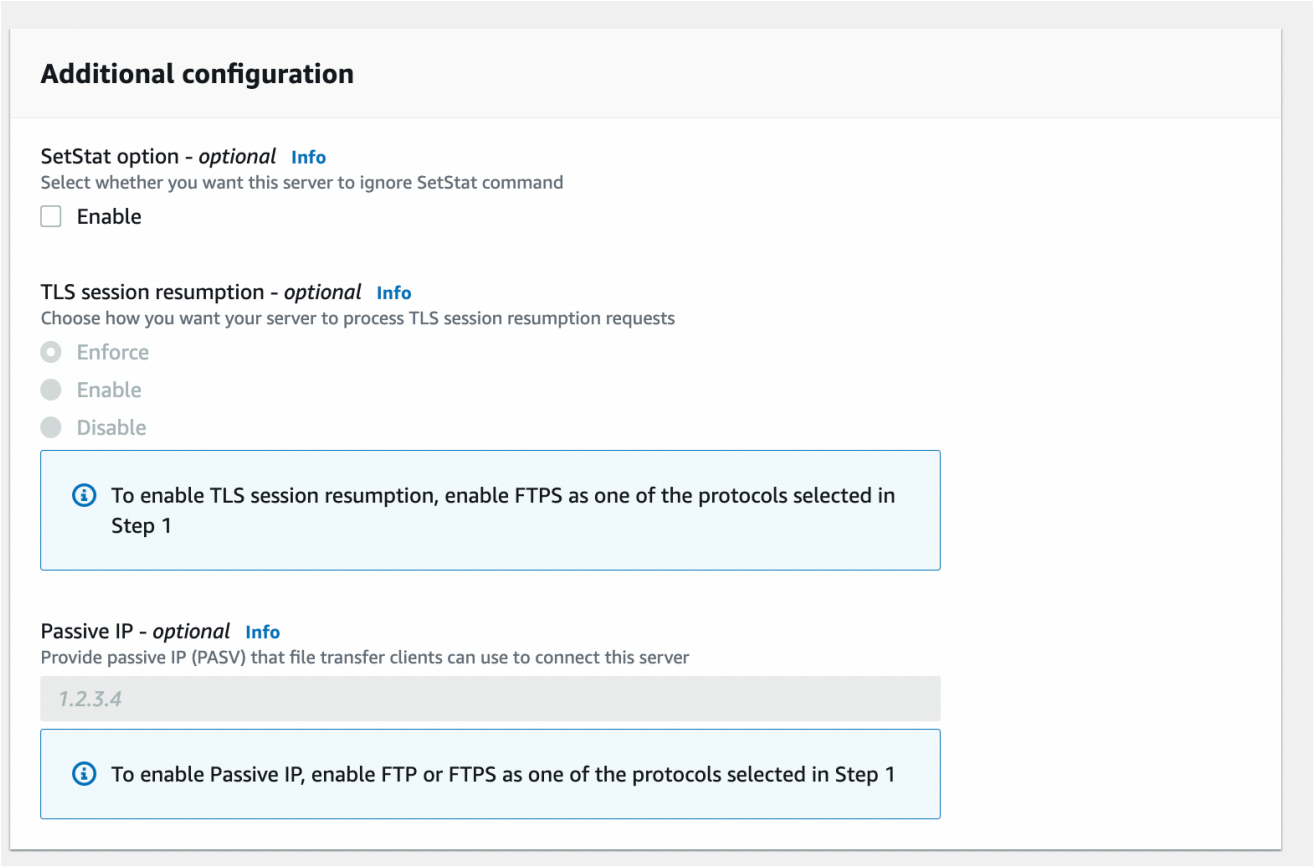


h. (Opsional) Konfigurasikan AWS Transfer Family server untuk menampilkan pesan yang disesuaikan seperti kebijakan organisasi atau syarat dan ketentuan kepada pengguna akhir Anda. Untuk spanduk Tampilan, di kotak teks spanduk tampilan Pra-otentikasi, masukkan pesan teks yang ingin ditampilkan kepada pengguna sebelum mereka mengautentikasi.



i. (Opsional) Anda dapat mengonfigurasi opsi tambahan berikut.

- SetStat opsi: aktifkan opsi ini untuk mengabaikan kesalahan yang dihasilkan saat klien mencoba menggunakan `SETSTAT` pada file yang Anda unggah ke bucket Amazon S3. Untuk detail tambahan, lihat `SetStatOption` dokumentasi di ProtocolDetails.
- Dimulainya kembali sesi TLS: opsi ini hanya tersedia jika Anda telah mengaktifkan FTPS sebagai salah satu protokol untuk server ini.
- IP Pasif: opsi ini hanya tersedia jika Anda telah mengaktifkan FTPS atau FTP sebagai salah satu protokol untuk server ini.



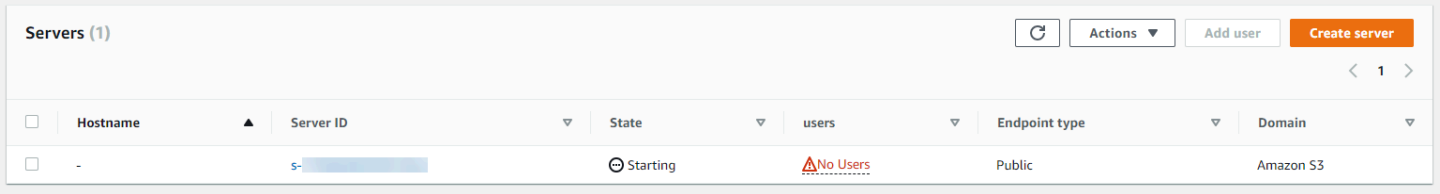
8. Di Tinjau dan buat, tinjau pilihan Anda.

   - Jika Anda ingin mengedit salah satu dari mereka, pilih Edit di sebelah langkah.

     > **Note**
     >
     > Anda harus meninjau setiap langkah setelah langkah yang Anda pilih untuk diedit.

   - Jika Anda tidak memiliki perubahan, pilih Buat server untuk membuat server Anda. Anda dibawa ke halaman Server, ditampilkan berikut, di mana server baru Anda terdaftar.

Diperlukan beberapa menit sebelum status server baru Anda berubah menjadi Online. Pada saat itu, server Anda dapat melakukan operasi file untuk pengguna Anda.



Buat server berkemampuan FTPS

File Transfer Protocol over SSL (FTPS) adalah ekstensi ke FTP. Ini menggunakan protokol kriptografi Transport Layer Security (TLS) dan Secure Sockets Layer (SSL) untuk mengenkripsi lalu lintas. FTPS memungkinkan enkripsi koneksi kontrol dan saluran data baik secara bersamaan maupun independen.

Untuk membuat server berkemampuan FTPS

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/ dan pilih Server dari panel navigasi, lalu pilih Buat server.
2. Di Pilih protokol, pilih FTPS.

   Untuk sertifikat Server, pilih sertifikat yang disimpan di AWS Certificate Manager (ACM) yang akan digunakan untuk mengidentifikasi server Anda ketika klien terhubung ke sana melalui FTPS dan kemudian pilih Berikutnya.

   Untuk meminta sertifikat publik baru, lihat Meminta sertifikat publik di Panduan AWS Certificate Manager Pengguna.

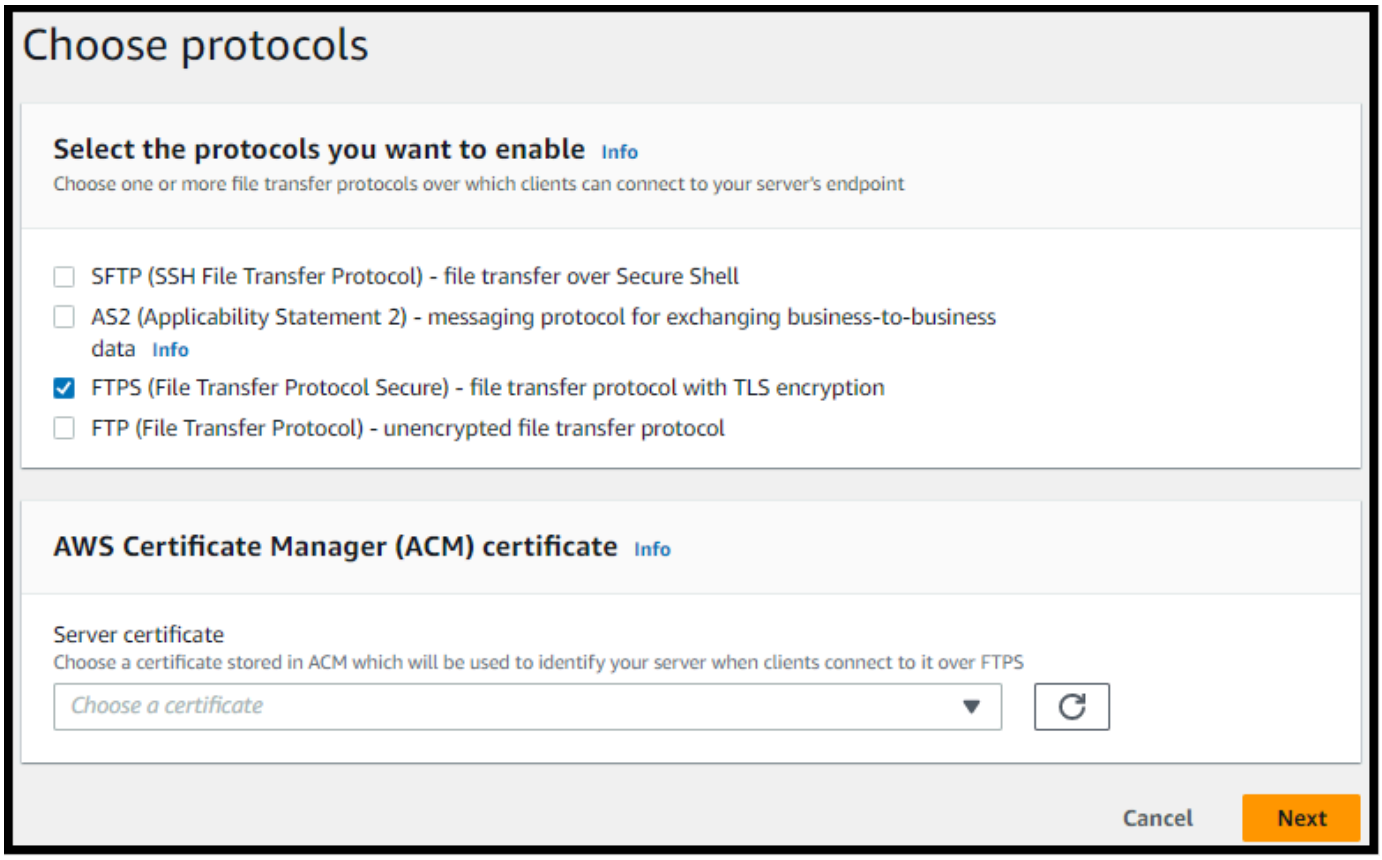
   Untuk mengimpor sertifikat yang ada ke ACM, lihat Mengimpor sertifikat ke ACM di Panduan Pengguna.AWS Certificate Manager

   Untuk meminta sertifikat pribadi untuk menggunakan FTPS melalui alamat IP pribadi, lihat Meminta Sertifikat Pribadi di AWS Certificate Manager Panduan Pengguna.

   Sertifikat dengan algoritme kriptografi dan ukuran kunci berikut didukung:

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

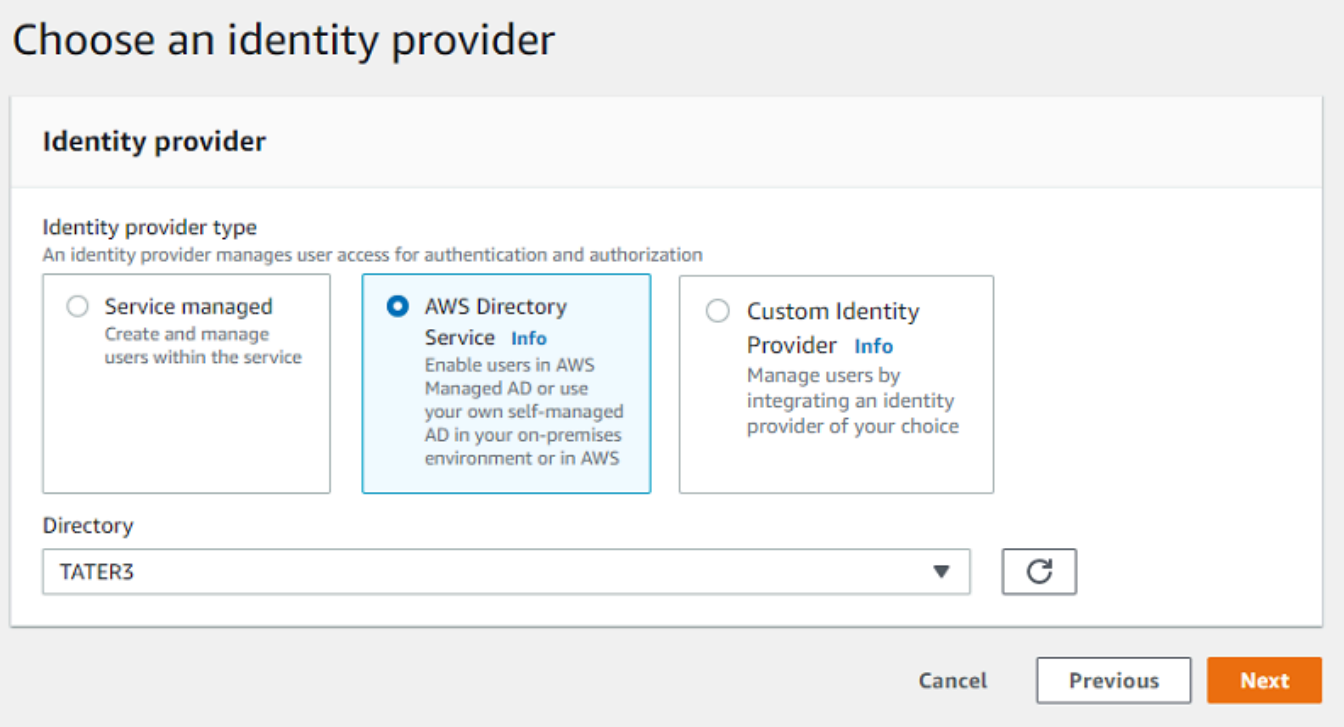
> **Note**
>
> Sertifikat harus berupa sertifikat SSL/TLS X.509 versi 3 yang valid dengan FQDN atau alamat IP yang ditentukan dan informasi tentang penerbitnya.



3. Di Pilih penyedia identitas, pilih penyedia identitas yang ingin Anda gunakan untuk mengelola akses pengguna. Anda memiliki opsi berikut:

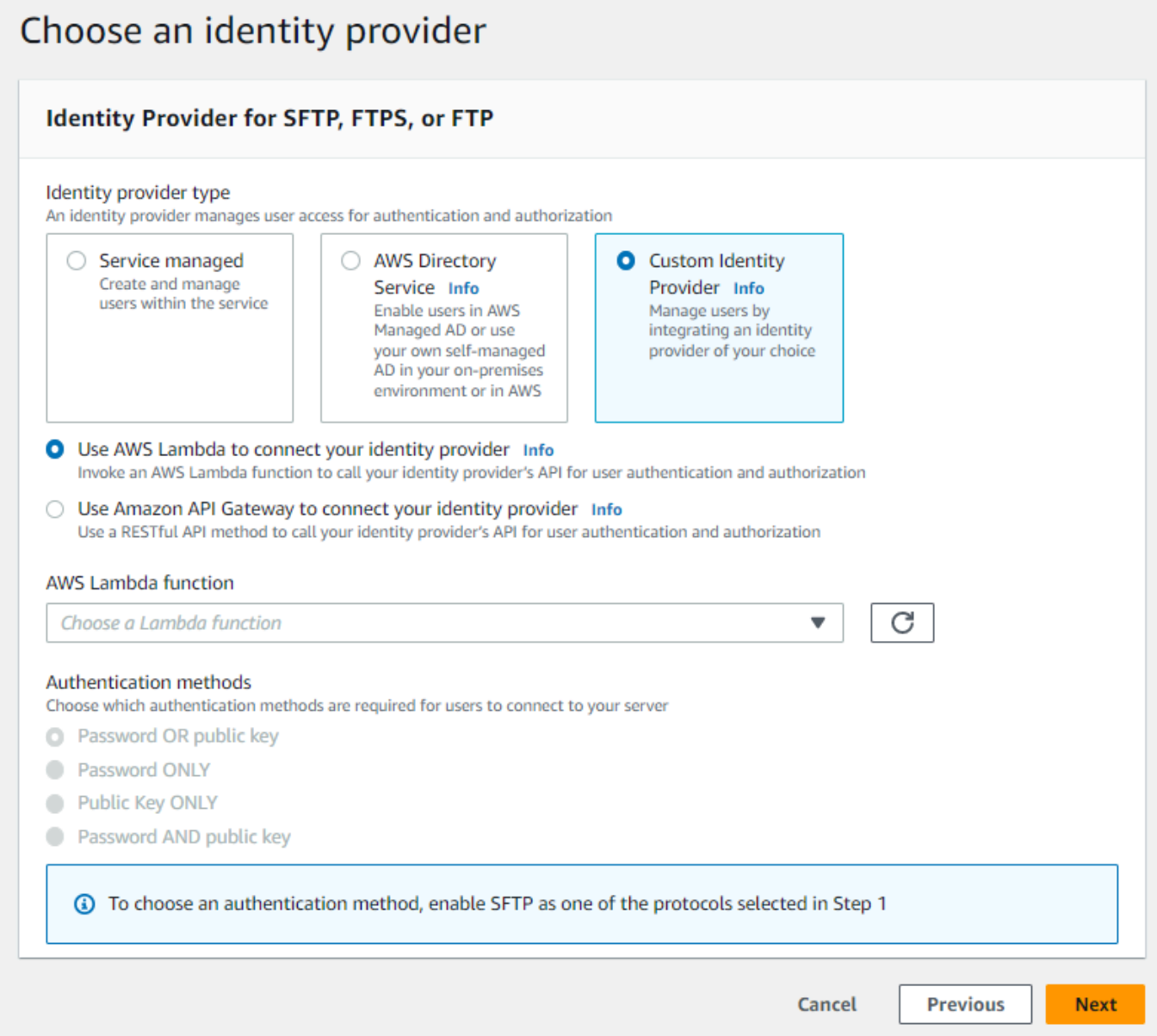
   - AWS Directory Service for Microsoft Active Directory— Anda menyediakan AWS Directory Service direktori untuk mengakses titik akhir. Dengan demikian, Anda dapat menggunakan kredensyal yang disimpan di Active Directory untuk mengautentikasi pengguna Anda. Untuk mempelajari lebih lanjut tentang bekerja dengan penyedia AWS Managed Microsoft AD identitas, lihat[Menggunakan penyedia identitas AWS Directory Service](#).

> **Note**
>
> - Direktori Cross-Account dan Shared tidak didukung untuk. AWS Managed Microsoft AD
> - Untuk menyiapkan server dengan Directory Service sebagai penyedia identitas Anda, Anda perlu menambahkan beberapa AWS Directory Service izin. Untuk detailnya, lihat Sebelum Anda mulai menggunakan AWS Directory Service for Microsoft Active Directory.



- Penyedia identitas khusus - Pilih salah satu opsi berikut:
  - Gunakan AWS Lambda untuk menghubungkan penyedia identitas Anda — Anda dapat menggunakan penyedia identitas yang ada, didukung oleh fungsi Lambda. Anda memberikan nama fungsi Lambda. Untuk informasi selengkapnya, lihat Menggunakan AWS Lambda untuk mengintegrasikan penyedia identitas Anda.
  - Gunakan Amazon API Gateway untuk menghubungkan penyedia identitas Anda — Anda dapat membuat metode API Gateway yang didukung oleh fungsi Lambda untuk digunakan sebagai penyedia identitas. Anda menyediakan URL Amazon API Gateway dan peran

pemanggilan. Untuk informasi selengkapnya, lihat Menggunakan Amazon API Gateway untuk mengintegrasikan penyedia identitas Anda.



4. Pilih Berikutnya.
5. Di Pilih titik akhir, lakukan hal berikut:

> **Note**
>
> Server FTPS untuk Transfer Family beroperasi melalui Port 21 (Saluran Kontrol) dan Rentang Port 8192-8200 (Saluran Data).

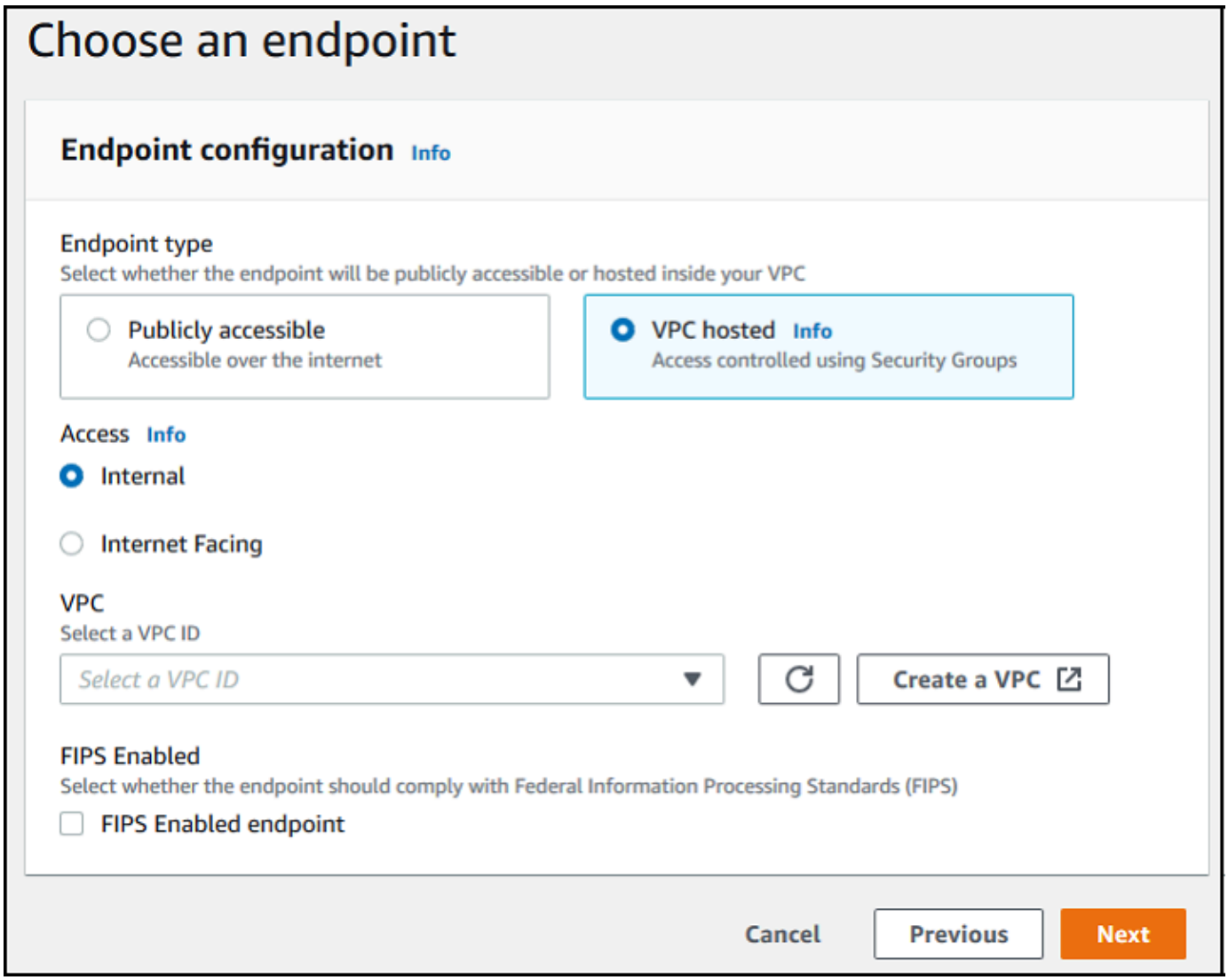
a. Untuk tipe Endpoint, pilih tipe endpoint yang dihosting VPC untuk meng-host endpoint server Anda. Untuk informasi tentang menyiapkan titik akhir yang dihosting VPC Anda, lihat. Buat server di cloud pribadi virtual

> **Note**
>
> Titik akhir yang dapat diakses publik tidak didukung.

b. (Opsional) Untuk FIPS Diaktifkan, pilih kotak centang titik akhir Diaktifkan FIPS untuk memastikan bahwa titik akhir sesuai dengan Standar Pemrosesan Informasi Federal (FIPS).

> **Note**
>
> Titik akhir berkemampuan FIPS hanya tersedia di Wilayah Amerika Utara. AWS Untuk Wilayah yang tersedia, lihat AWS Transfer Family titik akhir dan kuota di. Referensi Umum AWS Untuk informasi lebih lanjut tentang FIPS, lihat Federal Information Processing Standard (FIPS) 140-2.

c. Pilih Berikutnya.



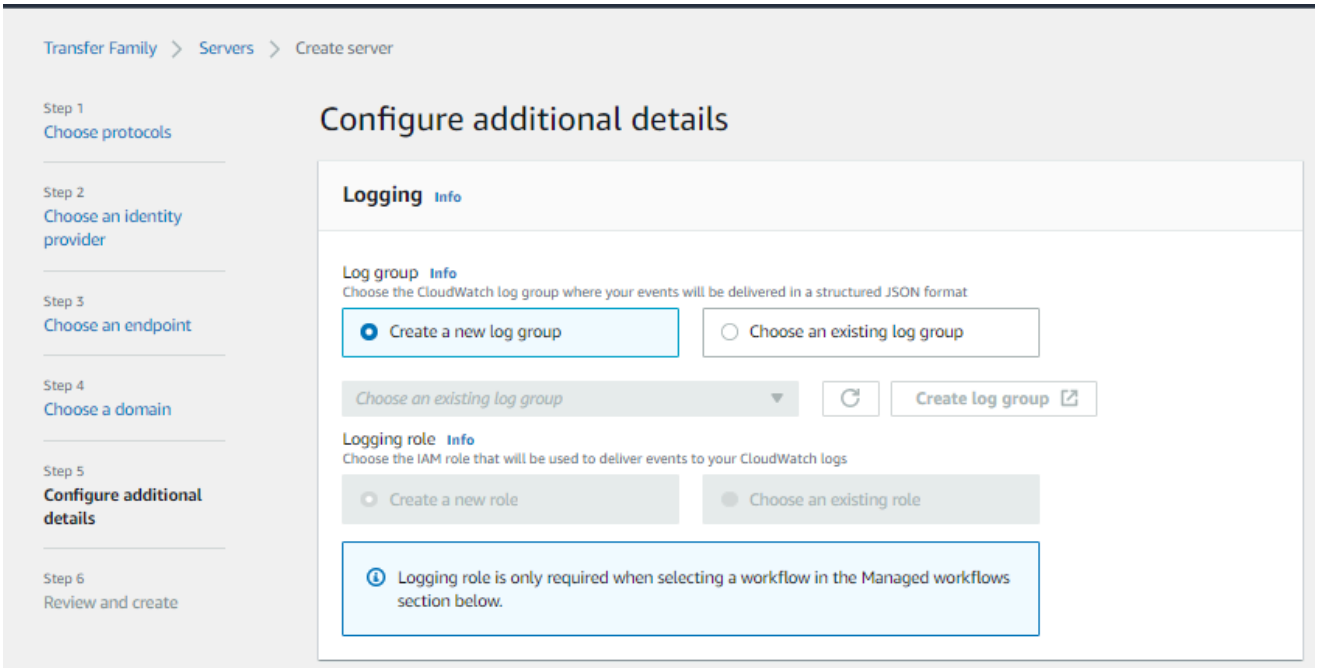
6. Pada halaman Pilih domain, pilih layanan AWS penyimpanan yang ingin Anda gunakan untuk menyimpan dan mengakses data Anda melalui protokol yang dipilih:

   - Pilih Amazon S3 untuk menyimpan dan mengakses file Anda sebagai objek di atas protokol yang dipilih.
   - Pilih Amazon EFS untuk menyimpan dan mengakses file Anda di sistem file Amazon EFS Anda melalui protokol yang dipilih.

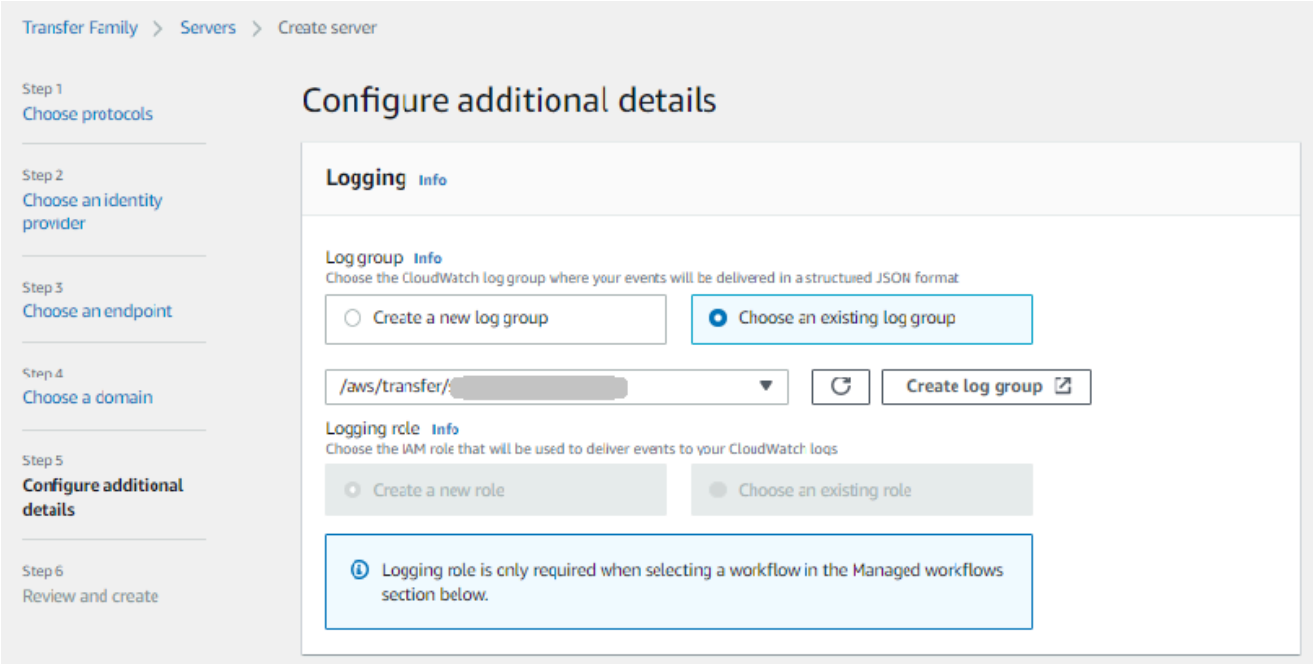
   Pilih Berikutnya.

7. Di Konfigurasikan detail tambahan, lakukan hal berikut:

   a. Untuk logging, tentukan grup log yang ada atau buat yang baru (opsi default).

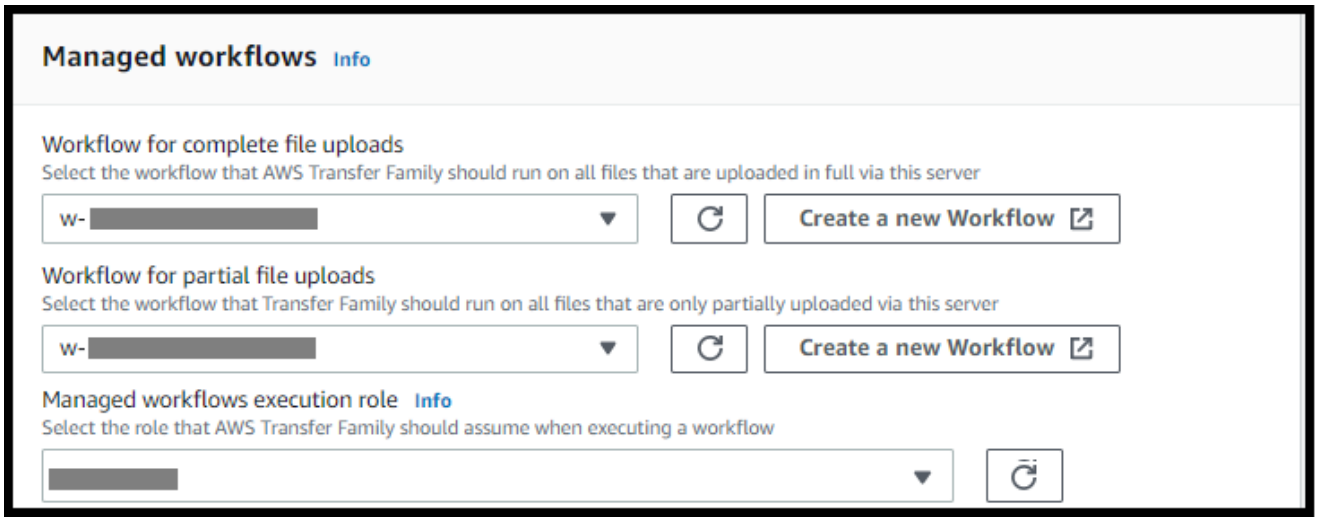


Jika Anda memilih grup log yang ada, Anda harus memilih salah satu yang terkait dengan Anda Akun AWS.



Jika Anda memilih Buat grup log, CloudWatch konsol (https://console.aws.amazon.com/cloudwatch/) terbuka ke halaman Buat grup log. Untuk detailnya, lihat Membuat grup log di CloudWatch Log.

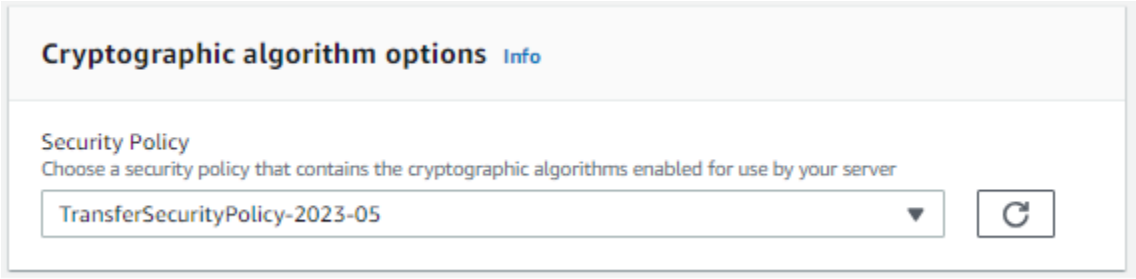
b. (Opsional) Untuk alur kerja Terkelola, pilih ID alur kerja (dan peran terkait) yang harus diasumsikan oleh Transfer Family saat menjalankan alur kerja. Anda dapat memilih satu alur kerja untuk dieksekusi setelah unggahan lengkap, dan satu lagi untuk mengeksekusi pada unggahan sebagian. Untuk mempelajari lebih lanjut tentang memproses file Anda menggunakan alur kerja terkelola, lihatAWS Transfer Family alur kerja terkelola.



c. Untuk opsi algoritma kriptografi, pilih kebijakan keamanan yang berisi algoritma kriptografi yang diaktifkan untuk digunakan oleh server Anda.

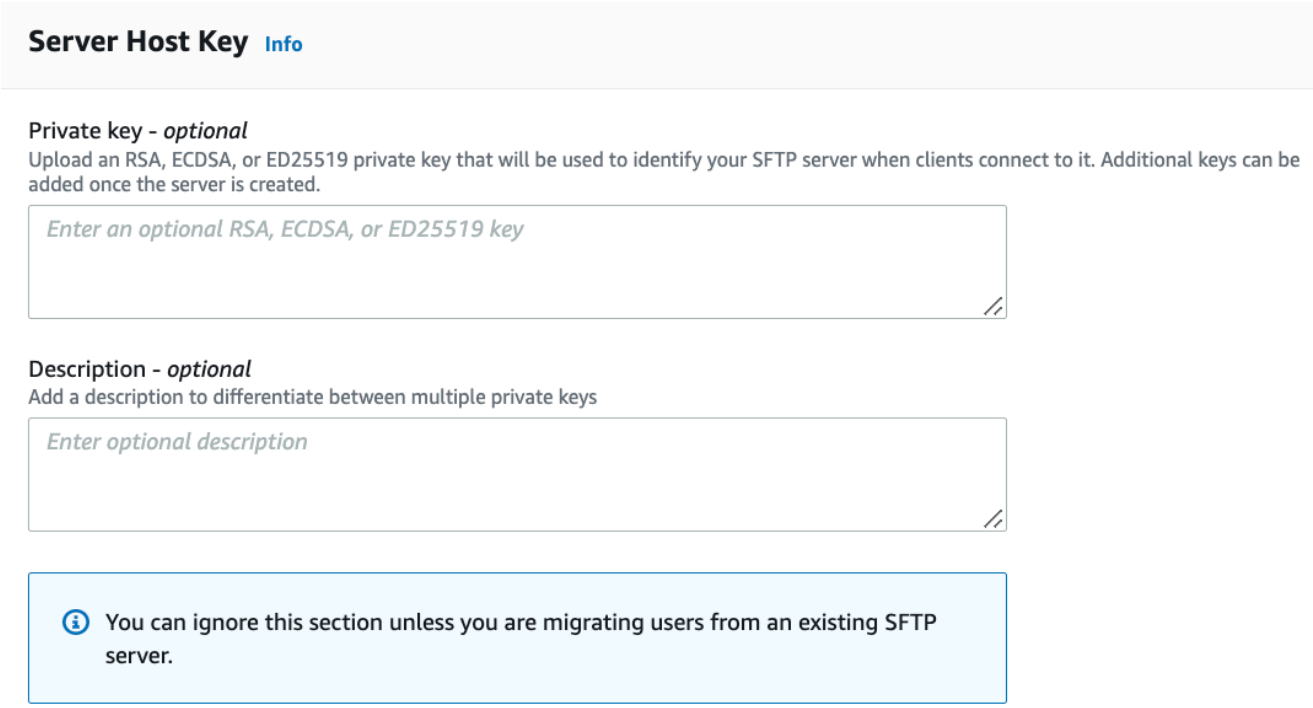
> **Note**
>
> Secara default:
>
> - Jika titik akhir FIPS Enabled tidak dipilih, kebijakan `TransferSecurityPolicy-2020-06` keamanan dilampirkan ke server Anda.
> - Jika titik akhir FIPS Enabled dipilih, kebijakan `TransferSecurityPolicy-FIPS-2020-06` keamanan dilampirkan ke server Anda.

Untuk informasi selengkapnya tentang kebijakan keamanan, lihatKebijakan keamanan untuk AWS Transfer Family server.



d. Untuk Server Host Key, biarkan kosong.

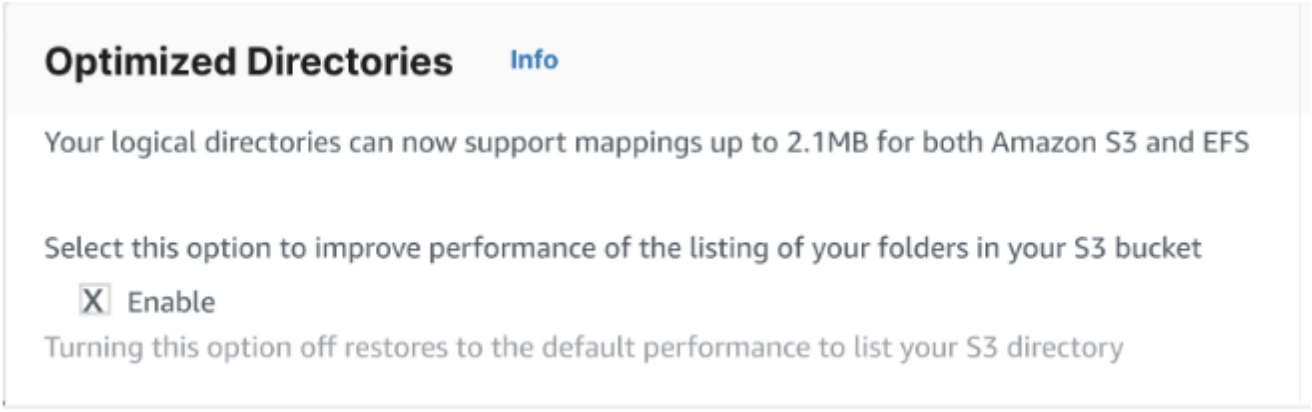
> **Note**
>
> Bagian Server Host Key hanya digunakan untuk memigrasikan pengguna dari server berkemampuan SFTP yang ada.



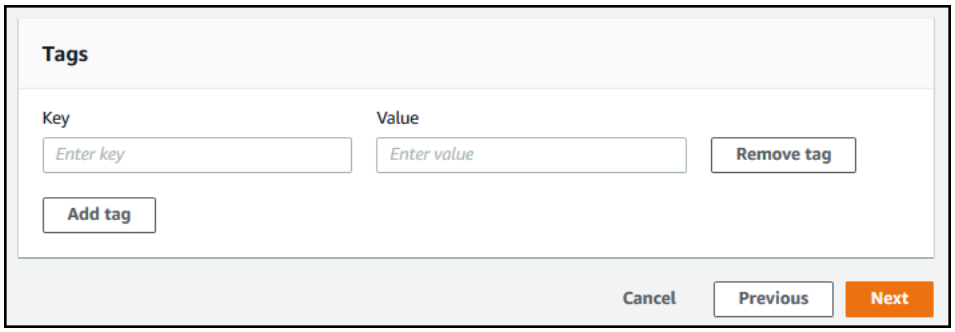
e. (Opsional) Untuk Tag, untuk Kunci dan Nilai, masukkan satu atau beberapa tag sebagai pasangan nilai kunci, lalu pilih Tambahkan tag.

f. Anda dapat mengoptimalkan kinerja untuk direktori Amazon S3 Anda. Misalnya, Anda masuk ke direktori home Anda, dan Anda memiliki 10.000 subdirektori. Dengan kata lain, bucket S3 Anda memiliki 10.000 folder. Dalam skenario ini, jika Anda menjalankan perintah

`ls` (daftar), operasi daftar memakan waktu antara enam dan delapan menit. Namun, jika Anda mengoptimalkan direktori Anda, operasi ini hanya membutuhkan beberapa detik.

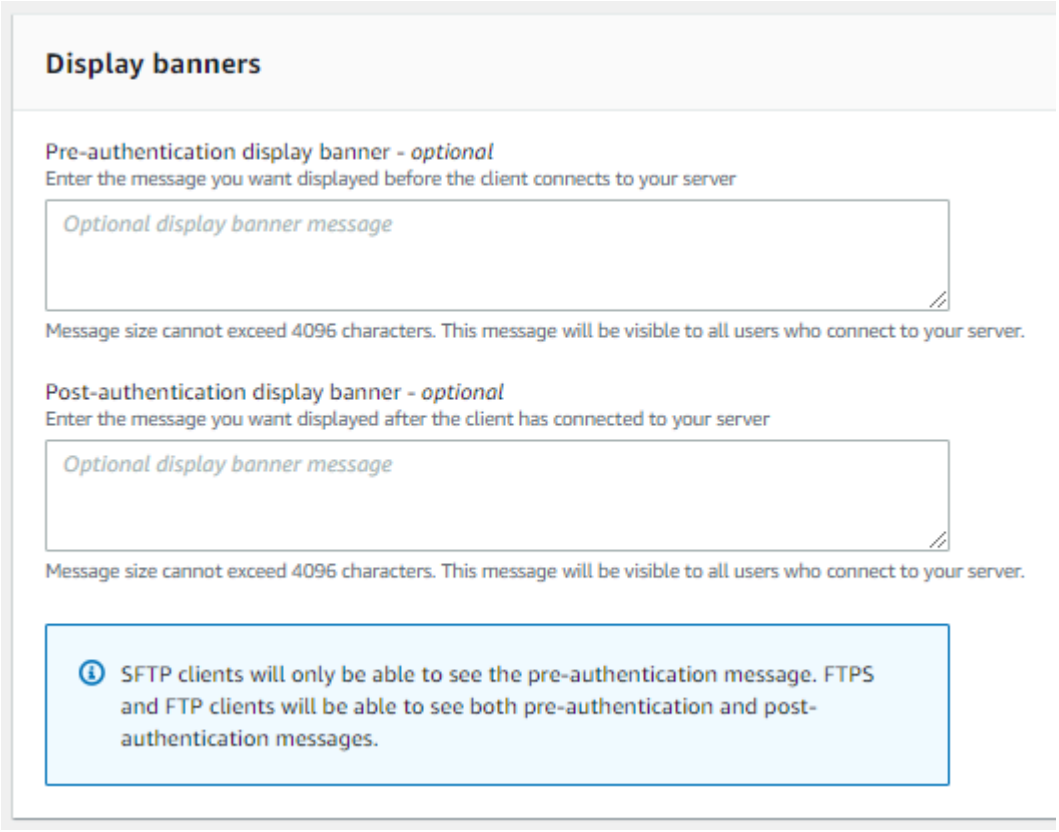


g. Pilih Berikutnya.



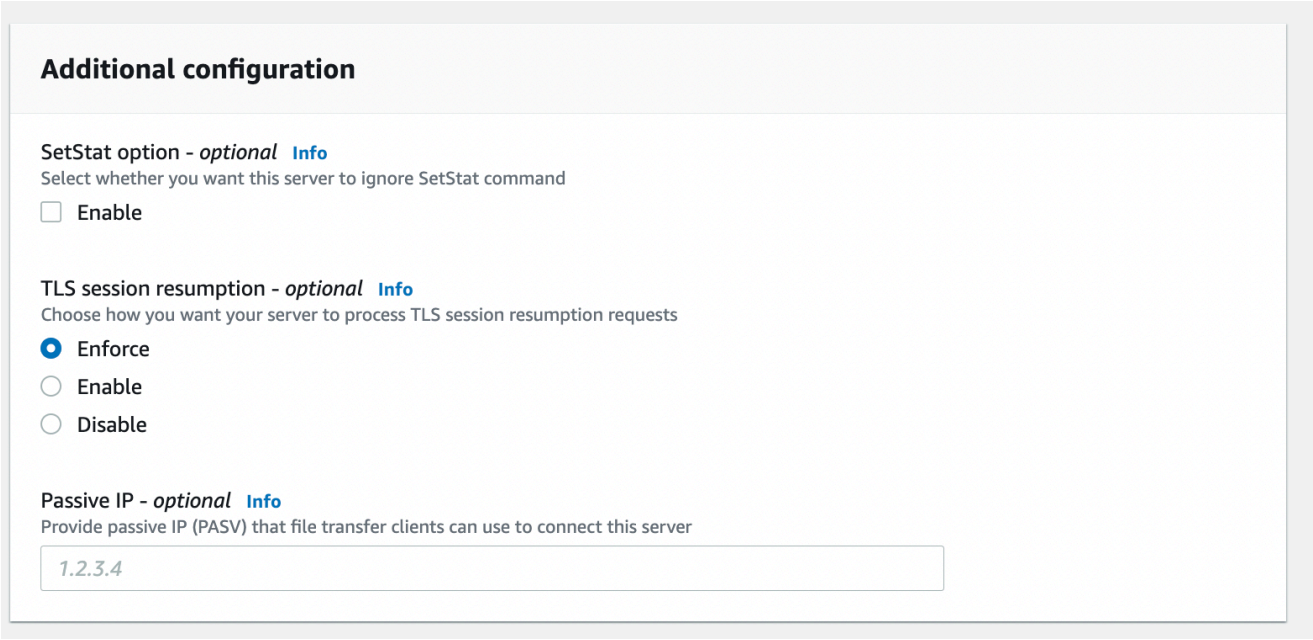
h. (Opsional) Anda dapat mengonfigurasi AWS Transfer Family server untuk menampilkan pesan yang disesuaikan seperti kebijakan organisasi atau syarat dan ketentuan kepada pengguna akhir Anda. Anda juga dapat menampilkan Message of The Day (MOTD) yang disesuaikan kepada pengguna yang telah berhasil diautentikasi.

   Untuk spanduk Tampilan, di kotak teks spanduk tampilan Pra-otentikasi, masukkan pesan teks yang ingin ditampilkan kepada pengguna Anda sebelum mereka mengautentikasi, dan di kotak teks spanduk tampilan pasca-otentikasi, masukkan teks yang ingin ditampilkan kepada pengguna Anda setelah mereka berhasil mengautentikasi.



i. (Opsional) Anda dapat mengonfigurasi opsi tambahan berikut.

- SetStat opsi: aktifkan opsi ini untuk mengabaikan kesalahan yang dihasilkan saat klien mencoba menggunakan `SETSTAT` pada file yang Anda unggah ke bucket Amazon S3. Untuk detail tambahan, lihat `SetStatOption` dokumentasi dalam ProtocolDetailstopik.
- Dimulainya kembali sesi TLS: menyediakan mekanisme untuk melanjutkan atau berbagi kunci rahasia yang dinegosiasikan antara kontrol dan koneksi data untuk sesi FTPS. Untuk detail tambahan, lihat `TlsSessionResumptionMode` dokumentasi dalam ProtocolDetailstopik.
- IP pasif: menunjukkan mode pasif, untuk protokol FTP dan FTPS. Masukkan satu alamat IPv4, seperti alamat IP publik firewall, router, atau penyeimbang beban. Untuk detail tambahan, lihat `PassiveIp` dokumentasi dalam ProtocolDetailstopik.



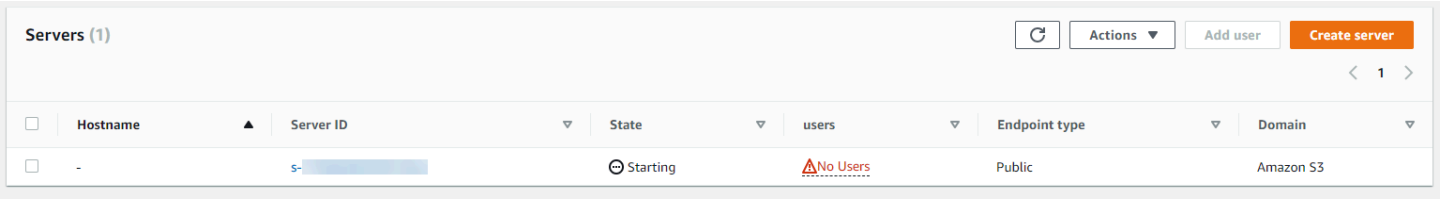
8. Di Tinjau dan buat, tinjau pilihan Anda.

   - Jika Anda ingin mengedit salah satu dari mereka, pilih Edit di sebelah langkah.

     > **Note**
     >
     > Anda harus meninjau setiap langkah setelah langkah yang Anda pilih untuk diedit.

   - Jika Anda tidak memiliki perubahan, pilih Buat server untuk membuat server Anda. Anda dibawa ke halaman Server, ditampilkan berikut, di mana server baru Anda terdaftar.

Diperlukan beberapa menit sebelum status server baru Anda berubah menjadi Online. Pada saat itu, server Anda dapat melakukan operasi file untuk pengguna Anda.



Langkah selanjutnya: Untuk langkah selanjutnya, lanjutkan Bekerja dengan penyedia identitas khusus untuk mengatur pengguna.

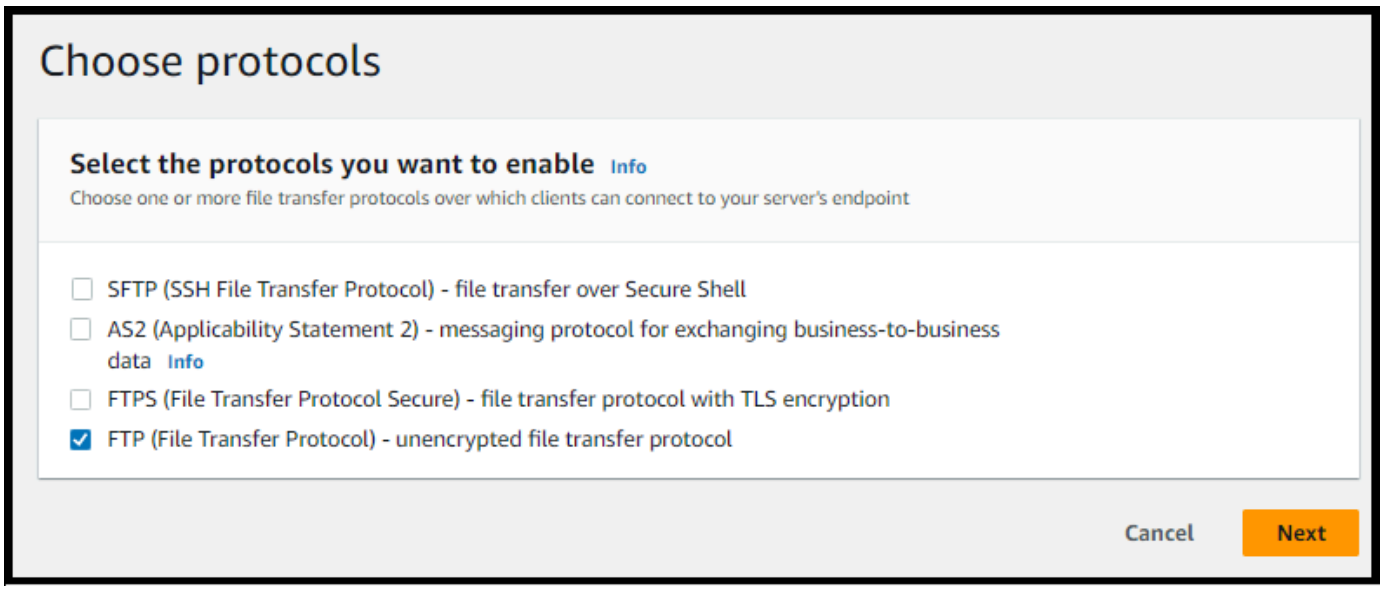
Buat server berkemampuan FTP

File Transfer Protocol (FTP) adalah protokol jaringan yang digunakan untuk transfer data. FTP menggunakan saluran terpisah untuk kontrol dan transfer data. Saluran kontrol terbuka hingga batas waktu dihentikan atau tidak aktif. Saluran data aktif selama transfer. FTP menggunakan teks yang jelas dan tidak mendukung enkripsi lalu lintas.

> **Note**
>
> Saat Anda mengaktifkan FTP, Anda harus memilih opsi akses internal untuk titik akhir yang dihosting VPC. Jika Anda membutuhkan server Anda untuk memiliki data melintasi jaringan publik, Anda harus menggunakan protokol aman, seperti SFTP atau FTPS.

Untuk membuat server berkemampuan FTP

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/ dan pilih Server dari panel navigasi, lalu pilih Buat server.
2. Di Pilih protokol, pilih FTP, lalu pilih Berikutnya.

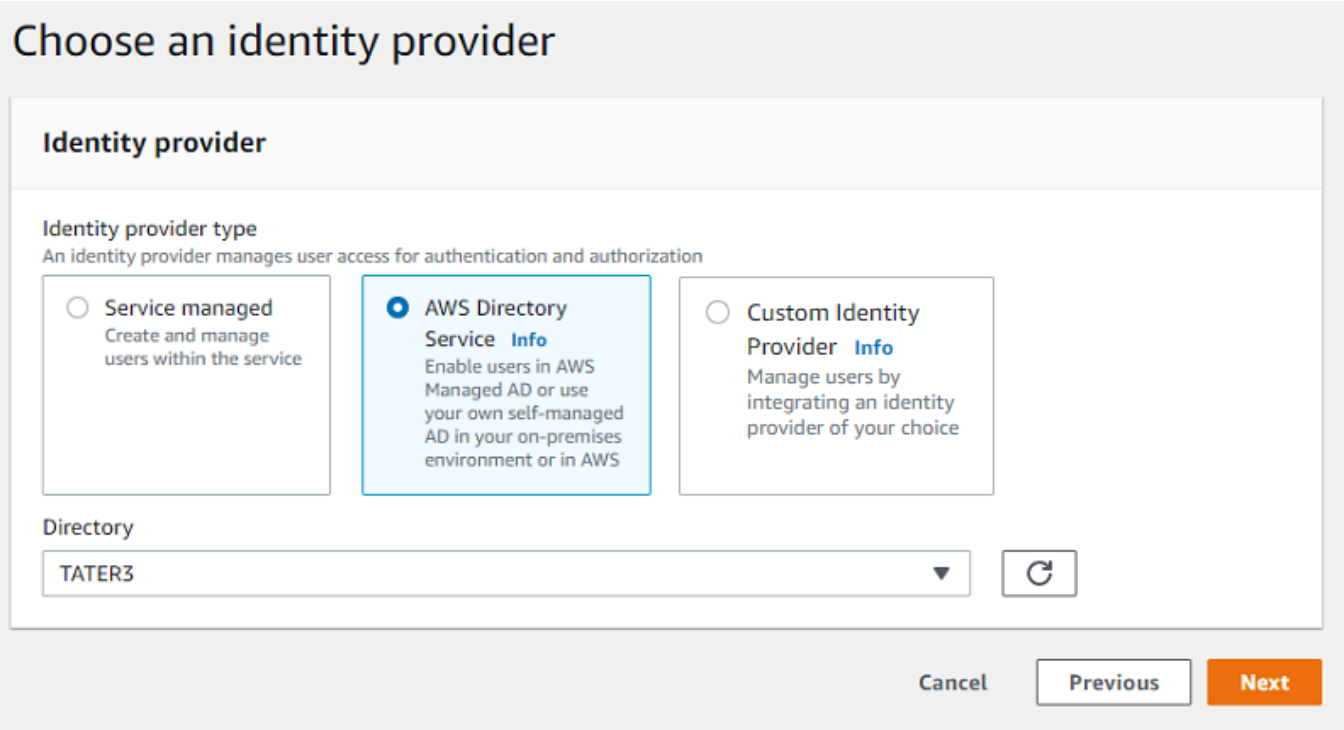


3. Di Pilih penyedia identitas, pilih penyedia identitas yang ingin Anda gunakan untuk mengelola akses pengguna. Anda memiliki opsi berikut:

   - AWS Directory Service for Microsoft Active Directory— Anda menyediakan AWS Directory Service direktori untuk mengakses titik akhir. Dengan demikian, Anda dapat menggunakan kredensyal yang disimpan di Active Directory untuk mengautentikasi pengguna Anda. Untuk

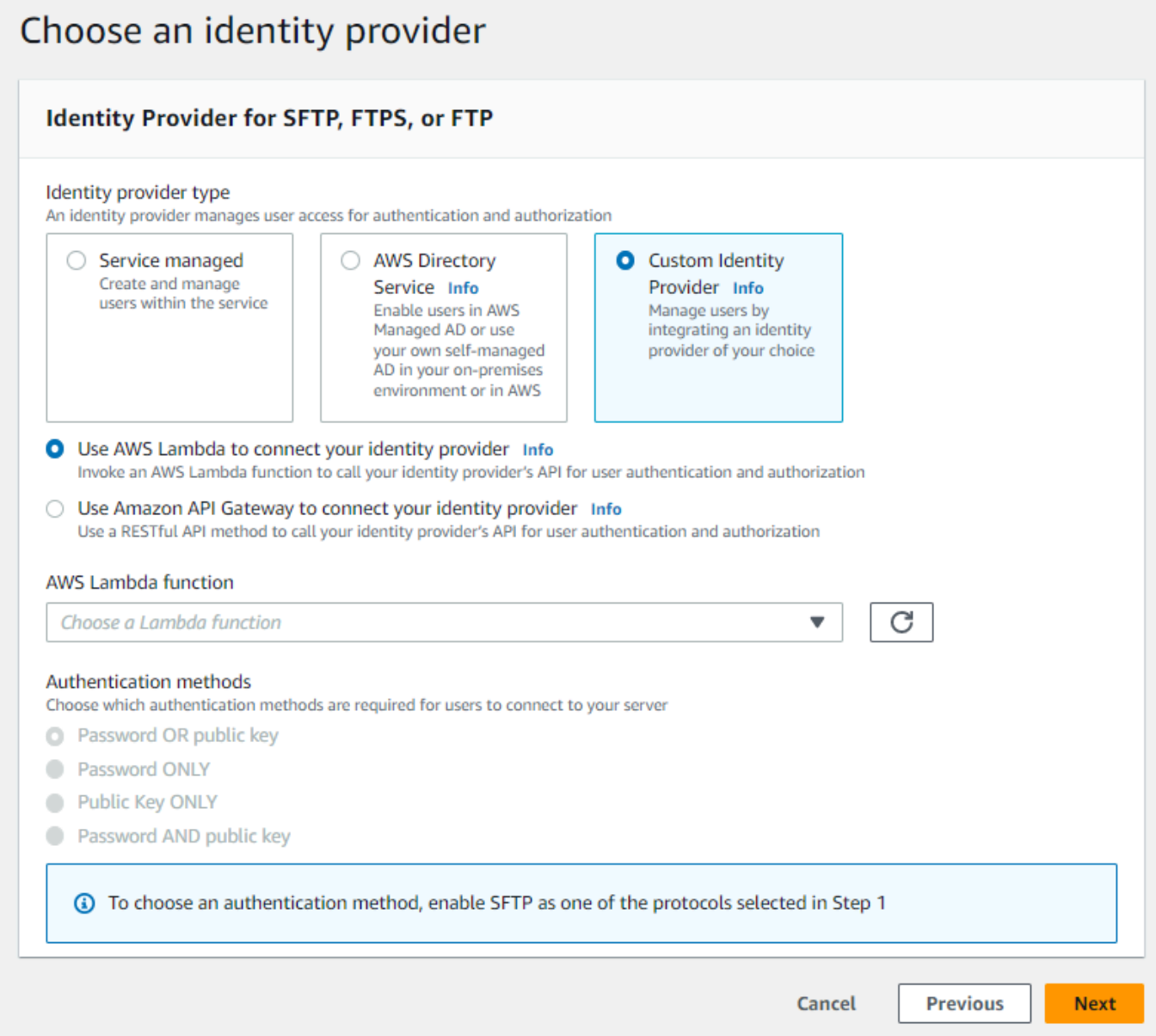
mempelajari lebih lanjut tentang bekerja dengan penyedia AWS Managed Microsoft AD identitas, lihatMenggunakan penyedia identitas AWS Directory Service.

> **Note**
>
> - Direktori Cross-Account dan Shared tidak didukung untuk. AWS Managed Microsoft AD
> - Untuk menyiapkan server dengan Directory Service sebagai penyedia identitas Anda, Anda perlu menambahkan beberapa AWS Directory Service izin. Untuk detailnya, lihat Sebelum Anda mulai menggunakan AWS Directory Service for Microsoft Active Directory.



- Penyedia identitas khusus - Pilih salah satu opsi berikut:
  - Gunakan AWS Lambda untuk menghubungkan penyedia identitas Anda — Anda dapat menggunakan penyedia identitas yang ada, didukung oleh fungsi Lambda. Anda memberikan nama fungsi Lambda. Untuk informasi selengkapnya, lihat Menggunakan AWS Lambda untuk mengintegrasikan penyedia identitas Anda.
  - Gunakan Amazon API Gateway untuk menghubungkan penyedia identitas Anda — Anda dapat membuat metode API Gateway yang didukung oleh fungsi Lambda untuk digunakan

sebagai penyedia identitas. Anda menyediakan URL Amazon API Gateway dan peran pemanggilan. Untuk informasi selengkapnya, lihat Menggunakan Amazon API Gateway untuk mengintegrasikan penyedia identitas Anda.



4. Pilih Berikutnya.
5. Di Pilih titik akhir, lakukan hal berikut:

> **Note**
>
> Server FTP untuk Transfer Family beroperasi melalui Port 21 (Saluran Kontrol) dan Rentang Port 8192-8200 (Saluran Data).

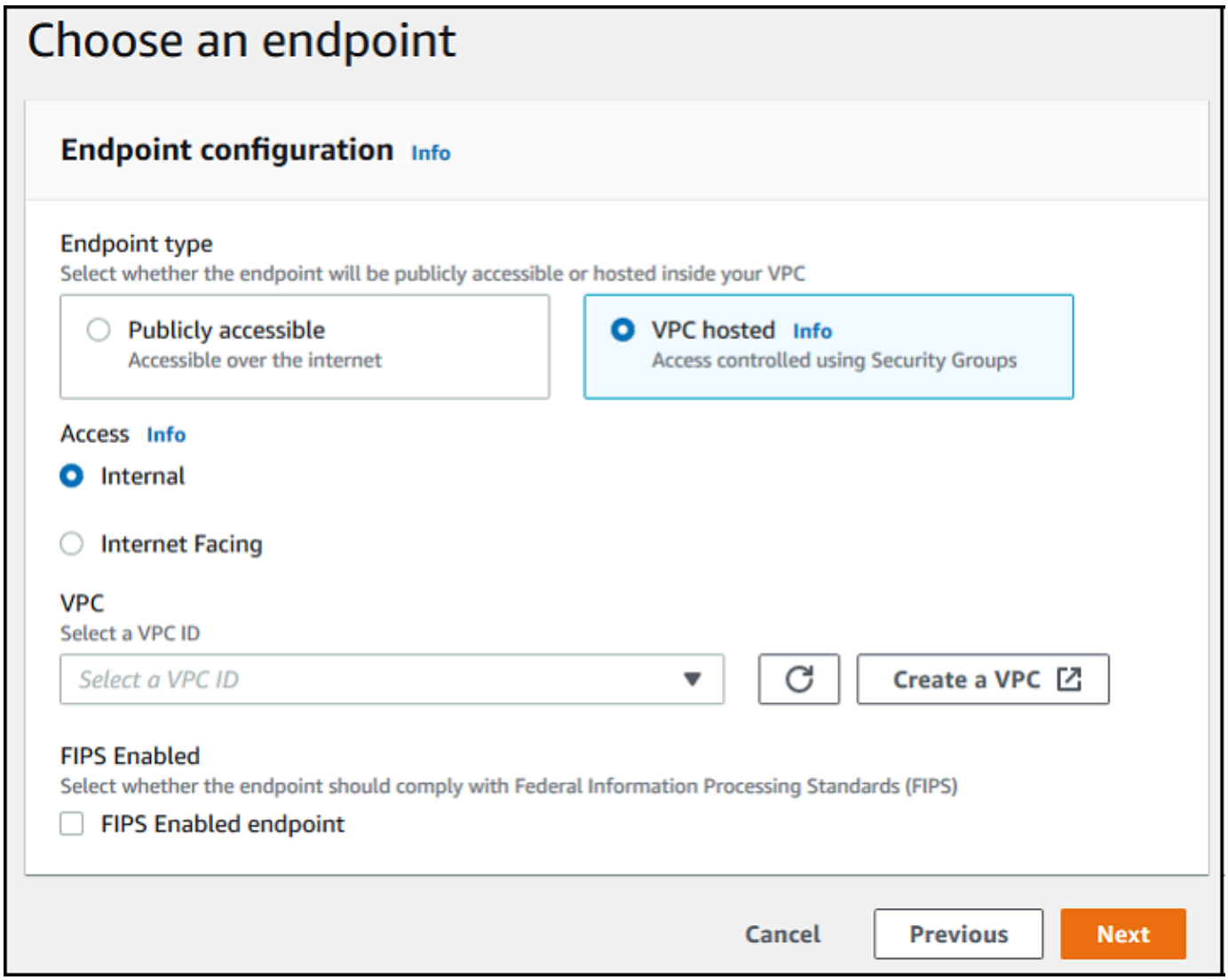
a. Untuk jenis Endpoint, pilih VPC yang dihosting untuk meng-host endpoint server Anda. Untuk informasi tentang menyiapkan titik akhir yang dihosting VPC Anda, lihat. Buat server di cloud pribadi virtual

> **Note**
>
> Titik akhir yang dapat diakses publik tidak didukung.

b. Untuk FIPS Diaktifkan, biarkan kotak centang titik akhir FIPS Enabled tetap bersih.

> **Note**
>
> Endpoint berkemampuan FIPS tidak didukung untuk server FTP.

c. Pilih Berikutnya.



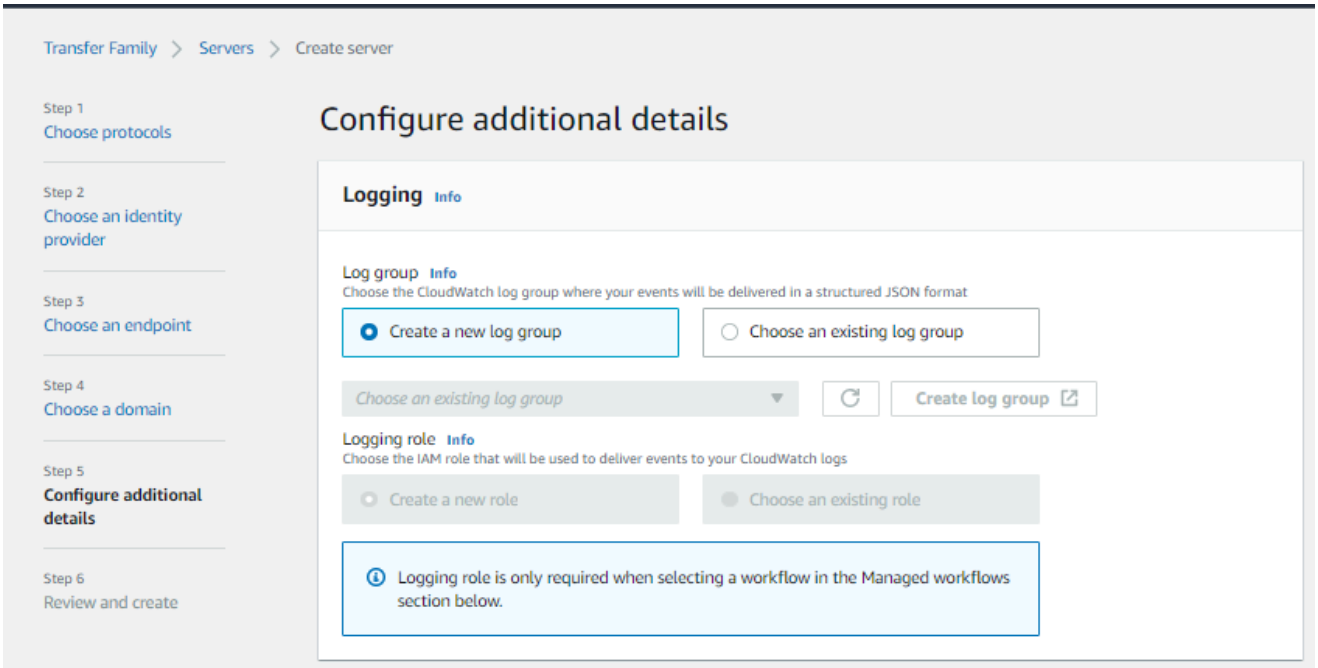
6. Pada halaman Pilih domain, pilih layanan AWS penyimpanan yang ingin Anda gunakan untuk menyimpan dan mengakses data Anda melalui protokol yang dipilih.

   - Pilih Amazon S3 untuk menyimpan dan mengakses file Anda sebagai objek di atas protokol yang dipilih.
   - Pilih Amazon EFS untuk menyimpan dan mengakses file Anda di sistem file Amazon EFS Anda melalui protokol yang dipilih.

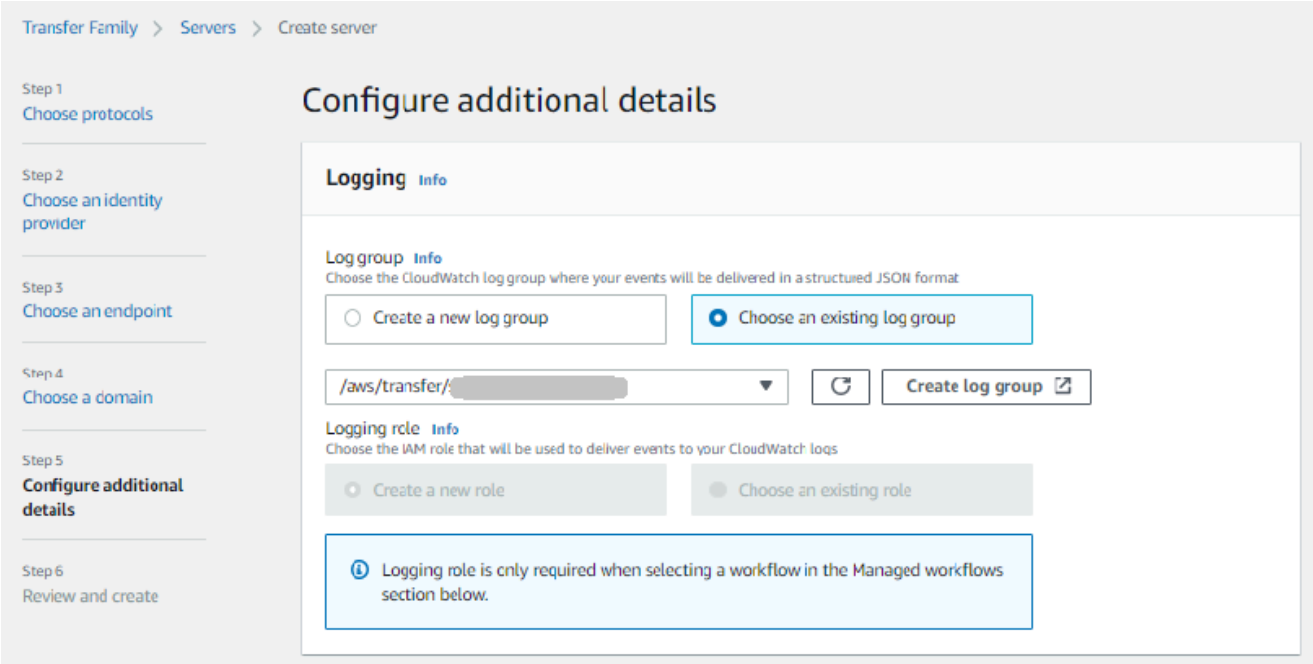
   Pilih Berikutnya.

7. Di Konfigurasikan detail tambahan, lakukan hal berikut:

   a. Untuk logging, tentukan grup log yang ada atau buat yang baru (opsi default).

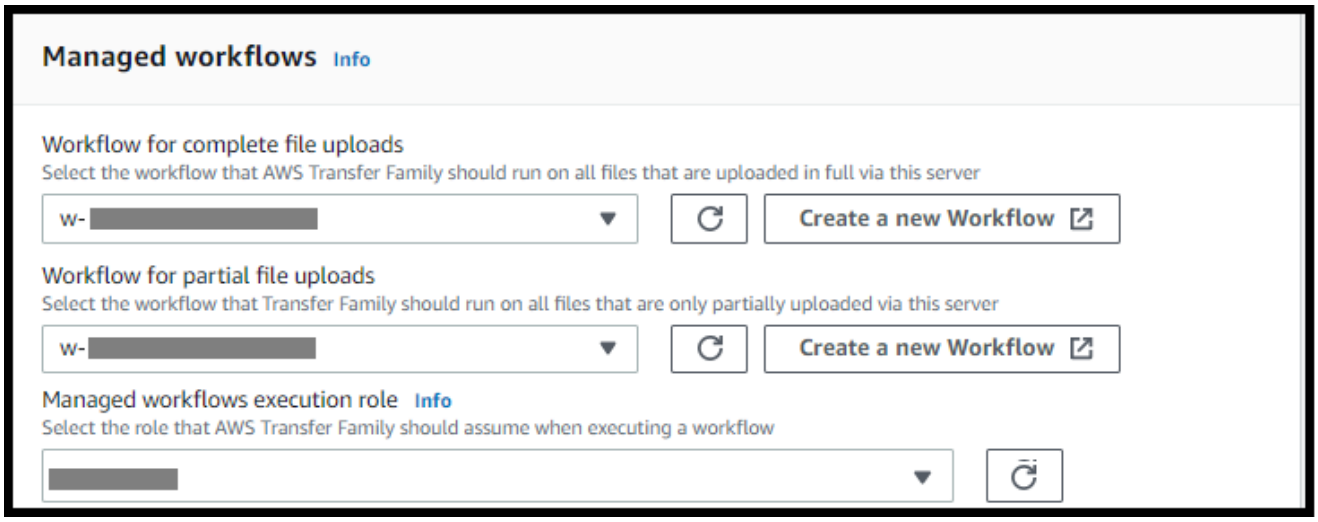


Jika Anda memilih grup log yang ada, Anda harus memilih salah satu yang terkait dengan Anda Akun AWS.



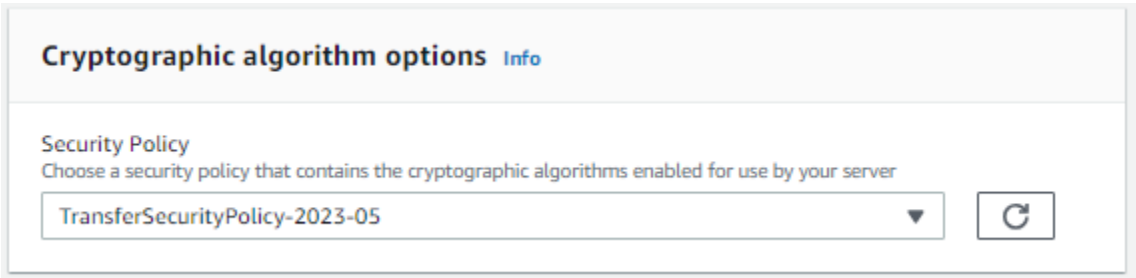
Jika Anda memilih Buat grup log, CloudWatch konsol (https://console.aws.amazon.com/cloudwatch/) terbuka ke halaman Buat grup log. Untuk detailnya, lihat Membuat grup log di CloudWatch Log.

b. (Opsional) Untuk alur kerja Terkelola, pilih ID alur kerja (dan peran terkait) yang harus diasumsikan oleh Transfer Family saat menjalankan alur kerja. Anda dapat memilih satu alur kerja untuk dieksekusi setelah unggahan lengkap, dan satu lagi untuk mengeksekusi pada unggahan sebagian. Untuk mempelajari lebih lanjut tentang memproses file Anda menggunakan alur kerja terkelola, lihatAWS Transfer Family alur kerja terkelola.



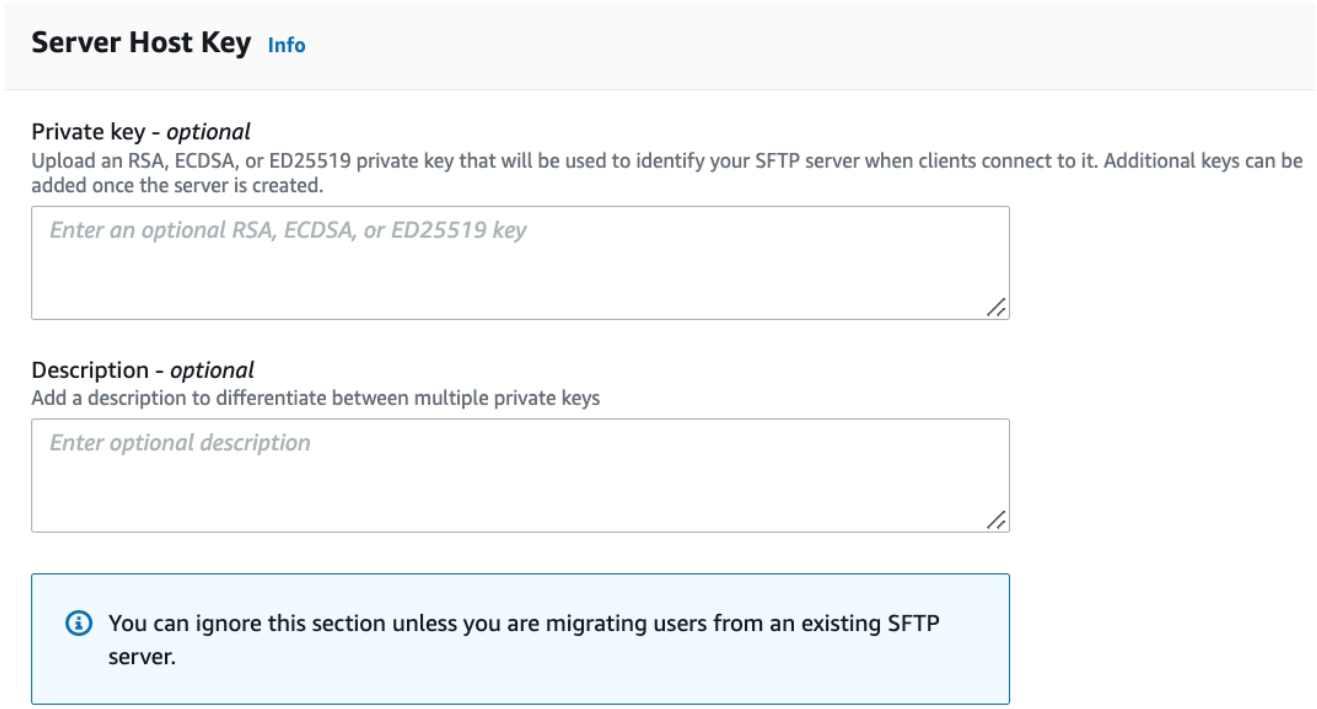
c. Untuk opsi algoritma kriptografi, pilih kebijakan keamanan yang berisi algoritma kriptografi yang diaktifkan untuk digunakan oleh server Anda.

> **Note**
>
> Transfer Family menetapkan kebijakan keamanan terbaru ke server FTP Anda. Namun, karena protokol FTP tidak menggunakan enkripsi apa pun, server FTP tidak menggunakan algoritma kebijakan keamanan apa pun. Kecuali server Anda juga menggunakan protokol FTPS atau SFTP, kebijakan keamanan tetap tidak digunakan.



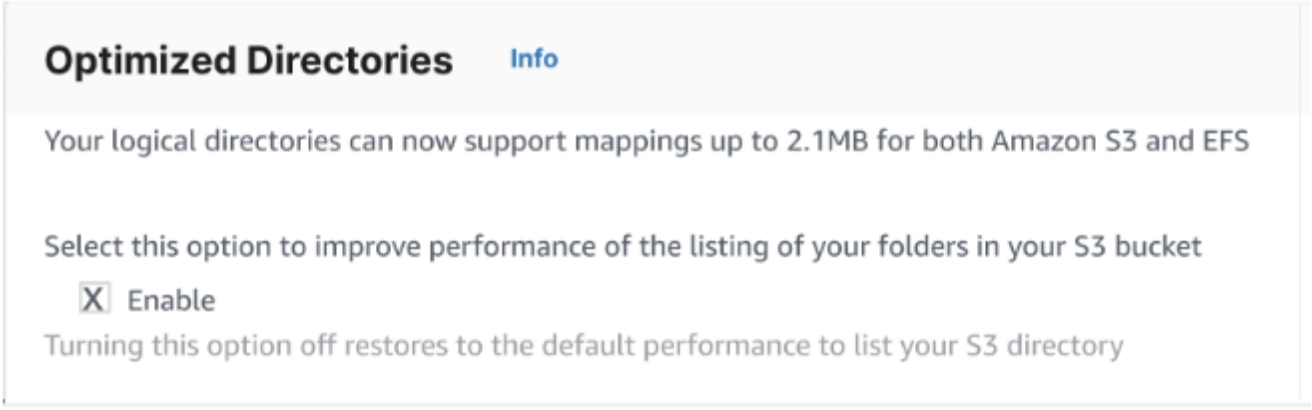
d. Untuk Server Host Key, biarkan kosong.

> **Note**
>
> Bagian Server Host Key hanya digunakan untuk memigrasikan pengguna dari server berkemampuan SFTP yang ada.

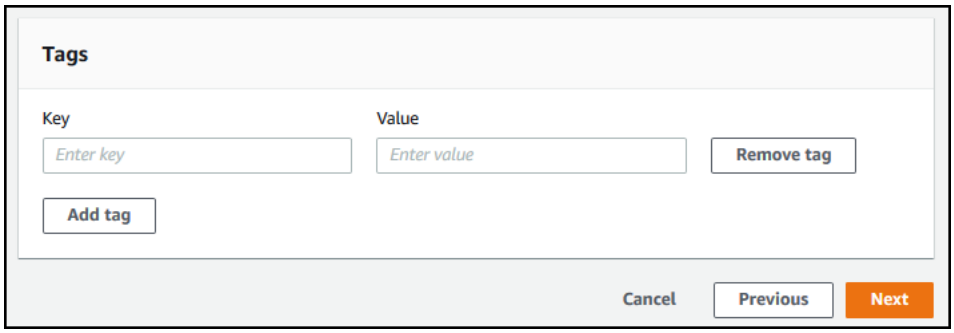


e. (Opsional) Untuk Tag, untuk Kunci dan Nilai, masukkan satu atau beberapa tag sebagai pasangan nilai kunci, lalu pilih Tambahkan tag.

f. Anda dapat mengoptimalkan kinerja untuk direktori Amazon S3 Anda. Misalnya, Anda masuk ke direktori home Anda, dan Anda memiliki 10.000 subdirektori. Dengan kata lain, bucket S3 Anda memiliki 10.000 folder. Dalam skenario ini, jika Anda menjalankan perintah `ls` (daftar), operasi daftar memakan waktu antara enam dan delapan menit. Namun, jika Anda mengoptimalkan direktori Anda, operasi ini hanya membutuhkan beberapa detik.

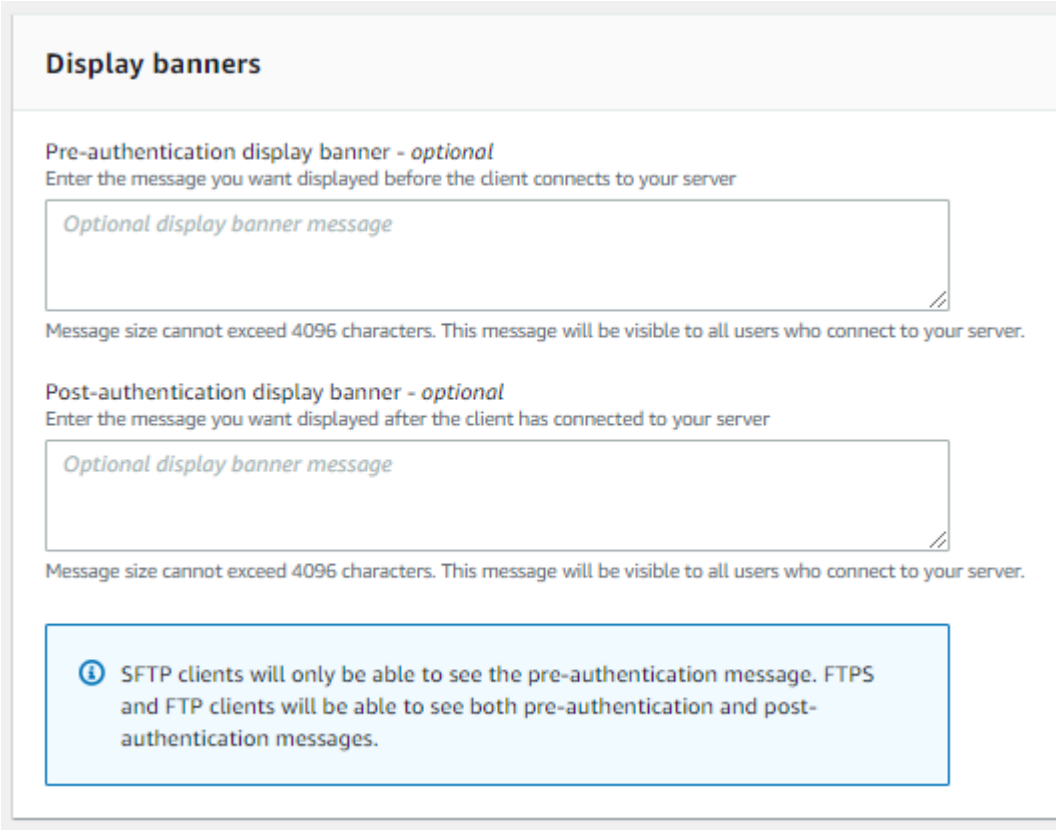


g. Pilih Berikutnya.



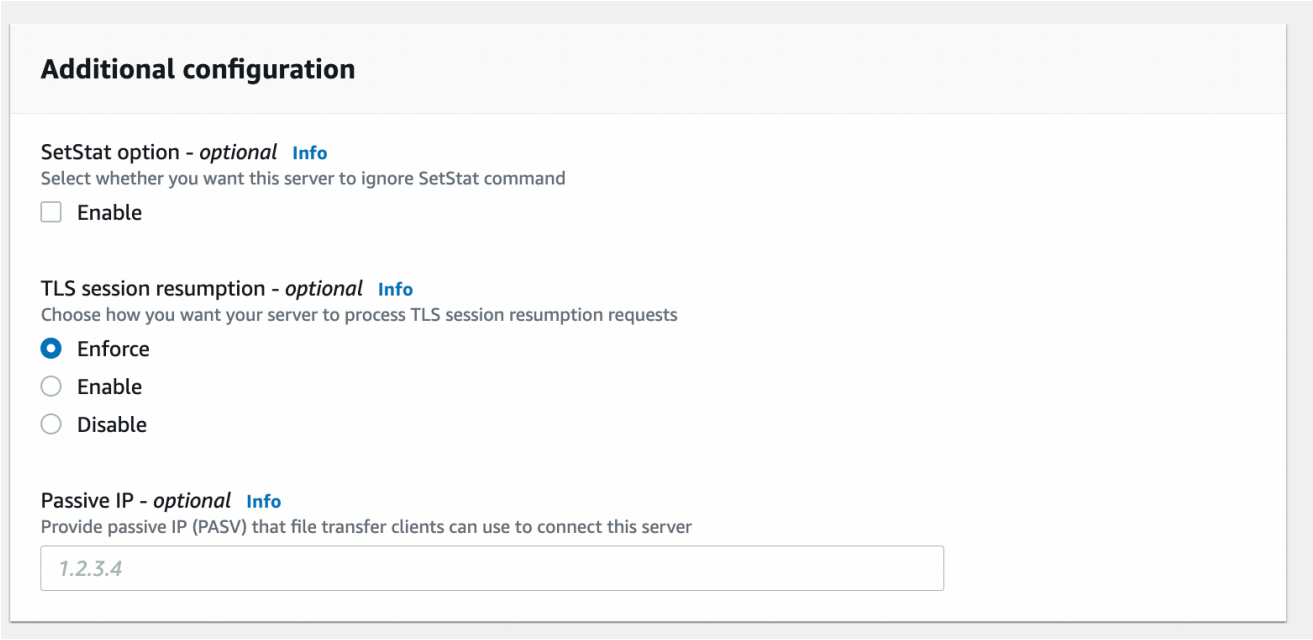
h. (Opsional) Anda dapat mengonfigurasi AWS Transfer Family server untuk menampilkan pesan yang disesuaikan seperti kebijakan organisasi atau syarat dan ketentuan kepada pengguna akhir Anda. Anda juga dapat menampilkan Message of The Day (MOTD) yang disesuaikan kepada pengguna yang telah berhasil diautentikasi.

   Untuk spanduk Tampilan, di kotak teks spanduk tampilan Pra-otentikasi, masukkan pesan teks yang ingin ditampilkan kepada pengguna Anda sebelum mereka mengautentikasi, dan di kotak teks spanduk tampilan pasca-otentikasi, masukkan teks yang ingin ditampilkan kepada pengguna Anda setelah mereka berhasil mengautentikasi.



i. (Opsional) Anda dapat mengonfigurasi opsi tambahan berikut.

- SetStat opsi: aktifkan opsi ini untuk mengabaikan kesalahan yang dihasilkan saat klien mencoba menggunakan `SETSTAT` pada file yang Anda unggah ke bucket Amazon S3. Untuk detail tambahan, lihat `SetStatOption` dokumentasi dalam ProtocolDetailstopik.
- Dimulainya kembali sesi TLS: menyediakan mekanisme untuk melanjutkan atau berbagi kunci rahasia yang dinegosiasikan antara kontrol dan koneksi data untuk sesi FTPS. Untuk detail tambahan, lihat `TlsSessionResumptionMode` dokumentasi dalam ProtocolDetailstopik.
- IP pasif: menunjukkan mode pasif, untuk protokol FTP dan FTPS. Masukkan satu alamat IPv4, seperti alamat IP publik firewall, router, atau penyeimbang beban. Untuk detail tambahan, lihat `PassiveIp` dokumentasi dalam ProtocolDetailstopik.



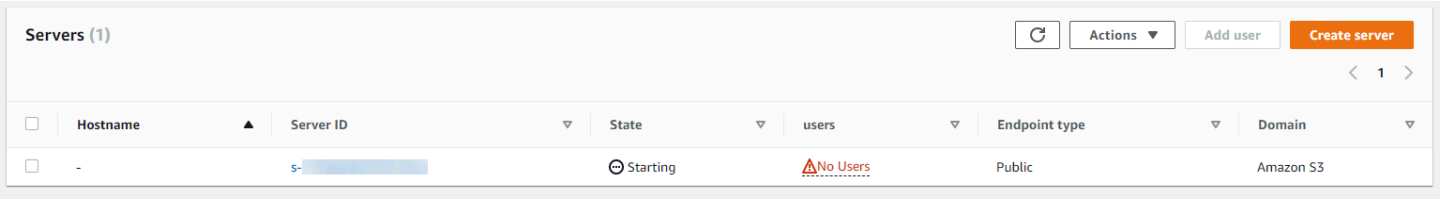
8. Di Tinjau dan buat, tinjau pilihan Anda.

   - Jika Anda ingin mengedit salah satu dari mereka, pilih Edit di sebelah langkah.

     > **Note**
     >
     > Anda harus meninjau setiap langkah setelah langkah yang Anda pilih untuk diedit.

   - Jika Anda tidak memiliki perubahan, pilih Buat server untuk membuat server Anda. Anda dibawa ke halaman Server, ditampilkan berikut, di mana server baru Anda terdaftar.

Diperlukan beberapa menit sebelum status server baru Anda berubah menjadi Online. Pada saat itu, server Anda dapat melakukan operasi file untuk pengguna Anda.



Langkah selanjutnya — Untuk langkah selanjutnya, lanjutkan Bekerja dengan penyedia identitas khusus untuk mengatur pengguna.

Buat server di cloud pribadi virtual

Anda dapat meng-host endpoint server Anda di dalam virtual private cloud (VPC) untuk digunakan untuk mentransfer data ke dan dari bucket Amazon S3 atau sistem file Amazon EFS tanpa melalui internet publik.

> **Note**
>
> Setelah 19 Mei 2021, Anda tidak akan dapat membuat server menggunakan `EndpointType=VPC_ENDPOINT` di AWS akun Anda jika akun Anda belum melakukannya sebelum 19 Mei 2021. Jika Anda telah membuat server dengan `EndpointType=VPC_ENDPOINT` di AWS akun Anda pada atau sebelum 21 Februari 2021, Anda tidak akan terpengaruh. Setelah tanggal ini, gunakan `EndpointType` **=VPC**. Untuk informasi selengkapnya, lihat the section called “Menghentikan penggunaan VPC_ENDPOINT”.

Jika Anda menggunakan Amazon Virtual Private Cloud (Amazon VPC) untuk meng-host AWS sumber daya Anda, Anda dapat membuat koneksi pribadi antara VPC dan server. Anda kemudian dapat menggunakan server ini untuk mentransfer data melalui klien Anda ke dan dari bucket Amazon S3 Anda tanpa menggunakan alamat IP publik atau memerlukan gateway internet.

Menggunakan Amazon VPC, Anda dapat meluncurkan AWS sumber daya di jaringan virtual khusus. Anda dapat menggunakan VPC untuk mengendalikan pengaturan jaringan, seperti rentang alamat IP, subnet, tabel rute, dan gateway jaringan. Untuk informasi selengkapnya tentang VPC, lihat Apa itu Amazon VPC? di Panduan Pengguna Amazon VPC.

Di bagian selanjutnya, temukan petunjuk tentang cara membuat dan menghubungkan VPC Anda ke server. Sebagai ikhtisar, Anda melakukan ini sebagai berikut:

1. Siapkan server menggunakan titik akhir VPC.
2. Connect ke server Anda menggunakan klien yang ada di dalam VPC Anda melalui titik akhir VPC. Melakukan hal ini memungkinkan Anda untuk mentransfer data yang disimpan di bucket Amazon S3 Anda melalui klien Anda menggunakan. AWS Transfer Family Anda dapat melakukan transfer ini meskipun jaringan terputus dari internet publik.
3. Selain itu, jika Anda memilih untuk membuat titik akhir server Anda menghadap ke internet, Anda dapat mengaitkan alamat IP Elastic dengan titik akhir Anda. Melakukan hal ini memungkinkan

klien di luar VPC Anda terhubung ke server Anda. Anda dapat menggunakan grup keamanan VPC untuk mengontrol akses ke pengguna yang diautentikasi yang permintaannya hanya berasal dari alamat yang diizinkan.

Topik

- Buat endpoint server yang hanya dapat diakses dalam VPC Anda
- Buat titik akhir yang menghadap ke internet untuk server Anda
- Ubah tipe endpoint untuk server Anda
- Menghentikan penggunaan VPC_ENDPOINT
- Memperbarui tipe endpoint AWS Transfer Family server dari VPC_ENDPOINT ke VPC

Buat endpoint server yang hanya dapat diakses dalam VPC Anda

Dalam prosedur berikut, Anda membuat endpoint server yang hanya dapat diakses oleh sumber daya dalam VPC Anda.

Untuk membuat endpoint server di dalam VPC

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Dari panel navigasi, pilih Server, lalu pilih Buat server.
3. Di Pilih protokol, pilih satu atau beberapa protokol, lalu pilih Berikutnya. Untuk informasi lebih lanjut tentang protokol, lihat. Langkah 2: Buat server berkemampuan SFTP
4. Di Pilih penyedia identitas, pilih Layanan yang dikelola untuk menyimpan identitas dan kunci penggunaAWS Transfer Family, lalu pilih Berikutnya.

   > **Note**
   >
   > Prosedur ini menggunakan opsi yang dikelola layanan. Jika Anda memilih Kustom, Anda menyediakan titik akhir Amazon API Gateway dan peran AWS Identity and Access Management (IAM) untuk mengakses titik akhir. Dengan demikian, Anda dapat mengintegrasikan layanan direktori Anda untuk mengautentikasi dan mengotorisasi pengguna Anda. Untuk mempelajari lebih lanjut tentang bekerja dengan penyedia identitas kustom, lihatBekerja dengan penyedia identitas khusus.

5. Di Pilih titik akhir, lakukan hal berikut:

> **Note**
>
> Server FTP dan FTPS untuk Transfer Family beroperasi melalui Port 21 (Control Channel) dan Port Range 8192-8200 (Saluran Data).

a. Untuk tipe Endpoint, pilih tipe endpoint yang dihosting VPC untuk meng-host endpoint server Anda.

b. Untuk Access, pilih Internal untuk membuat endpoint Anda hanya dapat diakses oleh klien menggunakan alamat IP pribadi endpoint.

> **Note**
>
> Untuk detail tentang opsi Menghadapi Internet, lihatBuat titik akhir yang menghadap ke internet untuk server Anda. Server yang dibuat dalam VPC untuk akses internal saja tidak mendukung nama host khusus.

c. Untuk VPC, pilih ID VPC yang ada atau pilih Buat VPC untuk membuat VPC baru.

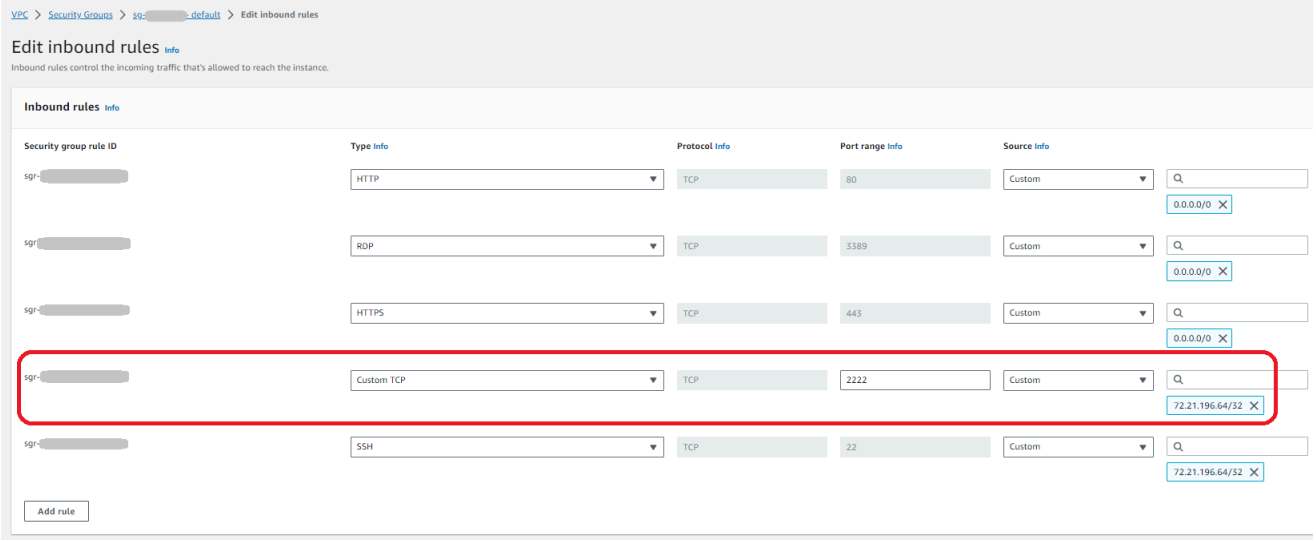
d. Di bagian Availability Zones, pilih hingga tiga Availability Zones dan subnet terkait.

e. Di bagian Grup Keamanan, pilih ID atau ID grup keamanan yang ada atau pilih Buat grup keamanan untuk membuat grup keamanan baru. Untuk informasi selengkapnya tentang grup keamanan, lihat Grup keamanan untuk VPC Anda di Panduan Pengguna Amazon Virtual Private Cloud. Untuk membuat grup keamanan, lihat Membuat grup keamanan di Panduan Pengguna Amazon Virtual Private Cloud.

> **Note**
>
> VPC Anda secara otomatis dilengkapi dengan grup keamanan default. Jika Anda tidak menentukan grup atau grup keamanan yang berbeda saat meluncurkan server, kami mengaitkan grup keamanan default dengan server Anda.

Untuk aturan masuk untuk grup keamanan, Anda dapat mengonfigurasi lalu lintas SSH untuk menggunakan port 22, 2222, atau keduanya. Port 22 dikonfigurasi secara default. Untuk menggunakan port 2222, Anda menambahkan aturan masuk ke grup keamanan Anda. Untuk jenisnya, pilih TCP Kustom, lalu masukkan **2222** untuk rentang Port, dan untuk

sumbernya, masukkan rentang CIDR yang sama dengan yang Anda miliki untuk aturan port SSH 22 Anda.



f. (Opsional) Untuk FIPS Enabled, pilih kotak centang titik akhir FIPS Enabled untuk memastikan titik akhir sesuai dengan Federal Information Processing Standards (FIPS).

> **Note**
>
> Titik akhir berkemampuan FIPS hanya tersedia di Wilayah Amerika Utara. AWS Untuk Wilayah yang tersedia, lihat AWS Transfer Familytitik akhir dan kuota di. Referensi Umum AWS Untuk informasi lebih lanjut tentang FIPS, lihat Federal Information Processing Standard (FIPS) 140-2.

g. Pilih Berikutnya.

6. Di Konfigurasikan detail tambahan, lakukan hal berikut:

a. Untuk CloudWatch logging, pilih salah satu dari berikut ini untuk mengaktifkan CloudWatch pencatatan Amazon dari aktivitas pengguna Anda:

- Buat peran baru untuk memungkinkan Transfer Family membuat peran IAM secara otomatis, selama Anda memiliki izin yang tepat untuk membuat peran baru. Peran IAM yang dibuat disebut`AWSTransferLoggingAccess`.
- Pilih peran yang ada untuk memilih peran IAM yang ada dari akun Anda. Di bawah Peran logging, pilih peran. Peran IAM ini harus menyertakan kebijakan kepercayaan dengan Layanan yang disetel `ketransfer.amazonaws.com`.

Untuk informasi selengkapnya tentang CloudWatch pencatatan, lihatKonfigurasikan peran CloudWatch logging.

> **Note**
>
> - Anda tidak dapat melihat aktivitas pengguna akhir CloudWatch jika Anda tidak menentukan peran logging.
> - Jika Anda tidak ingin menyiapkan peran CloudWatch logging, pilih Pilih peran yang ada, tetapi jangan pilih peran logging.

b. Untuk opsi algoritma kriptografi, pilih kebijakan keamanan yang berisi algoritma kriptografi yang diaktifkan untuk digunakan oleh server Anda.

> **Note**
>
> Secara default, kebijakan `TransferSecurityPolicy-2020-06` keamanan dilampirkan ke server Anda kecuali Anda memilih yang berbeda.

Untuk informasi selengkapnya tentang kebijakan keamanan, lihatKebijakan keamanan untuk AWS Transfer Family server.

c. (Opsional) Untuk Server Host Key, masukkan kunci pribadi RSA, ED25519, atau ECDSA yang akan digunakan untuk mengidentifikasi server Anda ketika klien terhubung ke sana melalui SFTP.

> **Note**
>
> Bagian ini hanya untuk memigrasi pengguna dari server berkemampuan SFTP yang ada.

d. (Opsional) Untuk Tag, untuk Kunci dan Nilai, masukkan satu atau beberapa tag sebagai pasangan nilai kunci, lalu pilih Tambahkan tag.

e. Pilih Berikutnya.

7. Di Tinjau dan buat, tinjau pilihan Anda. Jika Anda:

- Ingin mengedit salah satu dari mereka, pilih Edit di sebelah langkah.

  > **Note**
  >
  > Anda perlu meninjau setiap langkah setelah langkah yang Anda pilih untuk diedit.

- Tidak ada perubahan, pilih Buat server untuk membuat server Anda. Anda dibawa ke halaman Server, ditampilkan berikut, di mana server baru Anda terdaftar.

Diperlukan beberapa menit sebelum status server baru Anda berubah menjadi Online. Pada saat itu, server Anda dapat melakukan operasi file untuk pengguna Anda.

Buat titik akhir yang menghadap ke internet untuk server Anda

Dalam prosedur berikut, Anda membuat endpoint server. Titik akhir ini dapat diakses melalui internet hanya untuk klien yang alamat IP sumbernya diizinkan di grup keamanan default VPC Anda. Selain itu, dengan menggunakan alamat IP Elastic untuk membuat titik akhir Anda menghadap ke internet, klien Anda dapat menggunakan alamat IP Elastic untuk memungkinkan akses ke titik akhir Anda di firewall mereka.

> **Note**
>
> Hanya SFTP dan FTPS yang dapat digunakan pada titik akhir yang dihosting VPC yang menghadap ke internet.

Untuk membuat titik akhir yang menghadap ke internet

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Dari panel navigasi, pilih Server, lalu pilih Buat server.
3. Di Pilih protokol, pilih satu atau beberapa protokol, lalu pilih Berikutnya. Untuk informasi lebih lanjut tentang protokol, lihat. Langkah 2: Buat server berkemampuan SFTP
4. Di Pilih penyedia identitas, pilih Layanan yang dikelola untuk menyimpan identitas dan kunci penggunaAWS Transfer Family, lalu pilih Berikutnya.

> **Note**
>
> Prosedur ini menggunakan opsi yang dikelola layanan. Jika Anda memilih Kustom, Anda menyediakan titik akhir Amazon API Gateway dan peran AWS Identity and Access Management (IAM) untuk mengakses titik akhir. Dengan demikian, Anda dapat mengintegrasikan layanan direktori Anda untuk mengautentikasi dan mengotorisasi pengguna Anda. Untuk mempelajari lebih lanjut tentang bekerja dengan penyedia identitas kustom, lihatBekerja dengan penyedia identitas khusus.

5. Di Pilih titik akhir, lakukan hal berikut:

   a. Untuk tipe Endpoint, pilih tipe endpoint yang dihosting VPC untuk meng-host endpoint server Anda.

   b. Untuk Akses, pilih Internet Facing untuk membuat titik akhir Anda dapat diakses oleh klien melalui internet.

   > **Note**
   >
   > Ketika Anda memilih Internet Facing, Anda dapat memilih alamat IP Elastis yang ada di setiap subnet atau subnet. Atau Anda dapat pergi ke konsol VPC (https://console.aws.amazon.com/vpc/) untuk mengalokasikan satu atau lebih alamat IP elastis baru. Alamat ini dapat dimiliki oleh AWS atau oleh Anda. Anda tidak dapat mengaitkan alamat IP Elastis yang sudah digunakan dengan titik akhir Anda.

   c. (Opsional) Untuk nama host Kustom, pilih salah satu dari berikut ini:

   > **Note**
   >
   > Pelanggan AWS GovCloud (US) perlu terhubung melalui alamat IP Elastic secara langsung, atau membuat catatan nama host dalam Commercial Route 53 yang mengarah ke EIP mereka. Untuk informasi selengkapnya tentang menggunakan Route 53 untuk GovCloud titik akhir, lihat Menyiapkan Amazon Route 53 dengan AWS GovCloud (US) sumber daya Anda di Panduan AWS GovCloud (US) Pengguna.

- Amazon Route 53 Alias DNS — jika nama host yang ingin Anda gunakan terdaftar di Route 53. Anda kemudian dapat memasukkan nama host.
- DNS lainnya — jika nama host yang ingin Anda gunakan terdaftar dengan penyedia DNS lain. Anda kemudian dapat memasukkan nama host.
- Tidak ada - untuk menggunakan titik akhir server dan tidak menggunakan nama host khusus. Nama host server mengambil formulir`server-id`.server.transfer.`region`.amazonaws.com.

> **Note**
>
> Untuk pelanggan diAWS GovCloud (US), memilih None tidak membuat nama host dalam format ini.

Untuk mempelajari lebih lanjut tentang bekerja dengan nama host kustom, lihatBekerja dengan nama host khusus.

d. Untuk VPC, pilih ID VPC yang ada atau pilih Buat VPC untuk membuat VPC baru.

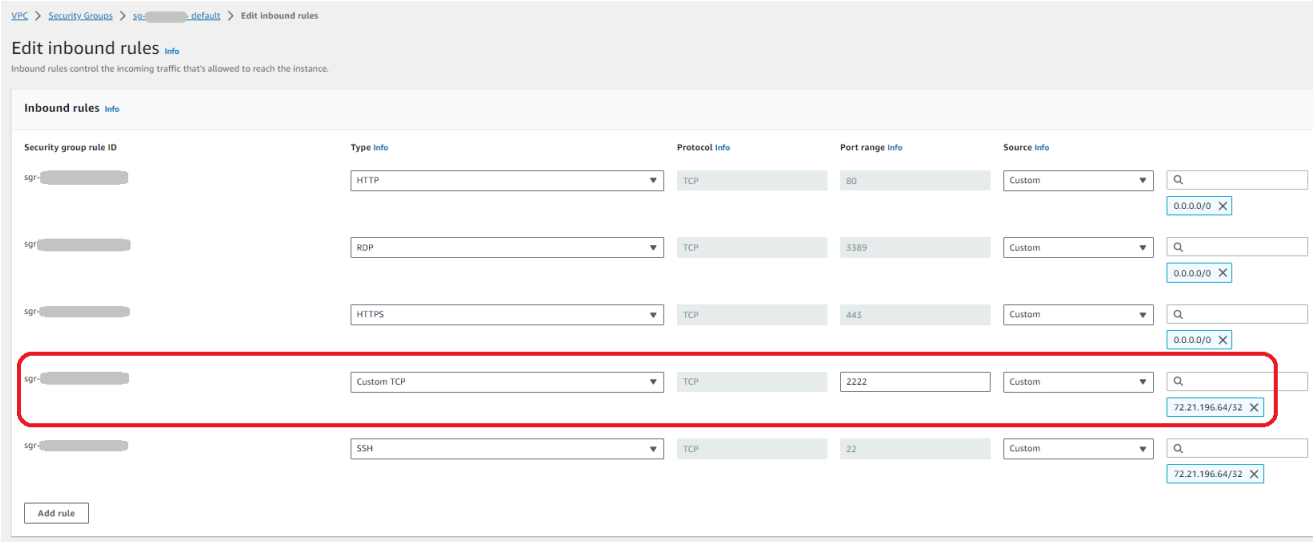
e. Di bagian Availability Zones, pilih hingga tiga Availability Zones dan subnet terkait. Untuk Alamat IPv4, pilih alamat IP Elastis untuk setiap subnet. Ini adalah alamat IP yang dapat digunakan klien Anda untuk memungkinkan akses ke titik akhir Anda di firewall mereka.

f. Di bagian Grup Keamanan, pilih ID atau ID grup keamanan yang ada atau pilih Buat grup keamanan untuk membuat grup keamanan baru. Untuk informasi selengkapnya tentang grup keamanan, lihat Grup keamanan untuk VPC Anda di Panduan Pengguna Amazon Virtual Private Cloud. Untuk membuat grup keamanan, lihat Membuat grup keamanan di Panduan Pengguna Amazon Virtual Private Cloud.

> **Note**
>
> VPC Anda secara otomatis dilengkapi dengan grup keamanan default. Jika Anda tidak menentukan grup atau grup keamanan yang berbeda saat meluncurkan server, kami mengaitkan grup keamanan default dengan server Anda.

Untuk aturan masuk untuk grup keamanan, Anda dapat mengonfigurasi lalu lintas SSH untuk menggunakan port 22, 2222, atau keduanya. Port 22 dikonfigurasi secara default.

Untuk menggunakan port 2222, Anda menambahkan aturan masuk ke grup keamanan Anda. Untuk jenisnya, pilih TCP Kustom, lalu masukkan **2222** untuk rentang Port, dan untuk sumbernya, masukkan rentang CIDR yang sama dengan yang Anda miliki untuk aturan port SSH 22 Anda.



g. (Opsional) Untuk FIPS Enabled, pilih kotak centang titik akhir FIPS Enabled untuk memastikan titik akhir sesuai dengan Federal Information Processing Standards (FIPS).

> **Note**
>
> Titik akhir berkemampuan FIPS hanya tersedia di Wilayah Amerika Utara. AWS Untuk Wilayah yang tersedia, lihat AWS Transfer Familytitik akhir dan kuota di. Referensi Umum AWS Untuk informasi lebih lanjut tentang FIPS, lihat Federal Information Processing Standard (FIPS) 140-2.

h. Pilih Berikutnya.

6. Di Konfigurasikan detail tambahan, lakukan hal berikut:

a. Untuk CloudWatch logging, pilih salah satu dari berikut ini untuk mengaktifkan CloudWatch pencatatan Amazon dari aktivitas pengguna Anda:

- Buat peran baru untuk memungkinkan Transfer Family membuat peran IAM secara otomatis, selama Anda memiliki izin yang tepat untuk membuat peran baru. Peran IAM yang dibuat disebut`AWSTransferLoggingAccess`.

- Pilih peran yang ada untuk memilih peran IAM yang ada dari akun Anda. Di bawah Peran logging, pilih peran. Peran IAM ini harus menyertakan kebijakan kepercayaan dengan Layanan yang disetel `ketransfer.amazonaws.com`.

  Untuk informasi selengkapnya tentang CloudWatch pencatatan, lihatKonfigurasikan peran CloudWatch logging.

> **Note**
>
> - Anda tidak dapat melihat aktivitas pengguna akhir CloudWatch jika Anda tidak menentukan peran logging.
> - Jika Anda tidak ingin menyiapkan peran CloudWatch logging, pilih Pilih peran yang ada, tetapi jangan pilih peran logging.

b. Untuk opsi algoritma kriptografi, pilih kebijakan keamanan yang berisi algoritma kriptografi yang diaktifkan untuk digunakan oleh server Anda.

> **Note**
>
> Secara default, kebijakan `TransferSecurityPolicy-2020-06` keamanan dilampirkan ke server Anda kecuali Anda memilih yang berbeda.

Untuk informasi selengkapnya tentang kebijakan keamanan, lihatKebijakan keamanan untuk AWS Transfer Family server.

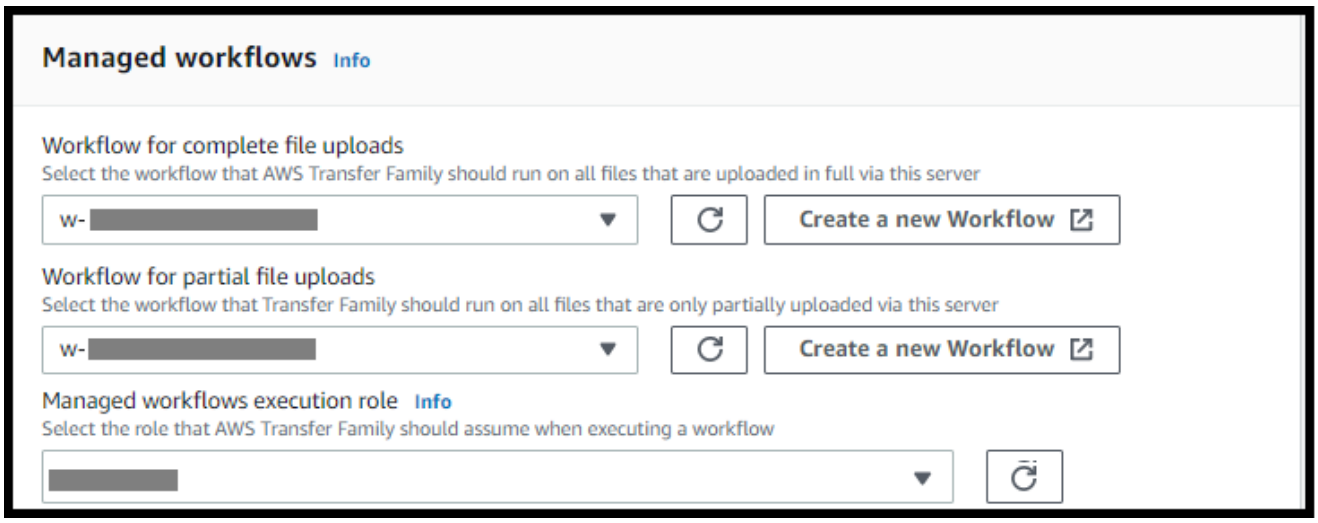
c. (Opsional) Untuk Server Host Key, masukkan kunci pribadi RSA, ED25519, atau ECDSA yang akan digunakan untuk mengidentifikasi server Anda ketika klien terhubung ke sana melalui SFTP.

> **Note**
>
> Bagian ini hanya untuk memigrasi pengguna dari server berkemampuan SFTP yang ada.

d. (Opsional) Untuk Tag, untuk Kunci dan Nilai, masukkan satu atau beberapa tag sebagai pasangan nilai kunci, lalu pilih Tambahkan tag.

e. Pilih Berikutnya.

f. (Opsional) Untuk alur kerja Terkelola, pilih ID alur kerja (dan peran terkait) yang harus diasumsikan oleh Transfer Family saat menjalankan alur kerja. Anda dapat memilih satu alur kerja untuk dieksekusi setelah unggahan lengkap, dan satu lagi untuk mengeksekusi pada unggahan sebagian. Untuk mempelajari lebih lanjut tentang memproses file Anda menggunakan alur kerja terkelola, lihatAWS Transfer Family alur kerja terkelola.



7. Di Tinjau dan buat, tinjau pilihan Anda. Jika Anda:

- Ingin mengedit salah satu dari mereka, pilih Edit di sebelah langkah.

> **Note**
>
> Anda perlu meninjau setiap langkah setelah langkah yang Anda pilih untuk diedit.

- Tidak ada perubahan, pilih Buat server untuk membuat server Anda. Anda dibawa ke halaman Server, ditampilkan berikut, di mana server baru Anda terdaftar.

Anda dapat memilih ID server untuk melihat pengaturan rinci dari server yang baru saja Anda buat. Setelah kolom alamat IPv4 Publik diisi, alamat IP Elastis yang Anda berikan berhasil dikaitkan dengan titik akhir server Anda.

> **Note**
>
> Ketika server Anda di VPC sedang online, hanya subnet yang dapat dimodifikasi dan hanya melalui API. UpdateServer Anda harus menghentikan server untuk menambah atau mengubah alamat IP Elastis titik akhir server.

Ubah tipe endpoint untuk server Anda

Jika Anda memiliki server yang sudah ada yang dapat diakses melalui internet (yaitu, memiliki tipe titik akhir publik), Anda dapat mengubah titik akhir ke titik akhir VPC.

> **Note**
>
> Jika Anda memiliki server yang ada di VPC yang ditampilkan sebagai`VPC_ENDPOINT`, kami sarankan Anda memodifikasinya ke jenis titik akhir VPC yang baru. Dengan tipe endpoint baru ini, Anda tidak perlu lagi menggunakan Network Load Balancer (NLB) untuk mengaitkan alamat IP Elastic dengan endpoint server Anda. Selain itu, Anda dapat menggunakan grup keamanan VPC untuk membatasi akses ke titik akhir server Anda. Namun, Anda dapat terus menggunakan tipe `VPC_ENDPOINT` endpoint sesuai kebutuhan.

Prosedur berikut mengasumsikan bahwa Anda memiliki server yang menggunakan tipe endpoint publik saat ini atau tipe yang lebih lama`VPC_ENDPOINT`.

Untuk mengubah tipe endpoint untuk server Anda

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi, pilih Server.
3. Pilih kotak centang server yang ingin Anda ubah tipe endpoint.

   > **Important**
   >
   > Anda harus menghentikan server sebelum Anda dapat mengubah titik akhirnya.

4. Untuk Tindakan, pilih Berhenti.
5. Di kotak dialog konfirmasi yang muncul, pilih Berhenti untuk mengonfirmasi bahwa Anda ingin menghentikan server.

> **Note**
>
> Sebelum melanjutkan ke langkah berikutnya, di detail Endpoint, tunggu Status server berubah menjadi Offline; ini bisa memakan waktu beberapa menit. Anda mungkin harus memilih Refresh di halaman Server untuk melihat perubahan status.
> Anda tidak akan dapat melakukan pengeditan apa pun sampai server Offline.

6. Di detail Endpoint, pilih Edit.
7. Dalam konfigurasi Edit titik akhir, lakukan hal berikut:

   a. Untuk jenis Edit titik akhir, pilih VPC yang dihosting.

   b. Untuk Access, pilih salah satu dari berikut ini:

   - Internal untuk membuat titik akhir Anda hanya dapat diakses oleh klien menggunakan alamat IP pribadi titik akhir.
   - Menghadapi Internet untuk membuat titik akhir Anda dapat diakses oleh klien melalui internet publik.

   > **Note**
   >
   > Ketika Anda memilih Internet Facing, Anda dapat memilih alamat IP Elastis yang ada di setiap subnet atau subnet. Atau, Anda dapat pergi ke konsol VPC (https://console.aws.amazon.com/vpc/) untuk mengalokasikan satu atau lebih alamat IP elastis baru. Alamat ini dapat dimiliki oleh AWS atau oleh Anda. Anda tidak dapat mengaitkan alamat IP Elastis yang sudah digunakan dengan titik akhir Anda.

   c. (Opsional untuk akses yang menghadap internet saja) Untuk nama host Kustom, pilih salah satu dari berikut ini:

   - Amazon Route 53 Alias DNS — jika nama host yang ingin Anda gunakan terdaftar di Route 53. Anda kemudian dapat memasukkan nama host.
   - DNS lainnya — jika nama host yang ingin Anda gunakan terdaftar dengan penyedia DNS lain. Anda kemudian dapat memasukkan nama host.
   - Tidak ada - untuk menggunakan titik akhir server dan tidak menggunakan nama host khusus. Nama host server mengambil formulir`serverId.server.transfer.regionId.amazonaws.com`.

Untuk mempelajari lebih lanjut tentang bekerja dengan nama host kustom, lihatBekerja dengan nama host khusus.

d. Untuk VPC, pilih ID VPC yang ada, atau pilih Buat VPC untuk membuat VPC baru.

e. Di bagian Availability Zones, pilih hingga tiga Availability Zones dan subnet terkait. Jika Internet Facing dipilih, pilih juga alamat IP Elastis untuk setiap subnet.

> **Note**
>
> Jika Anda menginginkan maksimum tiga Availability Zone, tetapi tidak cukup tersedia, buat di konsol VPC (https://console.aws.amazon.com/vpc/).
> Jika Anda memodifikasi subnet atau alamat IP Elastis, server membutuhkan beberapa menit untuk memperbarui. Anda tidak dapat menyimpan perubahan Anda sampai pembaruan server selesai.

f. Pilih Simpan.

8. Untuk Tindakan, pilih Mulai dan tunggu status server berubah menjadi Online; ini bisa memakan waktu beberapa menit.

> **Note**
>
> Jika Anda mengubah tipe titik akhir publik menjadi tipe titik akhir VPC, perhatikan bahwa tipe Endpoint untuk server Anda telah berubah menjadi VPC.

Grup keamanan default dilampirkan ke titik akhir. Untuk mengubah atau menambahkan grup keamanan tambahan, lihat Membuat Grup Keamanan.

Menghentikan penggunaan VPC_ENDPOINT

AWS Transfer Familymenghentikan kemampuan untuk membuat server dengan `EndpointType=VPC_ENDPOINT` untuk AWS akun baru. Per 19 Mei 2021, AWS akun yang tidak memiliki AWS Transfer Family server dengan tipe titik akhir tidak `VPC_ENDPOINT` akan dapat membuat server baru. `EndpointType=VPC_ENDPOINT` Jika Anda sudah memiliki server yang menggunakan tipe `VPC_ENDPOINT` endpoint, kami sarankan Anda mulai menggunakan `EndpointType=VPC` sesegera mungkin. Untuk detailnya, lihat Memperbarui jenis titik akhir AWS Transfer Family server Anda dari VPC_ENDPOINT ke VPC.

Kami meluncurkan tipe VPC endpoint baru di awal tahun 2020. Untuk informasi selengkapnya, lihat AWS Transfer Familyuntuk SFTP mendukung Grup Keamanan VPC dan alamat IP Elastis. Titik akhir baru ini lebih kaya fitur dan hemat biaya dan tidak ada PrivateLink biaya. Untuk informasi lebih lanjut, lihat AWS PrivateLink harga.

Tipe endpoint ini secara fungsional setara dengan tipe endpoint sebelumnya (). VPC_ENDPOINT Anda dapat melampirkan alamat IP Elastis langsung ke titik akhir untuk membuatnya menghadap internet dan menggunakan grup keamanan untuk penyaringan IP sumber. Untuk informasi selengkapnya, lihat daftar Use IP allow untuk mengamankan postingan blog server SFTP Anda AWS Transfer Family.

Anda juga dapat meng-host titik akhir ini di lingkungan VPC bersama. Untuk informasi selengkapnya, lihat AWS Transfer Familysekarang mendukung lingkungan VPC layanan bersama.

Selain SFTP, Anda dapat menggunakan VPC EndpointType untuk mengaktifkan FTPS dan FTP. Kami tidak berencana untuk menambahkan fitur ini dan dukungan FTPS/FTP. EndpointType=VPC_ENDPOINT Kami juga telah menghapus jenis titik akhir ini sebagai opsi dari AWS Transfer Family konsol.

Anda dapat mengubah jenis endpoint untuk server menggunakan konsol Transfer Family, APIAWS CLI, SDK, atau. AWS CloudFormation Untuk mengubah jenis endpoint server Anda, lihatMemperbarui tipe endpoint AWS Transfer Family server dari VPC_ENDPOINT ke VPC.

Jika Anda memiliki pertanyaan, hubungi AWS Support atau tim AWS akun Anda.

> **Note**
>
> Kami tidak berencana untuk menambahkan fitur ini dan dukungan FTPS atau FTP ke = VPC_ENDPOINT. EndpointType Kami tidak lagi menawarkannya sebagai opsi di AWS Transfer Family Konsol.

Jika Anda memiliki pertanyaan tambahan, Anda dapat menghubungi kami melalui AWS Support atau tim akun Anda.

Memperbarui tipe endpoint AWS Transfer Family server dari VPC_ENDPOINT ke VPC

Anda dapat menggunakanAWS Management Console,AWS CloudFormation, atau Transfer Family API untuk memperbarui server EndpointType dari VPC_ENDPOINT keVPC. Prosedur dan contoh terperinci untuk menggunakan masing-masing metode ini untuk memperbarui jenis endpoint server

disediakan di bagian berikut. Jika Anda memiliki server di beberapa AWS wilayah dan di beberapa AWS akun, Anda dapat menggunakan contoh skrip yang disediakan di bagian berikut, dengan modifikasi, untuk mengidentifikasi server menggunakan `VPC_ENDPOINT` jenis yang perlu Anda perbarui.

Topik

- Mengidentifikasi server menggunakan tipe VPC_ENDPOINT endpoint
- Memperbarui tipe endpoint server menggunakan AWS Management Console
- Memperbarui tipe endpoint server menggunakan AWS CloudFormation
- Memperbarui server EndpointType menggunakan API

Mengidentifikasi server menggunakan tipe **`VPC_ENDPOINT`** endpoint

Anda dapat mengidentifikasi server mana yang `VPC_ENDPOINT` menggunakanAWS Management Console.

Untuk mengidentifikasi server menggunakan tipe **`VPC_ENDPOINT`** endpoint menggunakan konsol

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Pilih Server di panel navigasi untuk menampilkan daftar server di akun Anda di wilayah tersebut.
3. Urutkan daftar server berdasarkan jenis Endpoint untuk melihat semua server yang menggunakan`VPC_ENDPOINT`.

Untuk mengidentifikasi server yang menggunakan **`VPC_ENDPOINT`** berbagai AWS Wilayah dan akun

Jika Anda memiliki server di beberapa AWS wilayah dan di beberapa AWS akun, Anda dapat menggunakan contoh skrip berikut, dengan modifikasi, untuk mengidentifikasi server menggunakan tipe `VPC_ENDPOINT` endpoint. Skrip contoh menggunakan Amazon EC2 DescribeRegionsdan panggilan Transfer Family ListServers API untuk mendapatkan daftar ID server dan wilayah dari semua server yang Anda gunakan. `VPC_ENDPOINT` Jika Anda memiliki banyak AWS akun, Anda dapat melakukan loop melalui akun Anda menggunakan Peran IAM dengan akses auditor hanya baca jika Anda mengautentikasi menggunakan profil sesi ke penyedia identitas Anda.

1. Berikut ini adalah contoh sederhana.

```
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. Setelah Anda memiliki daftar server untuk diperbarui, Anda dapat menggunakan salah satu metode yang dijelaskan di bagian berikut untuk memperbarui `EndpointType` keVPC.

Memperbarui tipe endpoint server menggunakan AWS Management Console

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi, pilih Server.
3. Pilih kotak centang server yang ingin Anda ubah tipe endpoint.

> **Important**
>
> Anda harus menghentikan server sebelum Anda dapat mengubah titik akhirnya.

4. Untuk Tindakan, pilih Berhenti.
5. Di kotak dialog konfirmasi yang muncul, pilih Berhenti untuk mengonfirmasi bahwa Anda ingin menghentikan server.

> **Note**
>
> Sebelum melanjutkan ke langkah berikutnya, tunggu Status server berubah menjadi Offline; ini bisa memakan waktu beberapa menit. Anda mungkin harus memilih Refresh di halaman Server untuk melihat perubahan status.

6. Setelah status berubah menjadi Offline, pilih server untuk menampilkan halaman detail server.
7. Di bagian Detail titik akhir, pilih Edit.
8. Pilih VPC yang dihosting untuk tipe Endpoint.
9. Pilih Simpan
10. Untuk Tindakan, pilih Mulai dan tunggu status server berubah menjadi Online; ini bisa memakan waktu beberapa menit.

Memperbarui tipe endpoint server menggunakan AWS CloudFormation

Bagian ini menjelaskan cara menggunakan AWS CloudFormation untuk memperbarui server `EndpointType` ke`VPC`. Gunakan prosedur ini untuk server Transfer Family yang telah Anda gunakan`AWS CloudFormation`. Dalam contoh ini, AWS CloudFormation template asli yang digunakan untuk menyebarkan server Transfer Family ditampilkan sebagai berikut:

```
AWSTemplateFormatVersion: '2010-09-09'
Description: 'Create AWS Transfer Server with VPC_ENDPOINT endpoint type'
Parameters:
  SecurityGroupId:
    Type: AWS::EC2::SecurityGroup::Id
  SubnetIds:
    Type: List<AWS::EC2::Subnet::Id>
  VpcId:
    Type: AWS::EC2::VPC::Id
Resources:
  TransferServer:
    Type: AWS::Transfer::Server
    Properties:
      Domain: S3
      EndpointDetails:
        VpcEndpointId: !Ref VPCEndpoint
      EndpointType: VPC_ENDPOINT
      IdentityProviderType: SERVICE_MANAGED
      Protocols:
        - SFTP
  VPCEndpoint:
    Type: AWS::EC2::VPCEndpoint
    Properties:
      ServiceName: com.amazonaws.us-east-1.transfer.server
      SecurityGroupIds:
        - !Ref SecurityGroupId
```

```
      SubnetIds:
        - !Select [0, !Ref SubnetIds]
        - !Select [1, !Ref SubnetIds]
        - !Select [2, !Ref SubnetIds]
      VpcEndpointType: Interface
      VpcId: !Ref VpcId
```

Template diperbarui dengan perubahan berikut:

- `EndpointType`Itu diubah menjadi`VPC`.
- `AWS::EC2::VPCEndpoint`Sumber daya dihapus.
- Itu `SecurityGroupIds``SubnetIds`,, dan `VpcId` dipindahkan ke `EndpointDetails` bagian `AWS::Transfer::Server` sumber daya,
- `VpcEndpointId`Properti `EndpointDetails` telah dihapus.

Template yang diperbarui terlihat sebagai berikut:

```
AWSTemplateFormatVersion: '2010-09-09'
Description: 'Create AWS Transfer Server with VPC endpoint type'
Parameters:
  SecurityGroupId:
    Type: AWS::EC2::SecurityGroup::Id
  SubnetIds:
    Type: List<AWS::EC2::Subnet::Id>
  VpcId:
    Type: AWS::EC2::VPC::Id
Resources:
  TransferServer:
    Type: AWS::Transfer::Server
    Properties:
      Domain: S3
      EndpointDetails:
        SecurityGroupIds:
          - !Ref SecurityGroupId
        SubnetIds:
          - !Select [0, !Ref SubnetIds]
          - !Select [1, !Ref SubnetIds]
          - !Select [2, !Ref SubnetIds]
        VpcId: !Ref VpcId
      EndpointType: VPC
      IdentityProviderType: SERVICE_MANAGED
```

```
      Protocols:
        - SFTP
```

Untuk memperbarui jenis endpoint dari server Transfer Family yang digunakan AWS CloudFormation

1. Hentikan server yang ingin Anda perbarui menggunakan langkah-langkah berikut.

   a. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.

   b. Di panel navigasi, pilih Server.

   c. Pilih kotak centang server yang ingin Anda ubah tipe endpoint.

   > **Important**
   >
   > Anda harus menghentikan server sebelum Anda dapat mengubah titik akhirnya.

   d. Untuk Tindakan, pilih Berhenti.

   e. Di kotak dialog konfirmasi yang muncul, pilih Berhenti untuk mengonfirmasi bahwa Anda ingin menghentikan server.

   > **Note**
   >
   > Sebelum melanjutkan ke langkah berikutnya, tunggu Status server berubah menjadi Offline; ini bisa memakan waktu beberapa menit. Anda mungkin harus memilih Refresh di halaman Server untuk melihat perubahan status.

2. Perbarui CloudFormation tumpukan

   a. Buka AWS CloudFormation konsol di https://console.aws.amazon.com/cloudformation.

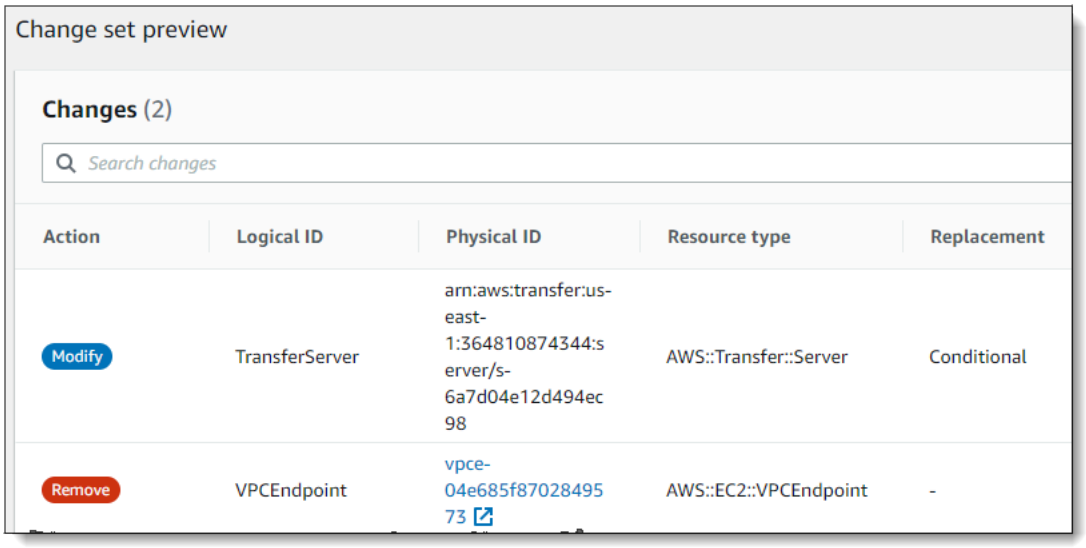
   b. Pilih tumpukan yang digunakan untuk membuat server Transfer Family.

   c. Pilih Perbarui.

   d. Pilih Ganti template saat ini

   e. Unggah template baru. CloudFormation Ubah Set membantu Anda memahami bagaimana perubahan template akan memengaruhi sumber daya yang sedang berjalan sebelum Anda menerapkannya. Dalam contoh ini, sumber daya server Transfer akan dimodifikasi, dan sumber daya VPcendPoint akan dihapus. Server tipe titik akhir VPC membuat titik akhir VPC atas nama Anda, menggantikan sumber daya asli. `VPCEndpoint`

   Setelah mengunggah template baru, set perubahan akan terlihat mirip dengan yang berikut:



f. Perbarui tumpukan.

3. Setelah pembaruan tumpukan selesai, navigasikan ke konsol manajemen Transfer Family di https://console.aws.amazon.com/transfer/.
4. Mulai ulang server. Pilih server yang Anda perbaruiAWS CloudFormation, lalu pilih Mulai dari menu Tindakan.

Memperbarui server EndpointType menggunakan API

Anda dapat menggunakan perintah deskripsi-server, atau AWS CLI perintah API. UpdateServer Contoh skrip berikut menghentikan server Transfer Family, memperbarui EndpointType, menghapus VPC_ENDPOINT, dan memulai server.

```
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])
```

Bekerja dengan nama host khusus

Nama host server Anda adalah nama host yang pengguna Anda masukkan di klien mereka ketika mereka terhubung ke server Anda. Anda dapat menggunakan domain khusus yang telah Anda daftarkan untuk nama host server Anda saat bekerja dengannyaAWS Transfer Family. Misalnya, Anda mungkin menggunakan nama host khusus sepertimy`sftpserver.mysubdomain.domain.com`.

Untuk mengarahkan lalu lintas dari domain kustom terdaftar ke titik akhir server, Anda dapat menggunakan Amazon Route 53 atau penyedia Sistem Nama Domain (DNS) apa pun. Route 53 adalah layanan DNS yang mendukung AWS Transfer Family secara native.

Topik

- Gunakan Amazon Route 53 sebagai penyedia DNS Anda
- Gunakan penyedia DNS lainnya
- Nama host khusus untuk server yang dibuat non-konsol

Di konsol, Anda dapat memilih salah satu opsi ini untuk menyiapkan nama host khusus:

- Amazon Route 53 Alias DNS — jika nama host yang ingin Anda gunakan terdaftar di Route 53. Anda kemudian dapat memasukkan nama host.
- DNS lainnya — jika nama host yang ingin Anda gunakan terdaftar dengan penyedia DNS lain. Anda kemudian dapat memasukkan nama host.
- Tidak ada - untuk menggunakan titik akhir server dan tidak menggunakan nama host khusus.

Anda mengatur opsi ini ketika Anda membuat server baru atau mengedit konfigurasi server yang ada. Untuk informasi selengkapnya tentang membuat server baru, lihatLangkah 2: Buat server berkemampuan SFTP. Untuk informasi selengkapnya tentang mengedit konfigurasi server yang ada, lihatEdit detail server.

Untuk detail selengkapnya tentang penggunaan domain Anda sendiri untuk nama host server dan cara AWS Transfer Family menggunakan Route 53, lihat bagian berikut.

Gunakan Amazon Route 53 sebagai penyedia DNS Anda

Saat membuat server, Anda dapat menggunakan Amazon Route 53 sebagai penyedia DNS Anda. Sebelum Anda menggunakan domain dengan Route 53, Anda mendaftarkan domain. Untuk informasi selengkapnya, lihat Cara kerja pendaftaran Domain di Panduan Pengembang Amazon Route 53.

Saat Anda menggunakan Route 53 untuk menyediakan perutean DNS ke server Anda, AWS Transfer Family gunakan nama host khusus yang Anda masukkan untuk mengekstrak zona hostingnya. Saat AWS Transfer Family mengekstrak zona yang dihosting, tiga hal dapat terjadi:

1. Jika Anda baru mengenal Route 53 dan tidak memiliki zona yang dihosting, AWS Transfer Family tambahkan zona host baru dan `CNAME` catatan. Nilai `CNAME` catatan ini adalah nama host endpoint untuk server Anda. CNAME adalah nama domain alternatif.
2. Jika Anda memiliki zona yang dihosting di Route 53 tanpa `CNAME` catatan apa pun, AWS Transfer Family tambahkan `CNAME` catatan ke zona yang dihosting.
3. Jika layanan mendeteksi bahwa `CNAME` rekaman sudah ada di zona yang dihosting, Anda akan melihat kesalahan yang menunjukkan bahwa `CNAME` rekaman sudah ada. Dalam hal ini, ubah nilai `CNAME` catatan ke nama host server Anda.

> **Note**
>
> Jika langkah ini merupakan bagian dari alur kerja pembuatan server, server Anda berhasil dibuat dan nama host khusus Anda disetel ke Tidak Ada.

Untuk informasi selengkapnya tentang zona yang dihosting di Route 53, lihat Zona yang dihosting di Panduan Pengembang Amazon Route 53.

Gunakan penyedia DNS lainnya

Saat membuat server, Anda juga dapat menggunakan penyedia DNS selain Amazon Route 53. Jika Anda menggunakan penyedia DNS alternatif, pastikan lalu lintas dari domain Anda diarahkan ke titik akhir server Anda.

Untuk melakukannya, atur domain Anda ke nama host endpoint untuk server. Nama host endpoint terlihat seperti ini di konsol:

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

> **Note**
>
> Jika server Anda memiliki titik akhir VPC, maka format untuk nama host berbeda dari yang dijelaskan di atas. Untuk menemukan titik akhir VPC Anda, pilih VPC di halaman detail server, lalu pilih ID titik akhir VPC di dasbor VPC. Endpoint adalah nama DNS pertama dari yang terdaftar.

Nama host khusus untuk server yang dibuat non-konsol

Saat Anda membuat server menggunakanAWS Cloud Development Kit (AWS CDK),AWS CloudFormation, atau melalui CLI, Anda harus menambahkan tag jika Anda ingin server tersebut memiliki nama host khusus. Saat Anda membuat server Transfer Family menggunakan konsol, penandaan dilakukan secara otomatis.

> **Note**
>
> Anda juga perlu membuat catatan DNS untuk mengarahkan lalu lintas dari domain Anda ke titik akhir server Anda. Untuk detailnya, lihat Bekerja dengan catatan di Panduan Pengembang Amazon Route 53.

Gunakan kunci berikut untuk nama host kustom Anda:

- Tambahkan `transfer:customHostname` untuk menampilkan nama host khusus di konsol.
- Jika Anda menggunakan Route 53 sebagai penyedia DNS Anda, tambahkan`transfer:route53HostedZoneId`. Tag ini menautkan nama host kustom ke Route 53 Hosted Zone ID Anda.

Untuk menambahkan nama host khusus, keluarkan perintah CLI berikut.

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

Sebagai contoh:

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

Jika Anda menggunakan Route 53, keluarkan perintah berikut untuk menautkan nama host kustom Anda ke Route 53 Hosted Zone ID.

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

Sebagai contoh:

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

Dengan asumsi nilai sampel dari perintah sebelumnya, jalankan perintah berikut untuk melihat tag Anda:

```
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**
>
> Zona publik Anda yang dihosting, dan ID mereka tersedia di Amazon Route 53. Masuk ke AWS Management Console dan bukalah konsol Route 53 di https://console.aws.amazon.com/route53/.

Mentransfer file melalui titik akhir server menggunakan klien

Anda mentransfer file melalui AWS Transfer Family layanan dengan menentukan operasi transfer di klien. AWS Transfer Family mendukung klien berikut:

- Kami mendukung versi 3 dari protokol SFTP.
- OpenSSH (macOS dan Linux)

> **Note**
>
> Klien ini hanya berfungsi dengan server yang diaktifkan untuk Secure Shell (SSH) File Transfer Protocol (SFTP).

- WinSCP (hanya Microsoft Windows)
- Cyberduck (Windows, macOS, dan Linux)
- FileZilla (Windows, macOS, dan Linux)

Batasan berikut berlaku untuk setiap klien:

- Jumlah maksimum sesi SFTP bersamaan, multipleks, per koneksi adalah 10.
- Amazon S3 dan Amazon EFS (karena protokol NFSv4) memerlukan nama file dalam pengkodean UTF-8. Menggunakan pengkodean yang berbeda dapat menyebabkan hasil yang tidak terduga. Untuk Amazon S3, lihat Pedoman penamaan kunci objek.
- Untuk Protokol Transfer File melalui SSL (FTPS), hanya mode Eksplisit yang didukung. Mode implisit tidak didukung.
- Untuk File Transfer Protocol (FTP) dan FTPS, hanya mode Pasif yang didukung.
- Untuk FTP dan FTPS, hanya mode STREAM yang didukung.
- Untuk FTP dan FTPS, hanya mode Gambar/Biner yang didukung.
- Untuk FTP dan FTPS, TLS - PROT C (tidak dilindungi) TLS untuk koneksi data adalah default tetapi PROT C tidak didukung dalam protokol FTPS. AWS Transfer Family Jadi untuk FTPS, Anda perlu mengeluarkan PROT P agar operasi data Anda dapat diterima.
- Jika Anda menggunakan Amazon S3 untuk penyimpanan server Anda, dan jika klien Anda berisi opsi untuk menggunakan beberapa koneksi untuk satu transfer, pastikan untuk menonaktifkan opsi tersebut. Jika tidak, unggahan file besar dapat gagal dengan cara yang tidak terduga. Perhatikan bahwa jika Anda menggunakan Amazon EFS sebagai backend penyimpanan, EFS mendukung beberapa koneksi untuk satu transfer.

Berikut ini adalah daftar perintah yang tersedia untuk FTP dan FTPS:

Perintah yang tersedia					
ABOR	PRESTASI	MLST	LULUS	RETR	BESAR
AUTENTIKASI	LANG	MKD	PASV	RMD	STOU
CDUP	DAFTAR	MODE	PBSZ	RNFR	STRU
CWD	MDTM	NLST	PROT	RNTO	SYST
DELE	MFMT	NOOP	PWD	UKURAN	TIPE

Perintah yang tersedia					
EPSV	MLSD	MEMILIH	QUIT (BERHENTI)	STAT	USER

> **Note**
>
> APPE tidak didukung.

Untuk SFTP, operasi berikut saat ini tidak didukung untuk pengguna yang menggunakan direktori home logis pada server yang menggunakan Amazon Elastic File System (Amazon EFS).

Perintah SFTP yang tidak didukung			
SSH_FXP_READLINK	SSH_FXP_SYMLINK	SSH_FXP_STAT ketika file yang diminta adalah symlink	SSH_FXP_REALPATH ketika jalur yang diminta berisi komponen symlink

Hasilkan key pair publik-pribadi

Sebelum Anda dapat mentransfer file, Anda harus memiliki key pair publik-pribadi yang tersedia. Jika sebelumnya Anda belum membuat key pair, lihatBuat kunci SSH untuk pengguna yang dikelola layanan.

Topik

- Perintah SFTP/FTPS/FTP yang tersedia
- Temukan titik akhir Amazon VPC Anda
- Hindari setstat kesalahan
- Gunakan OpenSSH
- Gunakan WinSCP
- Gunakan Cyberduck
- Gunakan FileZilla

- Gunakan klien Perl
- Pemrosesan unggahan pasca

Perintah SFTP/FTPS/FTP yang tersedia

Tabel berikut menjelaskan perintah yang tersedia untuk AWS Transfer Family, untuk protokol SFTP, FTPS, dan FTP.

> **Note**
>
> Tabel menyebutkan file dan direktori untuk Amazon S3, yang hanya mendukung bucket dan objek: tidak ada hierarki. Namun, Anda dapat menggunakan awalan dalam nama kunci objek untuk menyiratkan hierarki dan mengatur data Anda dengan cara yang mirip dengan folder. Perilaku ini dijelaskan dalam Bekerja dengan metadata objek di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Perintah SFTP/FTPS/FTP

Perintah	Amazon S3	Amazon EFS
`cd`	Didukung	Didukung
`chgrp`	Tidak Support	Didukung (`root`atau `owner` hanya)
`chmod`	Tidak didukung	Didukung (`root`hanya)
`chmtime`	Tidak didukung	Didukung
`chown`	Tidak Support	Didukung (`root`hanya)
`get`	Didukung	Didukung (termasuk menyelesaikan tautan simbolik)
`ln -s`	Tidak didukung	Didukung
`ls/dir`	Didukung	Didukung

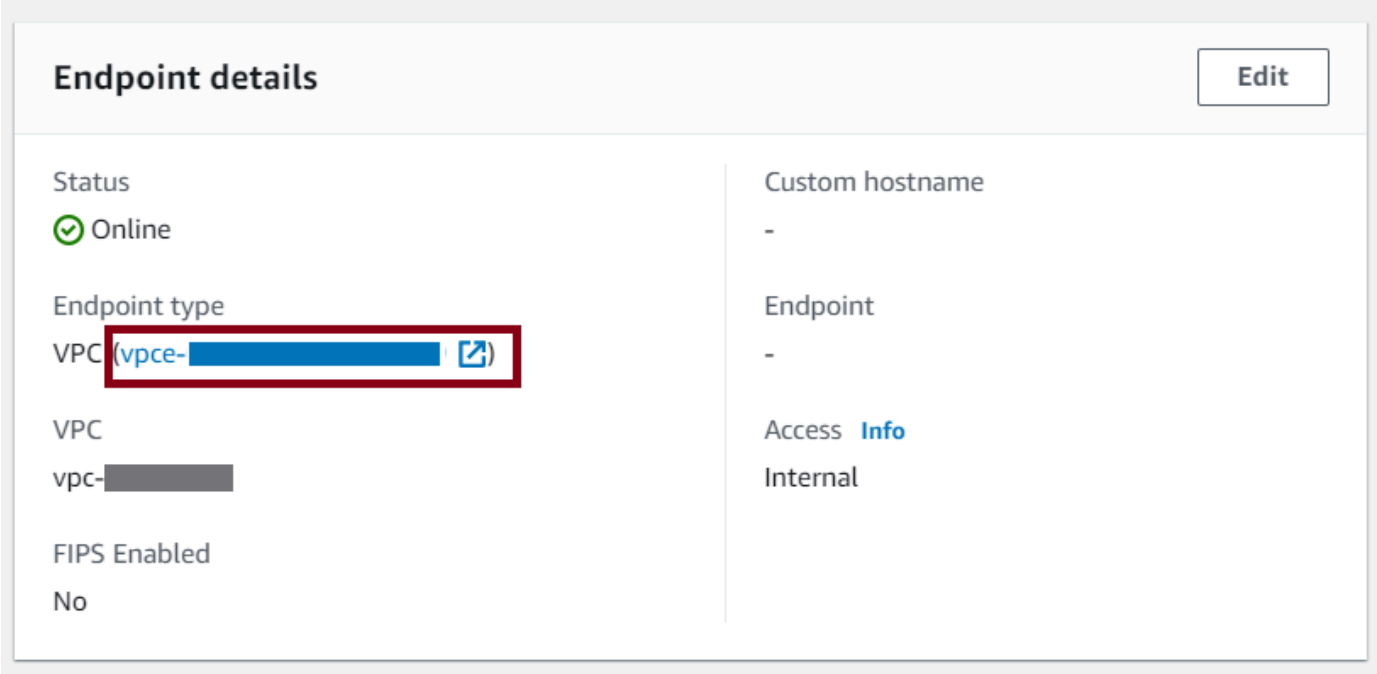
Perintah	Amazon S3	Amazon EFS
`mkdir`	Didukung	Didukung
`put`	Didukung	Didukung
`pwd`	Didukung	Didukung
`rename`	Didukung hanya untuk file	Didukung **Note** Mengganti nama yang akan menimpa file atau direktori yang ada tidak didukung.
`rm`	Didukung	Didukung
`rmdir`	Didukung (hanya direktori kosong)	Didukung
`version`	Didukung	Didukung

Temukan titik akhir Amazon VPC Anda

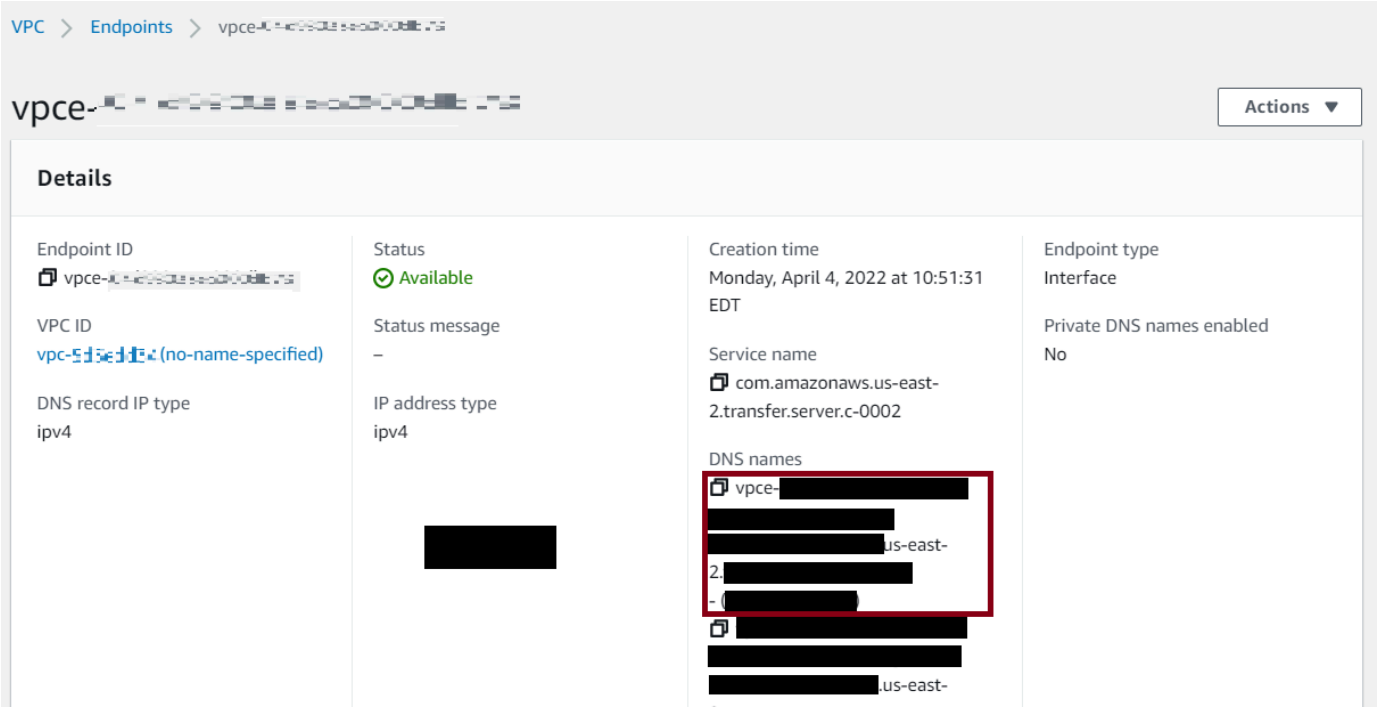
Jika tipe endpoint untuk server Transfer Family Anda adalah VPC, mengidentifikasi titik akhir yang akan digunakan untuk mentransfer file tidaklah mudah. Dalam hal ini, gunakan prosedur berikut untuk menemukan titik akhir VPC Amazon Anda.

Temukan titik akhir Amazon VPC Anda

1. Arahkan ke halaman detail server Anda.
2. Di panel Detail titik akhir, pilih VPC.



3. Di dasbor Amazon VPC, pilih ID titik akhir VPC.
4. Dalam daftar nama DNS, titik akhir server Anda adalah yang pertama terdaftar.



Hindari **`setstat`** kesalahan

Beberapa klien transfer file SFTP dapat mencoba mengubah atribut file jarak jauh, termasuk stempel waktu dan izin, menggunakan perintah, seperti SETSTAT saat mengunggah file. Namun, perintah ini tidak kompatibel dengan sistem penyimpanan objek, seperti Amazon S3. Karena ketidakcocokan ini, unggahan file dari klien ini dapat mengakibatkan kesalahan bahkan ketika file tersebut berhasil diunggah.

- Saat Anda memanggil `UpdateServer` API `CreateServer` atau, gunakan `ProtocolDetails` opsi `SetStatOption` untuk mengabaikan kesalahan yang dihasilkan saat klien mencoba menggunakan SETSTAT pada file yang Anda unggah ke bucket S3.
- Tetapkan nilainya `ENABLE_NO_OP` agar server Transfer Family mengabaikan perintah SETSTAT, dan unggah file tanpa perlu membuat perubahan apa pun pada klien SFTP Anda.
- Perhatikan bahwa meskipun `SetStatOption` `ENABLE_NO_OP` pengaturan mengabaikan kesalahan, itu menghasilkan entri CloudWatch log di Log, sehingga Anda dapat menentukan kapan klien melakukan panggilan SETSTAT.

Untuk detail API untuk opsi ini, lihat ProtocolDetails.

Gunakan OpenSSH

Gunakan instruksi yang mengikuti untuk mentransfer file dari baris perintah menggunakan OpenSSH.

> **Note**
>
> Klien ini hanya berfungsi dengan server berkemampuan SFTP.

Untuk mentransfer file AWS Transfer Family menggunakan utilitas baris perintah OpenSSH

1. Di Linux, macOS, atau Windows, buka terminal perintah.
2. Pada prompt, masukkan perintah berikut:

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

   Pada perintah sebelumnya, *`sftp_user`* adalah nama pengguna dan *`transfer-key`* merupakan kunci pribadi SSH. Di sini, *`service_endpoint`* adalah titik akhir server seperti yang ditunjukkan di AWS Transfer Family konsol untuk server yang dipilih.

> **Note**
>
> Perintah ini menggunakan pengaturan yang ada di `ssh_config` file default. Kecuali Anda sebelumnya telah mengedit file ini, SFTP menggunakan port 22. Anda dapat menentukan port yang berbeda (misalnya 2222) dengan menambahkan **`-P`** bendera ke perintah, sebagai berikut.
>
> ```
> sftp -P 2222 -i transfer-key sftp_user@service_endpoint
> ```
>
> Atau, jika Anda selalu ingin menggunakan port 2222, Anda dapat memperbarui port default Anda di `ssh_config` file Anda.

`sftp`Prompt akan muncul.

3. (Opsional) Untuk melihat direktori home pengguna, masukkan perintah berikut pada `sftp` prompt:

   ```
   pwd
   ```

4. Untuk mengunggah file dari sistem file Anda ke server Transfer Family, gunakan `put` perintah. Misalnya, untuk mengunggah `hello.txt` (dengan asumsi bahwa file ada di direktori Anda saat ini di sistem file Anda), jalankan perintah berikut pada `sftp` prompt:

   ```
   put hello.txt
   ```

   Pesan yang mirip dengan berikut ini muncul, menunjukkan bahwa transfer file sedang berlangsung, atau selesai.

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

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

> **Note**
>
> Setelah server Anda dibuat, diperlukan beberapa menit agar nama host endpoint server dapat diselesaikan oleh layanan DNS di lingkungan Anda.

Gunakan WinSCP

Gunakan instruksi yang mengikuti untuk mentransfer file dari baris perintah menggunakan WinSCP.

> **Note**
>
> Jika Anda menggunakan WinSCP 5.19, Anda dapat langsung terhubung ke Amazon S3 menggunakan kredensyal Anda dan mengunggah/mengunduh file. AWS Untuk detail selengkapnya, lihat Menghubungkan ke layanan Amazon S3.

Untuk mentransfer file AWS Transfer Family menggunakan WinSCP

1. Buka klien WinSCP.
2. Di kotak dialog Login, untuk protokol File, pilih protokol: SFTP atau FTP.

   Jika Anda memilih FTP, untuk Enkripsi, pilih salah satu dari berikut ini:

   - Tidak ada enkripsi untuk FTP
   - Enkripsi eksplisit TLS/SSL untuk FTPS
3. Untuk nama Host, masukkan endpoint server Anda. Titik akhir server terletak di halaman detail Server. Untuk informasi selengkapnya, lihat Lihat detail server SFTP, FTPS, dan FTP.

   > **Note**
   >
   > Jika server Anda menggunakan titik akhir VPC, lihat. Temukan titik akhir Amazon VPC Anda
4. Untuk nomor Port, masukkan yang berikut ini:

   - **22**untuk SFTP
   - **21**untuk FTP/FTPS
5. Untuk nama Pengguna, masukkan nama untuk pengguna yang Anda buat untuk penyedia identitas spesifik Anda.

> **Note**
>
> Nama pengguna harus menjadi salah satu pengguna yang Anda buat atau konfigurasikan untuk penyedia identitas Anda. AWS Transfer Family menyediakan penyedia identitas berikut:
>
> - Bekerja dengan pengguna yang dikelola layanan
> - Menggunakan penyedia identitas AWS Directory Service
> - Bekerja dengan penyedia identitas khusus

6. Pilih Advanced untuk membuka kotak dialog Advanced Site Settings. Di bagian SSH, pilih Otentikasi.
7. Untuk file kunci pribadi, telusuri dan pilih file kunci pribadi SSH dari sistem file Anda.

> **Note**
>
> Jika WinSCP menawarkan untuk mengonversi kunci pribadi SSH Anda ke format PPK, pilih OK.

8. Pilih OK untuk kembali ke kotak dialog Login, lalu pilih Simpan.
9. Dalam kotak dialog Simpan sesi sebagai situs, pilih OK untuk menyelesaikan pengaturan koneksi Anda.
10. Di kotak dialog Login, pilih Tools, lalu pilih Preferences.
11. Di kotak dialog Preferensi, untuk Transfer, pilih Endurance.

    Untuk opsi Aktifkan transfer lanjut/transfer ke nama file sementara untuk opsi, pilih Nonaktifkan.

> **Note**
>
> Jika Anda membiarkan opsi ini diaktifkan, ini meningkatkan biaya unggahan, secara substansional mengurangi kinerja unggahan. Hal ini juga dapat menyebabkan kegagalan upload file besar.

12. Untuk Transfer, pilih Latar Belakang, dan hapus kotak centang Gunakan beberapa koneksi untuk transfer tunggal.

> **Note**
>
> Jika Anda membiarkan opsi ini dipilih, unggahan file besar dapat gagal dengan cara yang tidak terduga. Misalnya, unggahan multipart yatim piatu yang dikenakan biaya Amazon S3 dapat dibuat. Korupsi data senyap juga dapat terjadi.

13. Lakukan transfer file Anda.

    Anda dapat menggunakan drag-and-drop metode untuk menyalin file antara target dan jendela sumber. Anda dapat menggunakan ikon bilah alat untuk mengunggah, mengunduh, menghapus, mengedit, atau memodifikasi properti file di WinSCP.

> **Note**
>
> Catatan ini tidak berlaku jika Anda menggunakan Amazon EFS untuk penyimpanan. Perintah yang mencoba mengubah atribut file jarak jauh, termasuk stempel waktu, tidak kompatibel dengan sistem penyimpanan objek seperti Amazon S3. Oleh karena itu, jika Anda menggunakan Amazon S3 untuk penyimpanan, pastikan untuk menonaktifkan pengaturan stempel waktu WinSCP (atau gunakan `SetStatOption` seperti yang dijelaskan dalam) sebelum Anda melakukan transfer file. Hindari `setstat` kesalahan Untuk melakukannya, di kotak dialog WinSCP Transfer settings, nonaktifkan opsi Setel izin upload dan opsi Preserve timestamp common.

Gunakan Cyberduck

Gunakan instruksi yang mengikuti untuk mentransfer file dari baris perintah menggunakan Cyberduck.

Untuk mentransfer file AWS Transfer Family menggunakan Cyberduck

1. Buka klien Cyberduck.
2. Pilih Buka Koneksi.
3. Dalam kotak dialog Open Connection, pilih protokol: SFTP (SSH File Transfer Protocol), FTP-SSL (Explicit AUTH TLS), atau FTP (File Transfer Protocol).

4. Untuk Server, masukkan endpoint server Anda. Titik akhir server terletak di halaman detail Server. Untuk informasi selengkapnya, lihat Lihat detail server SFTP, FTPS, dan FTP.

   > **Note**
   >
   > Jika server Anda menggunakan titik akhir VPC, lihat. Temukan titik akhir Amazon VPC Anda

5. Untuk nomor Port, masukkan yang berikut ini:

   - **22**untuk SFTP
   - **21**untuk FTP/FTPS

6. Untuk Nama Pengguna, masukkan nama untuk pengguna yang Anda buatMengelola pengguna untuk titik akhir server.
7. Jika SFTP dipilih, untuk SSH Private Key, pilih atau masukkan kunci pribadi SSH.
8. Pilih Hubungkan.
9. Lakukan transfer file Anda.

   Tergantung di mana file Anda berada, lakukan salah satu hal berikut:

   - Di direktori lokal Anda (sumber), pilih file yang ingin Anda transfer, dan seret dan jatuhkan ke direktori Amazon S3 (target).
   - Di direktori Amazon S3 (sumber), pilih file yang ingin Anda transfer, dan seret dan jatuhkan ke direktori lokal Anda (target).

Gunakan FileZilla

Gunakan instruksi yang mengikuti untuk mentransfer file menggunakan FileZilla.

FileZilla Untuk mengatur transfer file

1. Buka FileZilla klien.
2. Pilih File, lalu pilih Site Manager.
3. Di kotak dialog Pengelola Situs, pilih Situs baru.
4. Pada tab Umum, untuk Protokol, pilih protokol: SFTP atau FTP.

   Jika Anda memilih FTP, untuk Enkripsi, pilih salah satu dari berikut ini:

- Hanya gunakan FTP biasa (tidak aman) — untuk FTP
- Gunakan FTP eksplisit melalui TLS jika tersedia - untuk FTPS

5. Untuk nama Host, masukkan protokol yang Anda gunakan, diikuti oleh endpoint server Anda. Titik akhir server terletak di halaman detail Server. Untuk informasi selengkapnya, lihat Lihat detail server SFTP, FTPS, dan FTP.

   > **Note**
   >
   > Jika server Anda menggunakan titik akhir VPC, lihat. Temukan titik akhir Amazon VPC Anda

   - Jika Anda menggunakan SFTP, masukkan: `sftp://hostname`
   - Jika Anda menggunakan FTPS, masukkan: `ftps://hostname`

   Pastikan untuk mengganti *`nama host`* dengan endpoint server Anda yang sebenarnya.

6. Untuk nomor Port, masukkan yang berikut ini:

   - **22**untuk SFTP
   - **21**untuk FTP/FTPS

7. Jika SFTP dipilih, untuk Jenis Logon, pilih File kunci.

   Untuk file Kunci, pilih atau masukkan kunci pribadi SSH.

8. Untuk Pengguna, masukkan nama untuk pengguna yang Anda buatMengelola pengguna untuk titik akhir server.
9. Pilih Hubungkan.
10. Lakukan transfer file Anda.

    > **Note**
    >
    > Jika Anda mengganggu transfer file yang sedang berlangsung, AWS Transfer Family mungkin menulis sebagian objek di bucket Amazon S3 Anda. Jika Anda mengganggu unggahan, periksa apakah ukuran file di bucket Amazon S3 cocok dengan ukuran file objek sumber sebelum melanjutkan.

Gunakan klien Perl

Jika Anda menggunakan klien NET::SFTP::Foreign perl, Anda harus mengatur `queue_size` ke1. Sebagai contoh:

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

> **Note**
>
> Solusi ini diperlukan untuk revisi sebelum 1.92.02. `Net::SFTP::Foreign`

Pemrosesan unggahan pasca

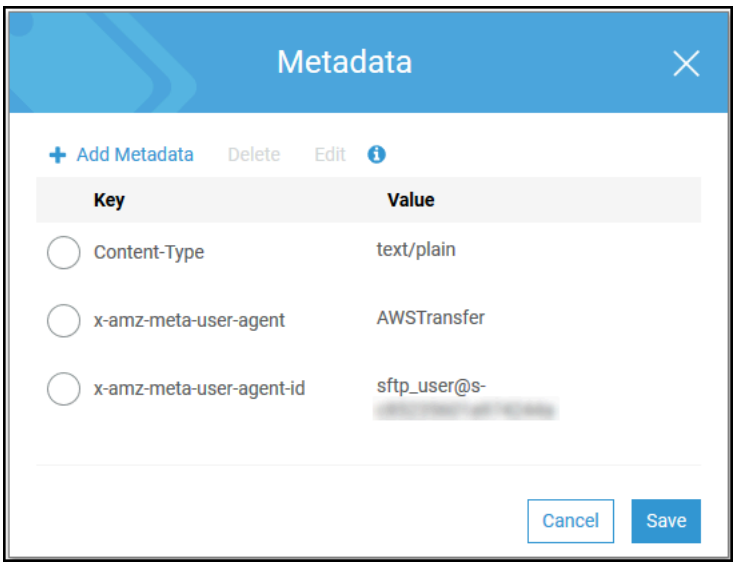
Anda dapat melihat informasi pemrosesan unggahan posting termasuk metadata objek Amazon S3 dan pemberitahuan acara.

Topik

- Metadata objek Amazon S3
- Pemberitahuan acara Amazon S3

Metadata objek Amazon S3

Sebagai bagian dari metadata objek Anda, Anda melihat kunci yang disebut `x-amz-meta-user-agent` yang nilainya `AWSTransfer` dan `x-amz-meta-user-agent-id` nilainya. `username@server-id` `username`Ini adalah pengguna Transfer Family yang mengunggah file dan `server-id` merupakan server yang digunakan untuk mengunggah. Informasi ini dapat diakses menggunakan HeadObjectoperasi pada objek S3 di dalam fungsi Lambda Anda.



Pemberitahuan acara Amazon S3

Saat objek diunggah ke bucket S3 Anda menggunakan Transfer Family, `RoleSessionName` terdapat dalam bidang Peminta dalam struktur notifikasi peristiwa S3 sebagai. `[AWS:Role Unique Identifier]/username.sessionid@server-id` Misalnya, berikut ini adalah konten untuk bidang Peminta sampel dari log akses S3 untuk file yang disalin ke bucket S3.

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

Di bidang Pemohon di atas, ini menunjukkan Peran IAM yang disebut. `IamRoleName` Untuk informasi selengkapnya tentang mengonfigurasi notifikasi peristiwa S3, lihat Mengonfigurasi notifikasi peristiwa Amazon S3 di Panduan Pengembang Layanan Penyimpanan Sederhana Amazon. Untuk informasi selengkapnya tentang pengidentifikasi unik peran AWS Identity and Access Management (IAM), lihat Pengidentifikasi unik di Panduan Pengguna.AWS Identity and Access Management

Mengelola pengguna untuk titik akhir server

Di bagian berikut, Anda dapat menemukan informasi tentang cara menambahkan pengguna yang menggunakanAWS Transfer Family, AWS Directory Service for Microsoft Active Directory atau penyedia identitas khusus.

Jika Anda menggunakan tipe identitas yang dikelola layanan, Anda menambahkan pengguna ke server yang diaktifkan protokol transfer file Anda. Ketika Anda melakukannya, setiap nama pengguna harus unik di server Anda.

Sebagai bagian dari properti setiap pengguna, Anda juga menyimpan kunci publik Secure Shell (SSH) pengguna tersebut. Melakukan hal itu diperlukan untuk otentikasi berbasis kunci, yang digunakan prosedur ini. Kunci pribadi disimpan secara lokal di komputer pengguna Anda. Ketika pengguna Anda mengirim permintaan otentikasi ke server Anda dengan menggunakan klien, server Anda terlebih dahulu mengonfirmasi bahwa pengguna memiliki akses ke kunci pribadi SSH terkait. Server kemudian berhasil mengotentikasi pengguna.

Selain itu, Anda menentukan direktori home pengguna, atau direktori landing, dan menetapkan peran AWS Identity and Access Management (IAM) kepada pengguna. Secara opsional, Anda dapat memberikan kebijakan sesi untuk membatasi akses pengguna hanya ke direktori home bucket Amazon S3 Anda.

> **Important**
>
> AWS Transfer Familymemblokir nama pengguna yang panjangnya 1 atau 2 karakter dari otentikasi ke server SFTP. Selain itu, kami juga memblokir nama `root` pengguna.
> Alasan di balik ini adalah karena volume besar upaya login berbahaya oleh pemindai kata sandi.

Amazon EFS vs Amazon S3

Karakteristik masing-masing opsi penyimpanan:

- Untuk membatasi akses: Amazon S3 mendukung kebijakan sesi; Amazon EFS mendukung ID pengguna, grup, dan grup sekunder POSIX
- Keduanya mendukung kunci publik/pribadi
- Keduanya mendukung direktori rumah
- Keduanya mendukung direktori logis

> **Note**
>
> Untuk Amazon S3, sebagian besar dukungan untuk direktori logis adalah melalui API/CLI. Anda dapat menggunakan kotak centang Dibatasi di konsol untuk mengunci pengguna ke direktori home mereka, tetapi Anda tidak dapat menentukan struktur direktori virtual.

Direktori logis

Jika Anda menentukan nilai direktori logis untuk pengguna Anda, parameter yang Anda gunakan tergantung pada jenis pengguna.

- Untuk pengguna yang dikelola layanan, berikan nilai direktori logis di. `HomeDirectoryMappings`
- Untuk pengguna penyedia identitas kustom, berikan nilai direktori logis di`HomeDirectoryDetails`.

Topik

- Bekerja dengan pengguna yang dikelola layanan
- Menggunakan penyedia identitas AWS Directory Service
- Bekerja dengan penyedia identitas khusus

Bekerja dengan pengguna yang dikelola layanan

Anda dapat menambahkan pengguna yang dikelola layanan Amazon S3 atau Amazon EFS ke server Anda, tergantung pada pengaturan Domain server. Untuk informasi selengkapnya, lihat Mengkonfigurasi titik akhir server SFTP, FTPS, atau FTP.

Untuk menambahkan pengguna yang dikelola layanan secara terprogram, lihat contoh untuk API. CreateUser

> **Note**
>
> Untuk pengguna yang dikelola layanan ada batas 2.000 entri direktori logis. Untuk informasi tentang menggunakan direktori logis, lihatMenggunakan direktori logis untuk menyederhanakan struktur direktori Transfer Family Anda.

Topik

- Menambahkan pengguna yang dikelola layanan Amazon S3
- Menambahkan pengguna yang dikelola layanan Amazon EFS
- Mengelola pengguna yang dikelola layanan

Menambahkan pengguna yang dikelola layanan Amazon S3

> **Note**
>
> Jika Anda ingin mengonfigurasi bucket Amazon S3 lintas akun, ikuti langkah-langkah yang disebutkan dalam artikel Pusat Pengetahuan ini: Bagaimana cara mengonfigurasi AWS Transfer Family server saya untuk menggunakan bucket Amazon Simple Storage Service yang ada di akun lain? AWS .

Untuk menambahkan pengguna yang dikelola layanan Amazon S3 ke server Anda

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/, lalu pilih Server dari panel navigasi.
2. Pada halaman Server, pilih kotak centang server yang ingin Anda tambahkan pengguna.
3. Pilih Tambahkan pengguna.
4. Di bagian Konfigurasi pengguna, untuk Nama Pengguna, masukkan nama pengguna. Nama pengguna ini harus minimal 3 dan maksimal 100 karakter. Anda dapat menggunakan karakter berikut dalam nama pengguna: a—z, A-Z, 0-9, garis bawah '_', tanda hubung '-', periode ' . ', dan pada tanda “@”. Nama pengguna tidak dapat dimulai dengan tanda hubung '-', titik ' . ', atau pada tanda “@”.
5. Untuk Access, pilih peran IAM yang sebelumnya Anda buat yang menyediakan akses ke bucket Amazon S3 Anda.

   Anda membuat peran IAM ini menggunakan prosedur diBuat peran dan kebijakan IAM. Peran IAM tersebut mencakup kebijakan IAM yang menyediakan akses ke bucket Amazon S3 Anda. Ini juga mencakup hubungan kepercayaan dengan AWS Transfer Family layanan, yang didefinisikan dalam kebijakan IAM lain. Jika Anda memerlukan kontrol akses berbutir halus untuk pengguna Anda, lihat Tingkatkan kontrol akses data dengan dan posting blog Amazon AWS Transfer Family S3.
6. (Opsional) Untuk Kebijakan, pilih salah satu dari berikut ini:

   - Tidak ada
   - Kebijakan yang ada
   - Pilih kebijakan dari IAM: memungkinkan Anda memilih kebijakan sesi yang ada. Pilih Lihat untuk melihat objek JSON yang berisi detail kebijakan.

- Kebijakan buat otomatis berdasarkan folder beranda: menghasilkan kebijakan sesi untuk Anda. Pilih Lihat untuk melihat objek JSON yang berisi detail kebijakan.

> **Note**
>
> Jika Anda memilih Kebijakan buat otomatis berdasarkan folder beranda, jangan pilih Dibatasi untuk pengguna ini.

Untuk mempelajari lebih lanjut tentang kebijakan sesi, lihatBuat peran dan kebijakan IAM. Untuk mempelajari lebih lanjut tentang membuat kebijakan sesi, lihatMembuat kebijakan sesi untuk bucket Amazon S3.

7. Untuk direktori Home, pilih bucket Amazon S3 untuk menyimpan data yang akan ditransfer. AWS Transfer Family Masukkan jalur ke `home` direktori tempat pengguna Anda mendarat saat mereka masuk menggunakan klien mereka.

   Jika parameter ini kosong, `root` direktori bucket Amazon S3 Anda akan digunakan. Dalam hal ini, pastikan bahwa peran IAM Anda menyediakan akses ke `root` direktori ini.

   > **Note**
   >
   > Sebaiknya pilih jalur direktori yang berisi nama pengguna pengguna, yang memungkinkan Anda menggunakan kebijakan sesi secara efektif. Kebijakan sesi membatasi akses pengguna di bucket Amazon S3 ke direktori pengguna tersebut. `home`

8. (Opsional) Untuk Dibatasi, pilih kotak centang sehingga pengguna Anda tidak dapat mengakses apa pun di luar folder itu dan tidak dapat melihat keranjang Amazon S3 atau nama folder.

   > **Note**
   >
   > Menugaskan pengguna direktori home dan membatasi pengguna ke direktori home itu harus cukup untuk mengunci akses pengguna ke folder yang ditunjuk. Jika Anda perlu menerapkan kontrol lebih lanjut, gunakan kebijakan sesi.
   > Jika Anda memilih Dibatasi untuk pengguna ini, Anda tidak dapat memilih Kebijakan buat otomatis berdasarkan folder beranda, karena folder beranda bukan nilai yang ditentukan untuk pengguna Terbatas.

9. Untuk kunci publik SSH, masukkan bagian kunci SSH publik dari key pair SSH.

Kunci Anda divalidasi oleh layanan sebelum Anda dapat menambahkan pengguna baru Anda.

> **Note**
>
> Untuk petunjuk tentang cara membuat key pair SSH, lihatBuat kunci SSH untuk pengguna yang dikelola layanan.

10. (Opsional) Untuk Kunci dan Nilai, masukkan satu atau beberapa tag sebagai pasangan nilai kunci, dan pilih Tambahkan tag.
11. Pilih Tambah untuk menambahkan pengguna baru Anda ke server yang Anda pilih.

    Pengguna baru muncul di bagian Pengguna pada halaman detail Server.

Langkah selanjutnya — Untuk langkah selanjutnya, lanjutkan keMentransfer file melalui titik akhir server menggunakan klien.

Menambahkan pengguna yang dikelola layanan Amazon EFS

Amazon EFS menggunakan model izin file Portable Operating System Interface (POSIX) untuk mewakili kepemilikan file.

- Untuk detail selengkapnya tentang kepemilikan file Amazon EFS, lihat Kepemilikan file Amazon EFS.
- Untuk detail selengkapnya tentang menyiapkan direktori untuk pengguna EFS Anda, lihatMenyiapkan pengguna Amazon EFS untuk Transfer Family.

Untuk menambahkan pengguna yang dikelola layanan Amazon EFS ke server Anda

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/, lalu pilih Server dari panel navigasi.
2. Pada halaman Server, pilih server Amazon EFS yang ingin Anda tambahkan pengguna.
3. Pilih Tambah pengguna untuk menampilkan halaman Tambah pengguna.
4. Di bagian Konfigurasi pengguna, gunakan pengaturan berikut.

    a. Nama pengguna, harus minimal 3 dan maksimal 100 karakter. Anda dapat menggunakan karakter berikut dalam nama pengguna: a—z, A-Z, 0-9, garis bawah '_', tanda hubung '-',

periode ' . ', dan pada tanda “@”. Nama pengguna tidak dapat dimulai dengan tanda hubung '-', titik ' . ', atau pada tanda “@”.

b. Untuk User ID dan Group ID, perhatikan hal berikut:

- Untuk pengguna pertama yang Anda buat, kami sarankan Anda memasukkan nilai **`0`** untuk ID Grup dan ID Pengguna. Ini memberikan hak administrator pengguna untuk Amazon EFS.
- Untuk pengguna tambahan, masukkan ID pengguna POSIX dan ID grup pengguna. ID ini digunakan untuk semua operasi Amazon Elastic File System yang dilakukan oleh pengguna.
- Untuk ID Pengguna dan ID Grup, jangan gunakan angka nol di depan. Misalnya, dapat **`12345`** diterima, **`012345`** tidak.

c. (Opsional) Untuk ID Grup Sekunder, masukkan satu atau lebih ID grup POSIX tambahan untuk setiap pengguna, dipisahkan dengan koma.

d. Untuk Access, pilih peran IAM yang:

- Memberikan pengguna akses hanya ke sumber daya Amazon EFS (sistem file) yang Anda ingin mereka akses.
- Mendefinisikan operasi sistem file mana yang dapat dan tidak dapat dilakukan oleh pengguna.

Kami menyarankan Anda menggunakan peran IAM untuk pemilihan sistem file Amazon EFS dengan akses mount dan izin baca/tulis. Misalnya, kombinasi dari dua kebijakan AWS terkelola berikut, meskipun cukup permisif, memberikan izin yang diperlukan untuk pengguna Anda:

- AmazonElasticFileSystemClientFullAccess
- AWSTransferConsoleFullAccess

Untuk informasi selengkapnya, lihat AWS Transfer Familydukungan posting blog untuk Amazon Elastic File System.

e. Untuk direktori Home, lakukan hal berikut:

- Pilih sistem file Amazon EFS yang ingin Anda gunakan untuk menyimpan data yang akan ditransferAWS Transfer Family.

- Putuskan apakah akan mengatur direktori home ke Restricted. Menyetel direktori home ke Restricted memiliki efek berikut:
  - Pengguna Amazon EFS tidak dapat mengakses file atau direktori apa pun di luar folder itu.
  - Pengguna Amazon EFS tidak dapat melihat nama sistem file Amazon EFS (fs-xxxxxxx).

  > **Note**
  >
  > Saat Anda memilih opsi Dibatasi, symlink tidak dapat diselesaikan untuk pengguna Amazon EFS.

- (Opsional) Masukkan jalur ke direktori home yang Anda inginkan agar pengguna masuk saat mereka masuk menggunakan klien mereka.

  Jika Anda tidak menentukan direktori home, direktori root sistem file Amazon EFS Anda akan digunakan. Dalam hal ini, pastikan bahwa peran IAM Anda menyediakan akses ke direktori root ini.

5. Untuk kunci publik SSH, masukkan bagian kunci SSH publik dari key pair SSH.

   Kunci Anda divalidasi oleh layanan sebelum Anda dapat menambahkan pengguna baru Anda.

   > **Note**
   >
   > Untuk petunjuk tentang cara membuat key pair SSH, lihatBuat kunci SSH untuk pengguna yang dikelola layanan.

6. (Opsional) Masukkan tag apa pun untuk pengguna. Untuk Kunci dan Nilai, masukkan satu atau beberapa tag sebagai pasangan nilai kunci, dan pilih Tambah tag.
7. Pilih Tambah untuk menambahkan pengguna baru Anda ke server yang Anda pilih.

   Pengguna baru muncul di bagian Pengguna pada halaman detail Server.

Masalah yang mungkin Anda temui saat pertama kali SFTP ke server Transfer Family Anda:

- Jika Anda menjalankan `sftp` perintah dan prompt tidak muncul, Anda mungkin menemukan pesan berikut:

```
Couldn't canonicalize: Permission denied
```

```
Need cwd
```

Dalam hal ini, Anda harus meningkatkan izin kebijakan untuk peran pengguna Anda. Anda dapat menambahkan kebijakan AWS terkelola, sepertiAmazonElasticFileSystemClientFullAccess.

- Jika Anda memasukkan pwd pada sftp prompt untuk melihat direktori home pengguna, Anda mungkin melihat pesan berikut, di mana *USER-HOME-DIRECTORY adalah direktori home untuk pengguna* SFTP:

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

Dalam hal ini, Anda harus dapat menavigasi ke direktori induk (cd ..), dan membuat direktori home pengguna (mkdir *username*).

Langkah selanjutnya — Untuk langkah selanjutnya, lanjutkan keMentransfer file melalui titik akhir server menggunakan klien.

Mengelola pengguna yang dikelola layanan

Di bagian ini, Anda dapat menemukan informasi tentang cara melihat daftar pengguna, cara mengedit detail pengguna, dan cara menambahkan kunci publik SSH.

- Melihat daftar pengguna
- Melihat atau mengedit detail pengguna
- Menghapus pengguna
- Tambahkan kunci publik SSH
- Hapus kunci publik SSH

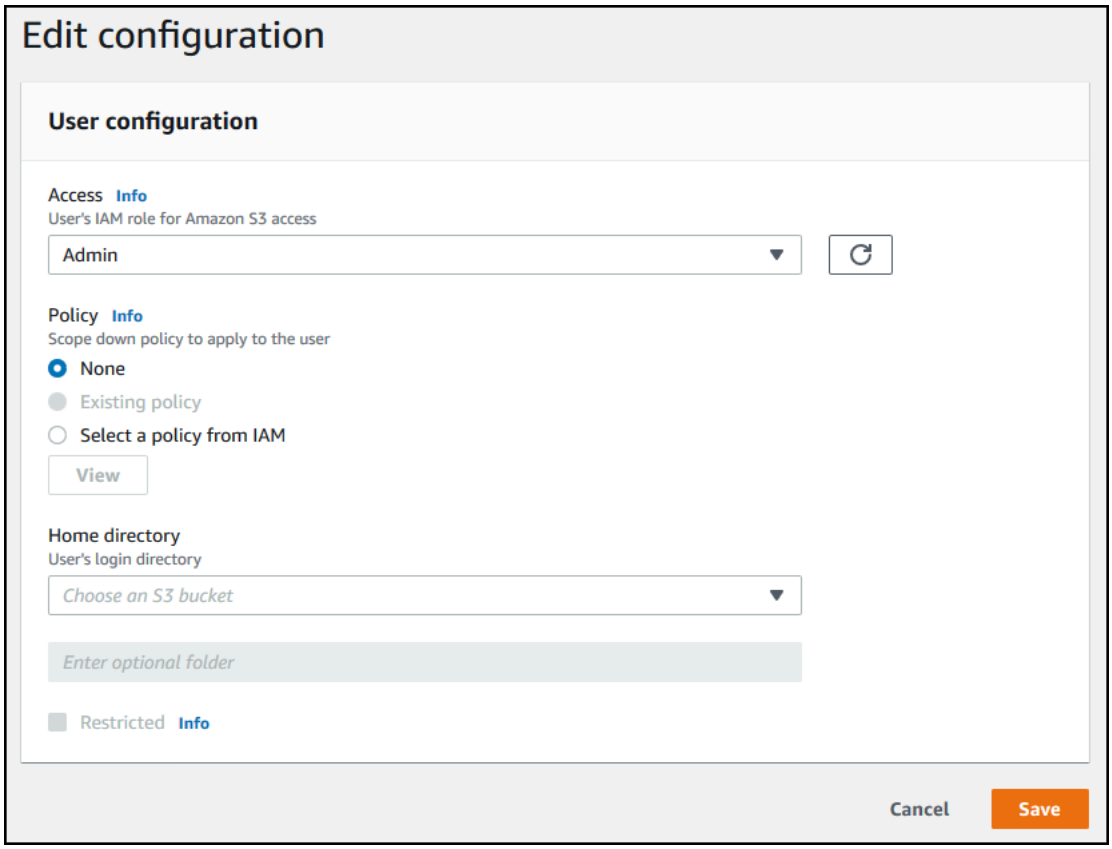
Untuk menemukan daftar pengguna Anda

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Pilih Server dari panel navigasi untuk menampilkan halaman Server.
3. Pilih pengenal di kolom ID Server untuk melihat halaman Detail Server.
4. Di bawah Pengguna, lihat daftar pengguna.

Untuk melihat atau mengedit detail pengguna

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Pilih Server dari panel navigasi untuk menampilkan halaman Server.
3. Pilih pengenal di kolom ID Server untuk melihat halaman Detail Server.
4. Di bawah Pengguna, pilih nama pengguna untuk melihat halaman Detail pengguna.

   Anda dapat mengubah properti pengguna di halaman ini dengan memilih Edit.
5. Pada halaman Detail pengguna, pilih Edit di samping Konfigurasi pengguna.



6. Pada halaman Edit konfigurasi, untuk Access, pilih peran IAM yang sebelumnya Anda buat yang menyediakan akses ke bucket Amazon S3 Anda.

   Anda membuat peran IAM ini menggunakan prosedur diBuat peran dan kebijakan IAM. Peran IAM tersebut mencakup kebijakan IAM yang menyediakan akses ke bucket Amazon S3 Anda. Ini juga mencakup hubungan kepercayaan dengan AWS Transfer Family layanan, yang didefinisikan dalam kebijakan IAM lain.
7. (Opsional) Untuk Kebijakan, pilih salah satu dari berikut ini:

   - Tidak ada

- Kebijakan yang ada
- Pilih kebijakan dari IAM untuk memilih kebijakan yang ada. Pilih Lihat untuk melihat objek JSON yang berisi detail kebijakan.

Untuk mempelajari lebih lanjut tentang kebijakan sesi, lihatBuat peran dan kebijakan IAM. Untuk mempelajari lebih lanjut tentang membuat kebijakan sesi, lihatMembuat kebijakan sesi untuk bucket Amazon S3.

8. Untuk direktori Home, pilih bucket Amazon S3 untuk menyimpan data yang akan ditransfer. AWS Transfer Family Masukkan jalur ke home direktori tempat pengguna Anda mendarat saat mereka masuk menggunakan klien mereka.

   Jika Anda membiarkan parameter ini kosong, `root` direktori bucket Amazon S3 Anda digunakan. Dalam hal ini, pastikan bahwa peran IAM Anda menyediakan akses ke `root` direktori ini.

   > **Note**
   >
   > Sebaiknya pilih jalur direktori yang berisi nama pengguna pengguna, yang memungkinkan Anda menggunakan kebijakan sesi secara efektif. Kebijakan sesi membatasi akses pengguna di bucket Amazon S3 ke direktori pengguna tersebut. home

9. (Opsional) Untuk Dibatasi, pilih kotak centang sehingga pengguna Anda tidak dapat mengakses apa pun di luar folder itu dan tidak dapat melihat keranjang Amazon S3 atau nama folder.

   > **Note**
   >
   > Saat menugaskan pengguna direktori home dan membatasi pengguna ke direktori home itu, ini harus cukup untuk mengunci akses pengguna ke folder yang ditunjuk. Gunakan kebijakan sesi saat Anda perlu menerapkan kontrol lebih lanjut.

10. Pilih Simpan untuk menyimpan perubahan Anda.

Untuk menghapus klaster

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Pilih Server dari panel navigasi untuk menampilkan halaman Server.

3. Pilih pengenal di kolom ID Server untuk melihat halaman Detail Server.
4. Di bawah Pengguna, pilih nama pengguna untuk melihat halaman Detail pengguna.
5. Pada halaman Detail pengguna, pilih Hapus di sebelah kanan nama pengguna.
6. Di kotak dialog konfirmasi yang muncul, masukkan kata**`delete`**, lalu pilih Hapus untuk mengonfirmasi bahwa Anda ingin menghapus pengguna.

Pengguna dihapus dari daftar pengguna.

Untuk menambahkan kunci publik SSH untuk pengguna

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi, pilih Server.
3. Pilih pengenal di kolom ID Server untuk melihat halaman Detail Server.
4. Di bawah Pengguna, pilih nama pengguna untuk melihat halaman Detail pengguna.
5. Pilih Tambahkan kunci publik SSH untuk menambahkan kunci publik SSH baru ke pengguna.

> **Note**
>
> Kunci SSH hanya digunakan oleh server yang diaktifkan untuk Secure Shell (SSH) File Transfer Protocol (SFTP). Untuk informasi tentang cara membuat key pair SSH, lihatBuat kunci SSH untuk pengguna yang dikelola layanan.

6. Untuk kunci publik SSH, masukkan bagian kunci publik SSH dari SSH key pair.

   Kunci Anda divalidasi oleh layanan sebelum Anda dapat menambahkan pengguna baru Anda. Format kunci SSH adalah`ssh-rsa` *`string`*. Untuk menghasilkan key pair SSH, lihatBuat kunci SSH untuk pengguna yang dikelola layanan.
7. Pilih Tambah kunci.

Untuk menghapus kunci publik SSH untuk pengguna

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi, pilih Server.
3. Pilih pengenal di kolom ID Server untuk melihat halaman Detail Server.
4. Di bawah Pengguna, pilih nama pengguna untuk melihat halaman Detail pengguna.

5. Untuk menghapus kunci publik, pilih kotak centang kunci SSH dan pilih Hapus.

Menggunakan penyedia identitas AWS Directory Service

Topik ini menjelaskan cara menggunakan penyedia identitas AWS Directory Service untuk AWS Transfer Family.

Topik

- Menggunakan AWS Directory Service for Microsoft Active Directory
- Menggunakan AWS Directory Service untuk Azure Active Directory Domain Services

Menggunakan AWS Directory Service for Microsoft Active Directory

Anda dapat menggunakan AWS Transfer Family untuk mengautentikasi pengguna akhir transfer file Anda menggunakan AWS Directory Service for Microsoft Active Directory. Ini memungkinkan migrasi mulus dari alur kerja transfer file yang mengandalkan otentikasi Active Directory tanpa mengubah kredensyal pengguna akhir atau memerlukan otorisasi khusus.

Dengan AWS Managed Microsoft AD, Anda dapat dengan aman memberikan akses kepada AWS Directory Service pengguna dan grup melalui SFTP, FTPS, dan FTP untuk data yang disimpan di Amazon Simple Storage Service (Amazon S3) atau Amazon Elastic File System (Amazon EFS). Jika Anda menggunakan Active Directory untuk menyimpan kredensi pengguna Anda, Anda sekarang memiliki cara yang lebih mudah untuk mengaktifkan transfer file untuk pengguna ini.

Anda dapat memberikan akses ke grup Active Directory AWS Managed Microsoft AD di lingkungan lokal atau di AWS Cloud menggunakan konektor Active Directory. Anda dapat memberi pengguna yang sudah dikonfigurasi di lingkungan Microsoft Windows Anda, baik di AWS Cloud atau di jaringan lokal mereka, akses ke AWS Transfer Family server yang menggunakan AWS Managed Microsoft AD identitas.

> **Note**
>
> - AWS Transfer Family tidak mendukung Simple AD.
> - Transfer Family tidak mendukung konfigurasi Direktori Aktif lintas wilayah: kami hanya mendukung integrasi Direktori Aktif yang berada di wilayah yang sama dengan server Transfer Family.

- Transfer Family tidak mendukung penggunaan salah satu AWS Managed Microsoft AD atau AD Connector untuk mengaktifkan otentikasi multi-faktor (MFA) untuk infrastruktur MFA berbasis Radius yang ada.
- AWS Transfer Family tidak mendukung wilayah yang direplikasi dari Direktori Aktif Terkelola.

Untuk menggunakannya AWS Managed Microsoft AD, Anda harus melakukan langkah-langkah berikut:

1. Buat satu atau lebih AWS Managed Microsoft AD direktori menggunakan AWS Directory Service konsol.
2. Gunakan konsol Transfer Family untuk membuat server yang digunakan AWS Managed Microsoft AD sebagai penyedia identitasnya.
3. Tambahkan akses dari satu atau beberapa AWS Directory Service grup Anda.
4. Meskipun tidak diperlukan, kami menyarankan Anda menguji dan memverifikasi akses pengguna.

Topik

- Sebelum Anda mulai menggunakan AWS Directory Service for Microsoft Active Directory
- Bekerja dengan ranah Active Directory
- Memilih AWS Managed Microsoft AD sebagai penyedia identitas Anda
- Memberikan akses ke grup
- Menguji pengguna
- Menghapus akses server untuk grup
- Menghubungkan ke server menggunakan SSH (Secure Shell)
- Menghubungkan AWS Transfer Family ke Active Directory yang dikelola sendiri menggunakan hutan dan trust

Sebelum Anda mulai menggunakan AWS Directory Service for Microsoft Active Directory

Menyediakan pengenal unik untuk grup iklan Anda

Sebelum dapat menggunakan AWS Managed Microsoft AD, Anda harus memberikan pengenal unik untuk setiap grup di direktori Microsoft AD Anda. Anda dapat menggunakan pengenal keamanan

(SID) untuk setiap grup untuk melakukan ini. Pengguna grup yang Anda asosiasikan memiliki akses ke sumber daya Amazon S3 atau Amazon EFS Anda melalui protokol yang diaktifkan menggunakan Transfer Family. AWS

Gunakan PowerShell perintah Windows berikut untuk mengambil SID untuk grup, ganti *YourGroupName*dengan nama grup.

```
Get-ADGroup -Filter {samAccountName -like "YourGroupName*"} -Properties * | Select
 SamAccountName,ObjectSid
```

> **Note**
>
> Jika Anda menggunakan AWS Directory Service sebagai penyedia identitas Anda, dan jika `userPrincipalName` dan `SamAccountName` memiliki nilai yang berbeda, AWS Transfer Family terima nilainya. `SamAccountName` Transfer Family tidak menerima nilai yang ditentukan dalam`userPrincipalName`.

Tambahkan AWS Directory Service izin ke peran Anda

Anda juga memerlukan izin AWS Directory Service API untuk digunakan AWS Directory Service sebagai penyedia identitas Anda. Izin berikut diperlukan atau disarankan:

- `ds:DescribeDirectories`Diperlukan Transfer Family untuk mencari direktori
- `ds:AuthorizeApplication`diperlukan untuk menambahkan otorisasi untuk Transfer Family
- `ds:UnauthorizeApplication`disarankan untuk menghapus sumber daya apa pun yang dibuat untuk sementara, jika terjadi kesalahan selama proses pembuatan server

Tambahkan izin ini ke peran yang Anda gunakan untuk membuat server Transfer Family Anda. Untuk detail selengkapnya tentang izin ini, lihat Izin AWS Directory Service API: Referensi tindakan, sumber daya, dan kondisi.

Bekerja dengan ranah Active Directory

Saat Anda mempertimbangkan cara agar pengguna Active Directory mengakses AWS Transfer Family server, ingatlah ranah pengguna, dan ranah grup mereka. Idealnya, ranah pengguna dan ranah grup mereka harus cocok. Artinya, baik pengguna maupun grup berada di ranah default, atau

keduanya berada di ranah tepercaya. Jika tidak demikian, pengguna tidak dapat diautentikasi oleh Transfer Family.

Anda dapat menguji pengguna untuk memastikan konfigurasi sudah benar. Untuk detailnya, lihat Menguji pengguna. Jika ada masalah dengan ranah pengguna/grup, Anda menerima kesalahan, Tidak ada akses terkait yang ditemukan untuk grup pengguna.

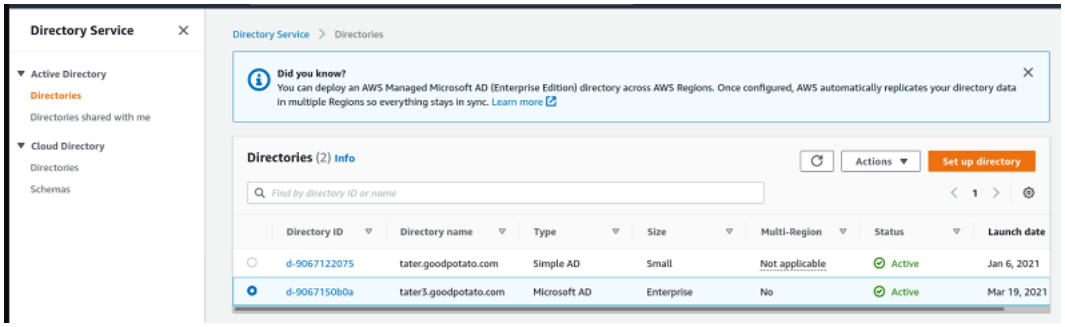
Memilih AWS Managed Microsoft AD sebagai penyedia identitas Anda

Bagian ini menjelaskan cara menggunakan AWS Directory Service for Microsoft Active Directory dengan server.

Untuk digunakan AWS Managed Microsoft AD dengan Transfer Family

1. Masuk ke AWS Management Console dan buka AWS Directory Service konsol di https://console.aws.amazon.com/directoryservicev2/.

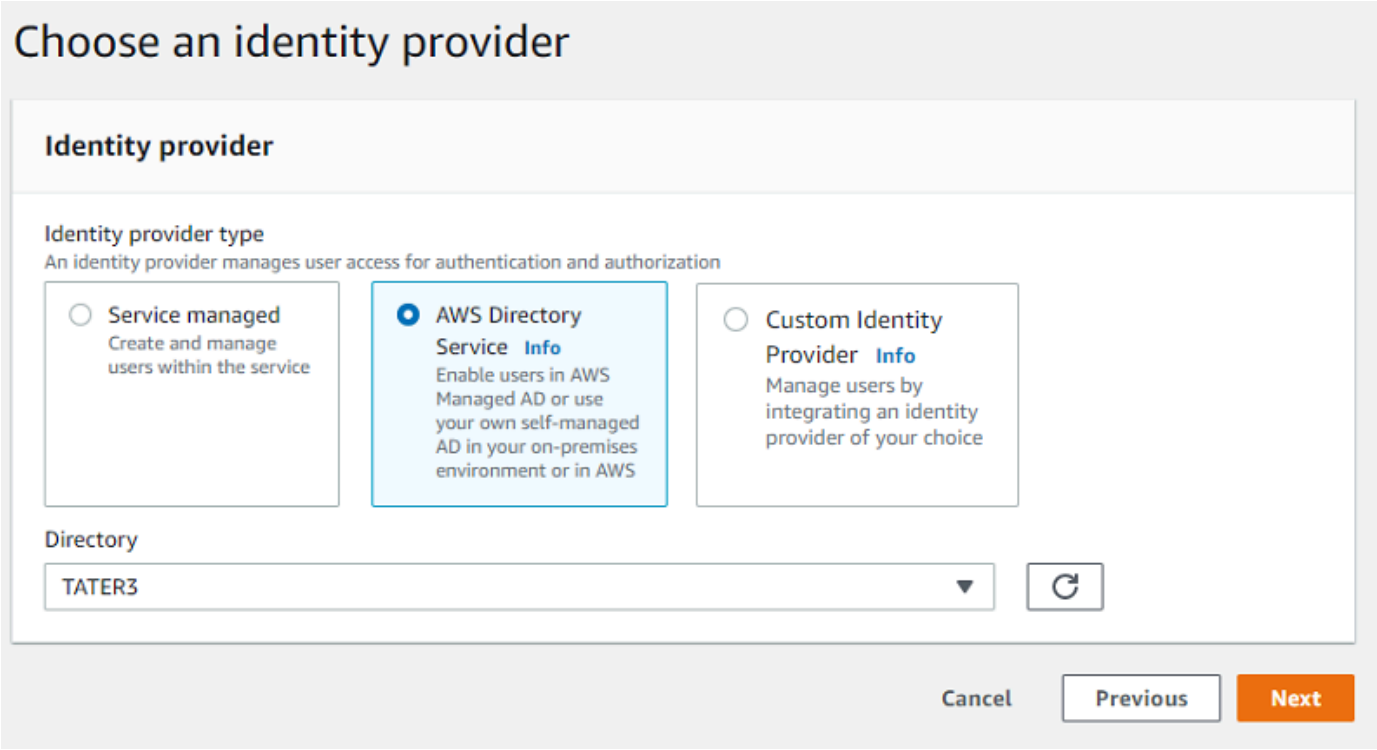
   Gunakan AWS Directory Service konsol untuk mengonfigurasi satu atau beberapa direktori terkelola. Untuk informasi lebih lanjut, lihat AWS Managed Microsoft AD dalam Panduan Admin AWS Directory Service .



2. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/, dan pilih Buat server.
3. Pada halaman Pilih protokol, pilih satu atau beberapa protokol dari daftar.

   > **Note**
   >
   > Jika Anda memilih FTPS, Anda harus memberikan AWS Certificate Manager sertifikat.

4. Untuk Pilih penyedia identitas, pilih AWS Directory Service.



5. Daftar Direktori berisi semua direktori terkelola yang telah Anda konfigurasi. Pilih direktori dari daftar, dan pilih Berikutnya.

> **Note**
>
> - Direktori Cross-Account dan Shared tidak didukung untuk. AWS Managed Microsoft AD
> - Untuk menyiapkan server dengan Directory Service sebagai penyedia identitas Anda, Anda perlu menambahkan beberapa AWS Directory Service izin. Untuk detailnya, lihat Sebelum Anda mulai menggunakan AWS Directory Service for Microsoft Active Directory.

6. Untuk menyelesaikan pembuatan server, gunakan salah satu prosedur berikut:

   - Buat server berkemampuan SFTP
   - Buat server berkemampuan FTPS
   - Buat server berkemampuan FTP

   Dalam prosedur tersebut, lanjutkan dengan langkah berikut memilih penyedia identitas.

> **Important**
>
> Anda tidak dapat menghapus direktori Microsoft AD AWS Directory Service jika Anda menggunakannya di server Transfer Family. Anda harus menghapus server terlebih dahulu, dan kemudian Anda dapat menghapus direktori.

Memberikan akses ke grup

Setelah Anda membuat server, Anda harus memilih grup mana di direktori yang harus memiliki akses untuk mengunggah dan mengunduh file melalui protokol yang diaktifkan menggunakan. AWS Transfer Family Anda melakukan ini dengan membuat akses.

> **Note**
>
> Pengguna harus menjadi bagian langsung dari grup tempat Anda memberikan akses. Misalnya, asumsikan bahwa Bob adalah pengguna dan dia milik GroupA, dan GroupA sendiri termasuk dalam GroupB.
>
> - Jika Anda memberikan akses ke GroUpa, Bob diberikan akses.
> - Jika Anda memberikan akses ke GroupB (dan bukan ke GroupA), Bob tidak memiliki akses.

Untuk memberikan akses ke grup

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Arahkan ke halaman detail server Anda.
3. Di bagian Accesses, pilih Tambah akses.
4. Masukkan SID untuk AWS Managed Microsoft AD direktori yang ingin Anda akses ke server ini.

   > **Note**
   >
   > Untuk informasi tentang cara menemukan SID untuk grup Anda, lihatthe section called “Sebelum Anda mulai menggunakan AWS Directory Service for Microsoft Active Directory”.

5. Untuk Access, pilih peran AWS Identity and Access Management (IAM) untuk grup.

6. Di bagian Kebijakan, pilih kebijakan. Pengaturan defaultnya adalah None.
7. Untuk direktori Home, pilih bucket S3 yang sesuai dengan direktori home grup.

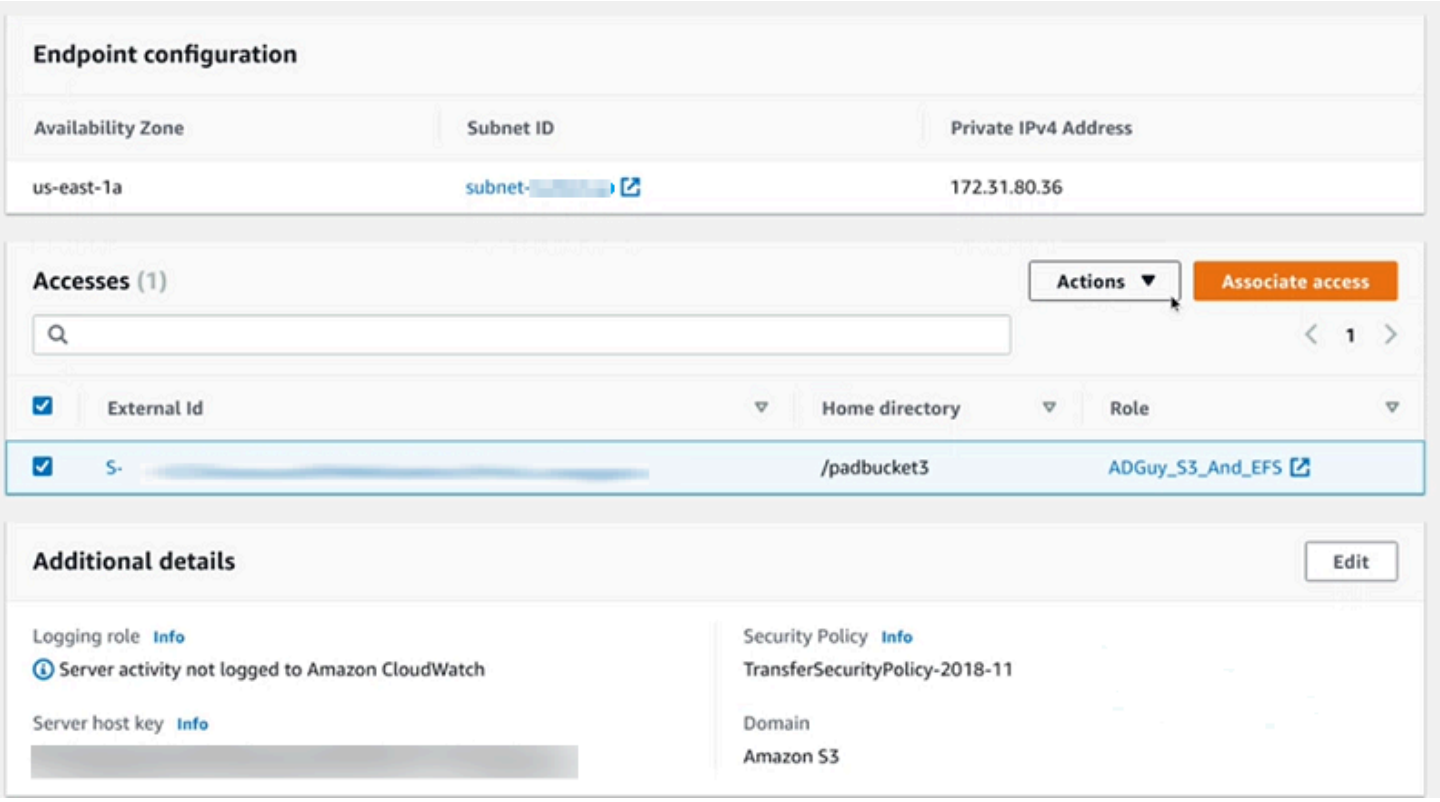
> **Note**
>
> Anda dapat membatasi bagian bucket yang dilihat pengguna dengan membuat kebijakan sesi. Misalnya, untuk membatasi pengguna ke folder mereka sendiri di bawah `/filetest` direktori, masukkan teks berikut di dalam kotak.
>
> ```
> /filetest/${transfer:UserName}
> ```
>
> Untuk mempelajari selengkapnya tentang membuat kebijakan sesi, lihatMembuat kebijakan sesi untuk bucket Amazon S3.

8. Pilih Tambah untuk membuat asosiasi.
9. Pilih server Anda.
10. Pilih Tambahkan akses.
    - Masukkan SID untuk grup.

    > **Note**
    >
    > Untuk informasi tentang cara menemukan SID, lihatthe section called “Sebelum Anda mulai menggunakan AWS Directory Service for Microsoft Active Directory”.

11. Pilih Tambahkan akses.

Di bagian Accesses, akses untuk server terdaftar.



Menguji pengguna

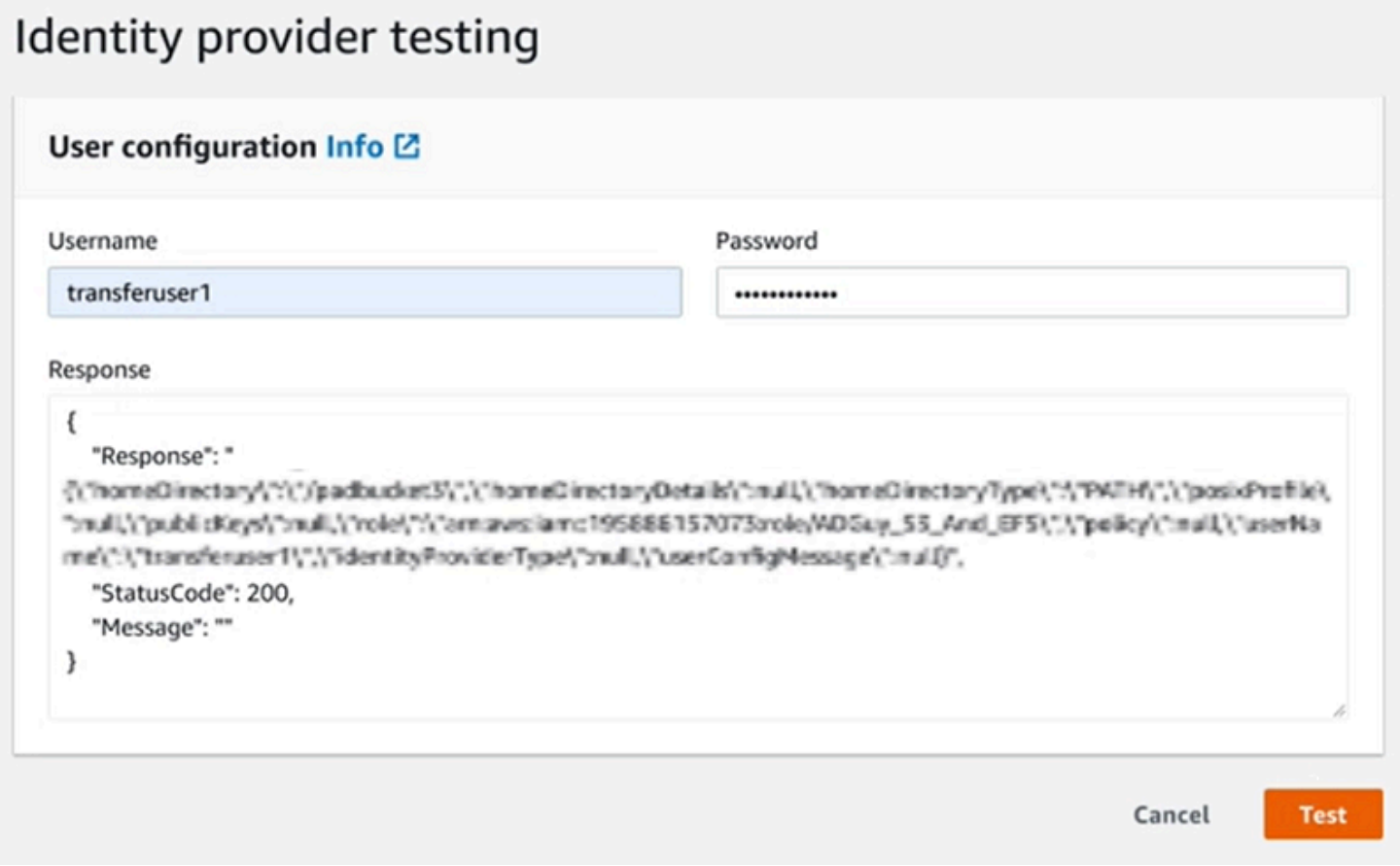
Anda dapat menguji apakah pengguna memiliki akses ke AWS Managed Microsoft AD direktori untuk server Anda.

> **Note**
>
> Seorang pengguna harus berada dalam satu grup (ID eksternal) yang tercantum di bagian Access pada halaman konfigurasi Endpoint. Jika pengguna tidak berada dalam grup, atau berada di lebih dari satu grup, pengguna tersebut tidak diberikan akses.

Untuk menguji apakah pengguna tertentu memiliki akses

1. Pada halaman detail server, pilih Tindakan, lalu pilih Uji.
2. Untuk pengujian penyedia Identitas, masukkan kredensi masuk untuk pengguna yang berada di salah satu grup yang memiliki akses.
3. Pilih Uji.

Anda melihat tes penyedia identitas yang berhasil, menunjukkan bahwa pengguna yang dipilih telah diberikan akses ke server.



Jika pengguna termasuk dalam lebih dari satu grup yang memiliki akses, Anda menerima tanggapan berikut.

```
"Response":"",
"StatusCode":200,
"Message":"More than one associated access found for user's groups."
```

Menghapus akses server untuk grup

Untuk menghapus akses server untuk grup

1. Pada halaman detail server, pilih Tindakan, lalu pilih Hapus Akses.
2. Di kotak dialog, konfirmasikan bahwa Anda ingin menghapus akses untuk grup ini.

Ketika Anda kembali ke halaman detail server, Anda melihat bahwa akses untuk grup ini tidak lagi terdaftar.

Menghubungkan ke server menggunakan SSH (Secure Shell)

Setelah Anda mengkonfigurasi server dan pengguna Anda, Anda dapat terhubung ke server menggunakan SSH dan menggunakan nama pengguna yang sepenuhnya memenuhi syarat untuk pengguna yang memiliki akses.

```
sftp user@active-directory-domain@vpc-endpoint
```

Misalnya: `transferuserexample@mycompany.com@vpce-0123456abcdef-789xyz.vpc-svc-987654zyxabc.us-east-1.vpce.amazonaws.com`.

Format ini menargetkan pencarian federasi, membatasi pencarian Direktori Aktif yang berpotensi besar.

> **Note**
>
> Anda dapat menentukan nama pengguna sederhana. Namun, dalam hal ini, kode Active Directory harus mencari semua direktori di federasi. Ini mungkin membatasi pencarian, dan otentikasi mungkin gagal bahkan jika pengguna harus memiliki akses.

Setelah mengautentikasi, pengguna berada di direktori home yang Anda tentukan saat Anda mengonfigurasi pengguna.

Menghubungkan AWS Transfer Family ke Active Directory yang dikelola sendiri menggunakan hutan dan trust

Pengguna di Active Directory (AD) yang dikelola sendiri juga dapat digunakan AWS IAM Identity Center untuk akses masuk tunggal ke dan Transfer Akun AWS Family server. Untuk melakukan itu, AWS Directory Service sediakan opsi berikut:

- Kepercayaan hutan satu arah (keluar dari AWS Managed Microsoft AD dan masuk untuk Active Directory lokal) hanya berfungsi untuk domain root.
- Untuk domain anak, Anda dapat menggunakan salah satu dari berikut ini:
  - Gunakan kepercayaan dua arah antara Active AWS Managed Microsoft AD Directory dan lokal
  - Gunakan kepercayaan eksternal satu arah untuk setiap domain anak.

Saat menghubungkan ke server menggunakan domain tepercaya, pengguna perlu menentukan domain tepercaya, misalnya`transferuserexample@mycompany.com`.

Menggunakan AWS Directory Service untuk Azure Active Directory Domain Services

- Untuk memanfaatkan hutan Active Directory yang ada untuk kebutuhan Transfer SFTP Anda, Anda dapat menggunakan Active Directory Connector.
- Jika Anda menginginkan manfaat Active Directory dan ketersediaan tinggi dalam layanan yang dikelola sepenuhnya, Anda dapat menggunakannya AWS Directory Service for Microsoft Active Directory. Untuk detailnya, lihat Menggunakan penyedia identitas AWS Directory Service.

Topik ini menjelaskan cara menggunakan Konektor Direktori Aktif dan Layanan Domain Direktori Aktif Azure (Azure ADDS) untuk mengautentikasi pengguna Transfer SFTP dengan Azure Active Directory.

Topik

- Sebelum Anda mulai menggunakan AWS Directory Service untuk Azure Active Directory Domain Services
- Langkah 1: Menambahkan Layanan Domain Direktori Aktif Azure
- Langkah 2: Membuat akun layanan
- Langkah 3: Menyiapkan AWS Direktori menggunakan AD Connector
- Langkah 4: Menyiapkan AWS Transfer Family server
- Langkah 5: Memberikan akses ke grup
- Langkah 6: Menguji pengguna

Sebelum Anda mulai menggunakan AWS Directory Service untuk Azure Active Directory Domain Services

Untuk AWS, Anda memerlukan yang berikut ini:

- Virtual Private Cloud (VPC) di AWS wilayah tempat Anda menggunakan server Transfer Family
- Setidaknya dua subnet pribadi di VPC Anda
- VPC harus memiliki konektivitas internet
- Gateway pelanggan dan gateway pribadi Virtual untuk koneksi site-to-site VPN dengan Microsoft Azure

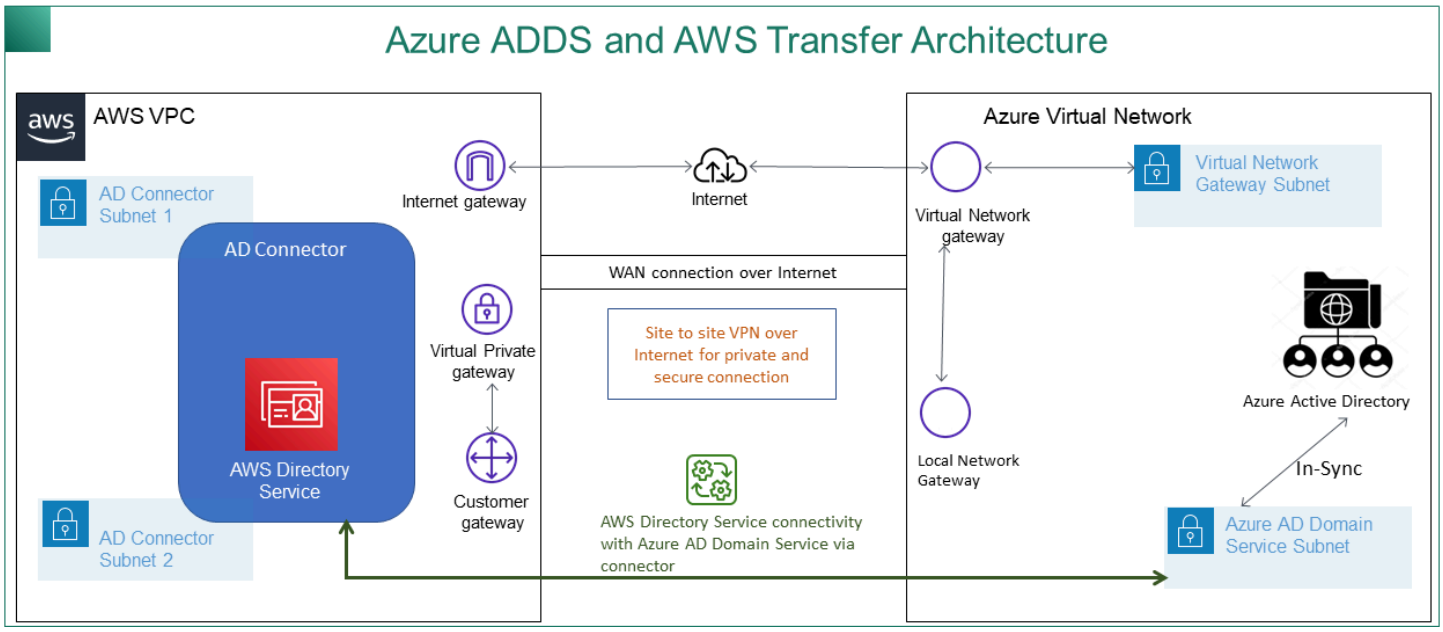
Untuk Microsoft Azure, Anda memerlukan yang berikut ini:

- Layanan domain Direktori Aktif Azure dan direktori Aktif (Azure ADDS)
- Grup sumber daya Azure
- Jaringan virtual Azure
- Konektivitas VPN antara VPC Amazon Anda dan grup sumber daya Azure Anda

> **Note**
>
> Ini bisa melalui terowongan IPSEC asli atau menggunakan peralatan VPN. Dalam topik ini, kami menggunakan terowongan IPSEC antara gateway jaringan Azure Virtual dan gateway jaringan lokal. Terowongan harus dikonfigurasi untuk memungkinkan lalu lintas antara titik akhir Azure ADDS Anda dan subnet yang menampung VPC Anda. AWS

- Gateway pelanggan dan gateway pribadi Virtual untuk koneksi site-to-site VPN dengan Microsoft Azure

Diagram berikut menunjukkan konfigurasi yang diperlukan sebelum Anda mulai.



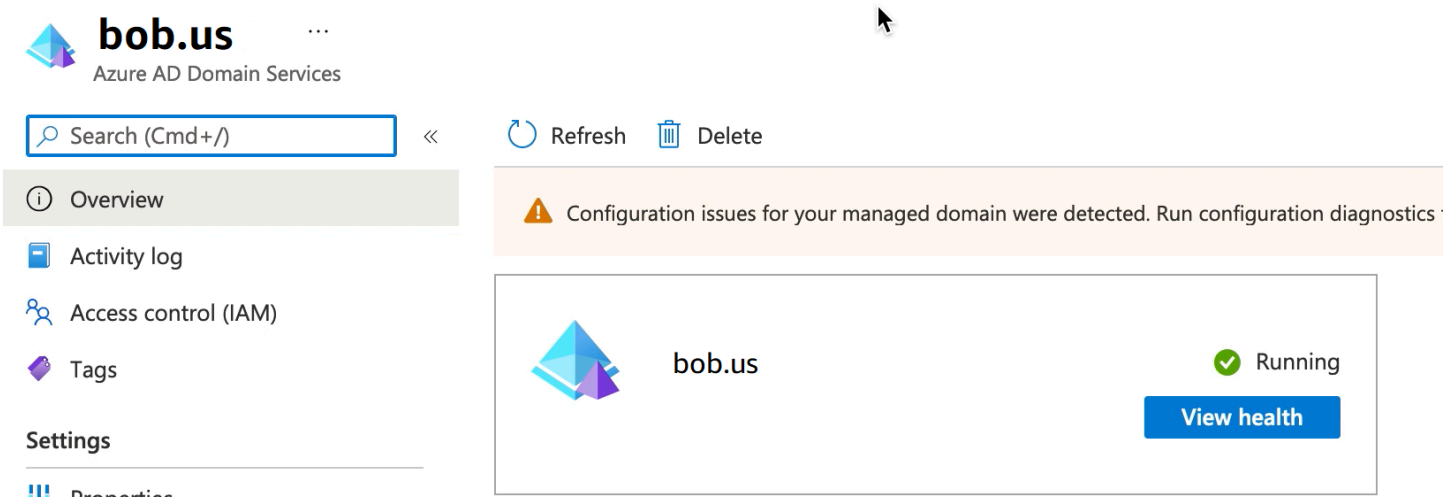
Langkah 1: Menambahkan Layanan Domain Direktori Aktif Azure

Azure AD tidak mendukung instans penggabungan Domain secara default. Untuk melakukan tindakan seperti Gabung Domain, dan untuk menggunakan alat seperti Kebijakan Grup, administrator harus mengaktifkan Azure Active Directory Domain Services. Jika Anda belum menambahkan

Azure AD DS, atau implementasi yang ada tidak terkait dengan domain yang ingin digunakan server Transfer SFTP, Anda harus menambahkan instance baru.

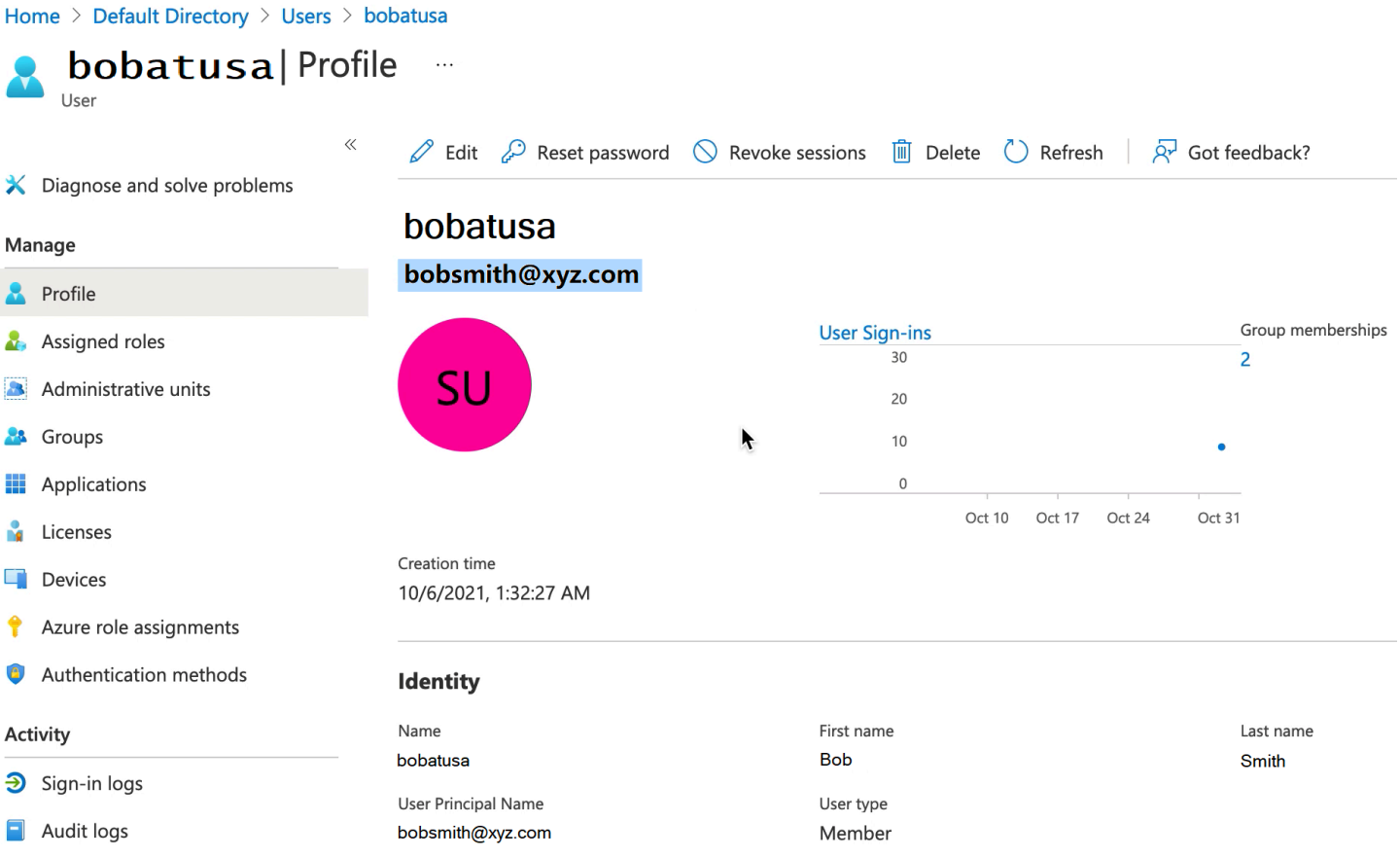
Untuk informasi tentang mengaktifkan Azure Active Directory Domain Services (Azure ADDS), lihat Tutorial: Membuat dan mengonfigurasi domain terkelola Azure Active Directory Domain Services.

> **Note**
>
> Saat Anda mengaktifkan Azure ADDS, pastikan itu dikonfigurasi untuk grup sumber daya dan domain Azure AD tempat Anda menghubungkan server Transfer SFTP Anda.



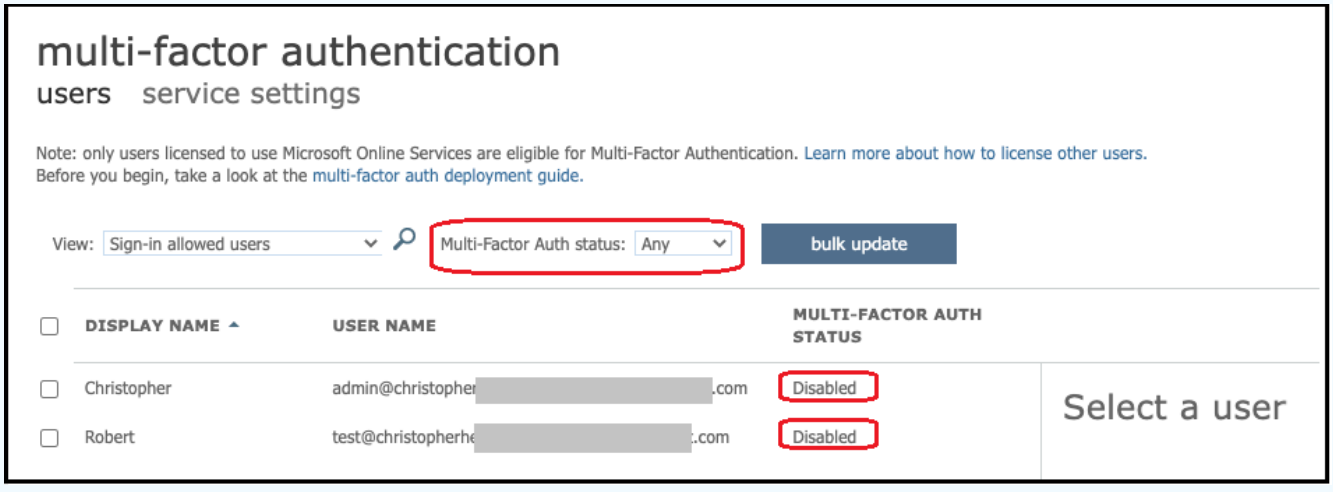
Langkah 2: Membuat akun layanan

Azure AD harus memiliki satu akun layanan yang merupakan bagian dari grup Admin di Azure ADDS. Akun ini digunakan dengan konektor AWS Active Directory. Pastikan akun ini sinkron dengan Azure ADDS.



Tip

Autentikasi multi-faktor untuk Azure Active Directory tidak didukung untuk server Transfer Family yang menggunakan protokol SFTP. Server Transfer Family tidak dapat menyediakan token MFA setelah pengguna mengautentikasi ke SFTP. Pastikan untuk menonaktifkan MFA sebelum Anda mencoba untuk terhubung.



Langkah 3: Menyiapkan AWS Direktori menggunakan AD Connector

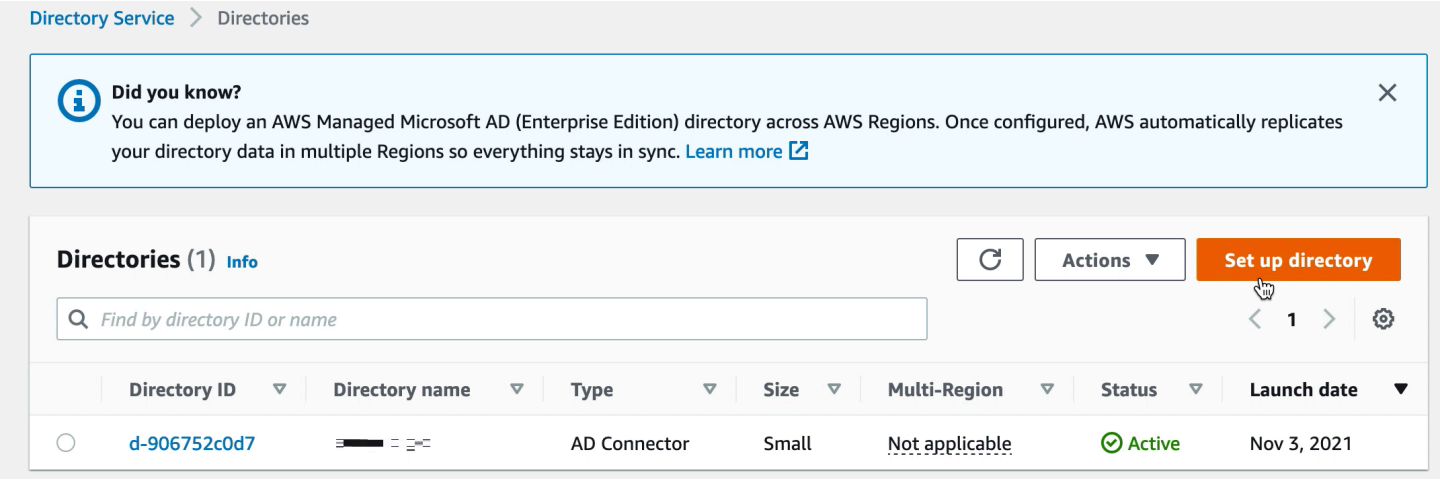
Setelah Anda mengonfigurasi Azure ADDS, dan membuat akun layanan dengan terowongan VPN IPSEC antara AWS VPC dan jaringan Virtual Azure, Anda dapat menguji konektivitas dengan melakukan ping ke alamat IP DNS Azure ADDS dari instans EC2 mana pun. AWS

Setelah Anda memverifikasi koneksi aktif, Anda dapat melanjutkan di bawah ini.

Untuk mengatur AWS Direktori menggunakan AD Connector

1. Buka konsol Directory Service dan pilih Directories.
2. Pilih Siapkan direktori.
3. Untuk jenis direktori, pilih AD Connector.
4. Pilih ukuran direktori, pilih Berikutnya, lalu pilih VPC dan Subnet Anda.
5. Pilih Berikutnya, lalu isi kolom sebagai berikut:
   - Nama DNS direktori: masukkan nama domain yang Anda gunakan untuk Azure ADDS Anda.
   - Alamat IP DNS: masukkan alamat IP Azure ADD Anda.
   - Nama pengguna dan kata sandi akun server: masukkan detail untuk akun layanan yang Anda buat di Langkah 2: Buat akun layanan.
6. Lengkapi layar untuk membuat layanan direktori.

Sekarang status direktori harus Aktif, dan siap digunakan dengan server Transfer SFTP.



Langkah 4: Menyiapkan AWS Transfer Family server

Buat server Transfer Family dengan protokol SFTP, dan jenis penyedia identitas AWS Directory Service. Dari daftar drop-down Directory, pilih direktori yang Anda tambahkan di Langkah 3: Setup AWS Directory menggunakan AD Connector.

> **Note**
>
> Anda tidak dapat menghapus direktori Microsoft AD di AWS Directory Service jika Anda menggunakannya di server Transfer Family. Anda harus menghapus server terlebih dahulu, dan kemudian Anda dapat menghapus direktori.

Langkah 5: Memberikan akses ke grup

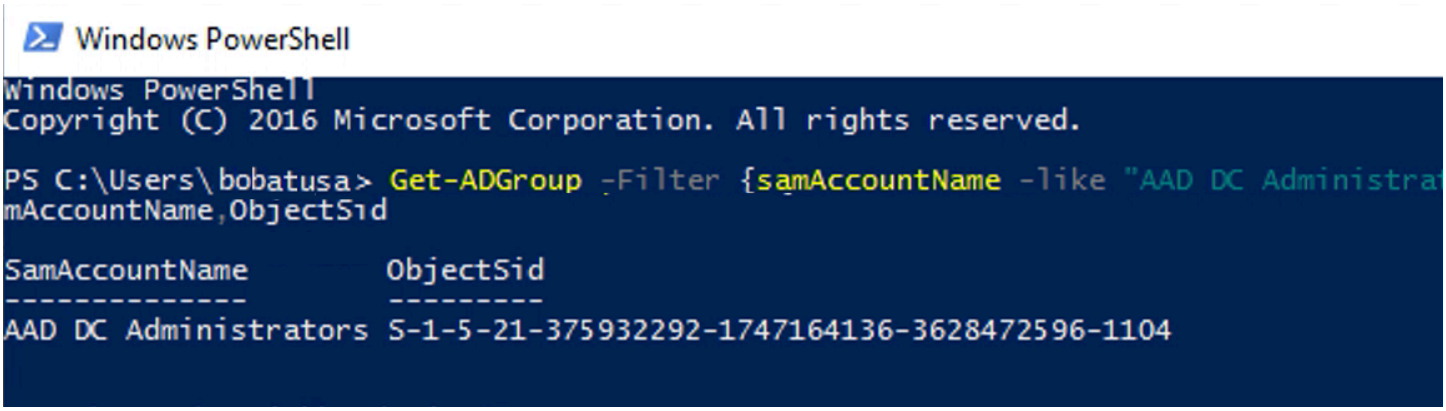
Setelah Anda membuat server, Anda harus memilih grup mana di direktori yang harus memiliki akses untuk mengunggah dan mengunduh file melalui protokol yang diaktifkan menggunakan. AWS Transfer Family Anda melakukan ini dengan membuat akses.

> **Note**
>
> Pengguna harus menjadi bagian langsung dari grup tempat Anda memberikan akses. Misalnya, asumsikan bahwa Bob adalah pengguna dan dia milik GroupA, dan GroupA sendiri termasuk dalam GroupB.
>
> - Jika Anda memberikan akses ke GroUpa, Bob diberikan akses.
> - Jika Anda memberikan akses ke GroupB (dan bukan ke GroupA), Bob tidak memiliki akses.

Untuk memberikan akses, Anda perlu mengambil SID untuk grup.

Gunakan PowerShell perintah Windows berikut untuk mengambil SID untuk grup, ganti *YourGroupName*dengan nama grup.

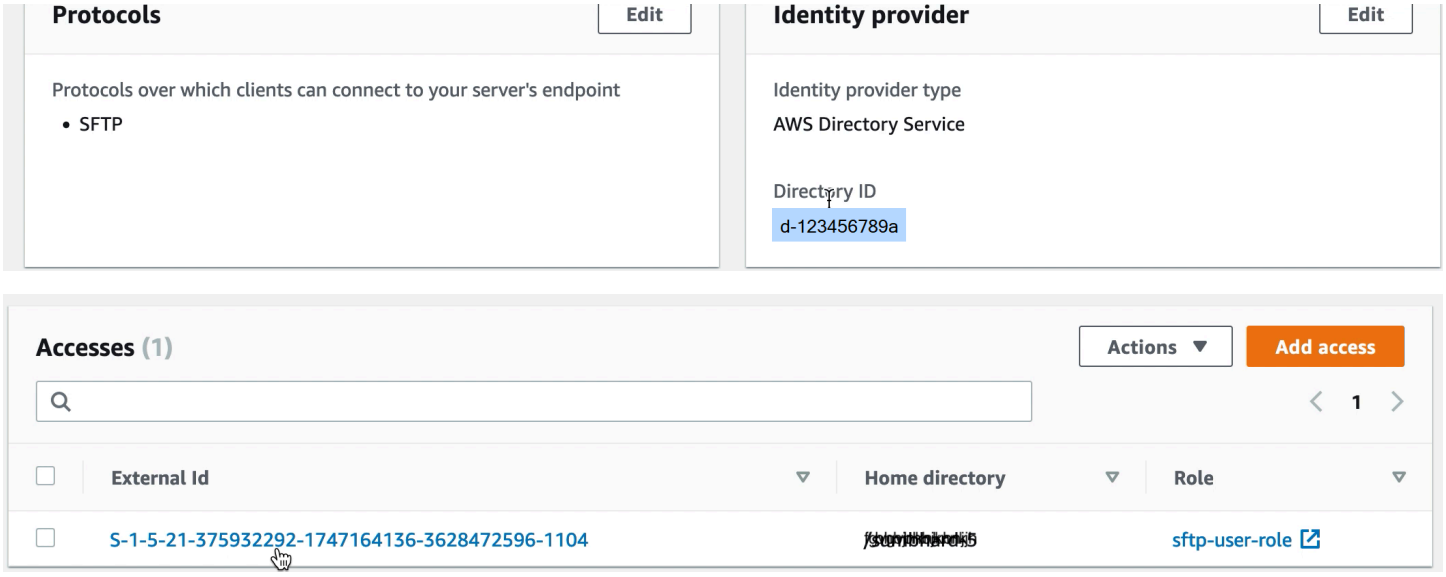
```
Get-ADGroup -Filter {samAccountName -like "YourGroupName*"} -Properties * | Select
 SamAccountName,ObjectSid
```



Berikan akses ke grup

1. Buka https://console.aws.amazon.com/transfer/.
2. Arahkan ke halaman detail server Anda dan di bagian Akses, pilih Tambahkan akses.
3. Masukkan SID yang Anda terima dari output dari prosedur sebelumnya.
4. Untuk Access, pilih AWS Identity and Access Management peran untuk grup.
5. Di bagian Kebijakan, pilih kebijakan. Nilai defaultnya adalah None.
6. Untuk direktori Home, pilih bucket S3 yang sesuai dengan direktori home grup.
7. Pilih Tambah untuk membuat asosiasi.

Detail dari server Transfer Anda akan terlihat mirip dengan yang berikut ini:



Langkah 6: Menguji pengguna

Anda dapat menguji (Menguji pengguna) apakah pengguna memiliki akses ke AWS Managed Microsoft AD direktori untuk server Anda. Seorang pengguna harus berada dalam satu grup (ID eksternal) yang tercantum di bagian Access pada halaman konfigurasi Endpoint. Jika pengguna tidak berada dalam grup, atau berada di lebih dari satu grup, pengguna tersebut tidak diberikan akses.

Bekerja dengan penyedia identitas khusus

Untuk mengautentikasi pengguna, Anda dapat menggunakan penyedia identitas yang ada dengan AWS Transfer Family. Anda mengintegrasikan penyedia identitas Anda menggunakan AWS Lambda fungsi, yang mengautentikasi dan memberi wewenang kepada pengguna Anda untuk mengakses Amazon S3 atau Amazon Elastic File System (Amazon EFS). Untuk detailnya, lihat Menggunakan AWS Lambda untuk mengintegrasikan penyedia identitas Anda. Anda juga dapat mengakses CloudWatch grafik untuk metrik seperti jumlah file dan byte yang ditransfer di AWS Transfer Family Management Console, memberi Anda satu panel kaca untuk memantau transfer file menggunakan dasbor terpusat.

Atau, Anda dapat menyediakan antarmuka RESTful dengan satu metode Amazon API Gateway. Transfer Family memanggil metode ini untuk terhubung ke penyedia identitas Anda, yang mengautentikasi dan memberi wewenang kepada pengguna Anda untuk mengakses Amazon S3 atau Amazon EFS. Gunakan opsi ini jika Anda memerlukan RESTful API untuk mengintegrasikan penyedia identitas Anda atau jika Anda ingin menggunakannya untuk memanfaatkan kemampuannya AWS WAF untuk permintaan pemblokiran geografis atau pembatasan laju. Untuk detailnya, lihat Menggunakan Amazon API Gateway untuk mengintegrasikan penyedia identitas Anda.

Dalam kedua kasus, Anda dapat membuat server baru menggunakan AWS Transfer Family konsol atau operasi CreateServerAPI.

> **Note**
>
> Kami memiliki lokakarya yang dapat Anda hadiri, di mana Anda dapat membangun solusi transfer file. Solusi ini memanfaatkan AWS Transfer Family endpoint SFTP/FTPS terkelola serta Amazon Cognito dan DynamoDB untuk manajemen pengguna. Anda dapat melihat detail untuk lokakarya ini di sini.

AWS Transfer Family menyediakan opsi berikut untuk bekerja dengan penyedia identitas khusus.

- Gunakan AWS Lambda untuk menghubungkan penyedia identitas Anda — Anda dapat menggunakan penyedia identitas yang ada, didukung oleh fungsi Lambda. Anda memberikan nama fungsi Lambda. Untuk informasi selengkapnya, lihat Menggunakan AWS Lambda untuk mengintegrasikan penyedia identitas Anda.
- Gunakan Amazon API Gateway untuk menghubungkan penyedia identitas Anda — Anda dapat membuat metode API Gateway yang didukung oleh fungsi Lambda untuk digunakan sebagai penyedia identitas. Anda menyediakan URL Amazon API Gateway dan peran pemanggilan. Untuk informasi selengkapnya, lihat Menggunakan Amazon API Gateway untuk mengintegrasikan penyedia identitas Anda.

Untuk salah satu opsi, Anda juga dapat menentukan cara mengautentikasi.

- Kata Sandi ATAU Kunci — pengguna dapat mengautentikasi dengan kata sandi atau kunci mereka. Ini adalah nilai default.
- Hanya kata sandi — pengguna harus memberikan kata sandi mereka untuk terhubung.
- Hanya kunci — pengguna harus menyediakan kunci pribadi mereka untuk terhubung.
- Kata Sandi dan Kunci — pengguna harus memberikan kunci pribadi dan kata sandi mereka untuk terhubung. Server memeriksa kunci terlebih dahulu, dan kemudian jika kuncinya valid, sistem meminta kata sandi. Jika kunci pribadi yang disediakan tidak cocok dengan kunci publik yang disimpan, otentikasi gagal.

Menggunakan beberapa metode otentikasi untuk mengautentikasi dengan penyedia identitas kustom Anda

Server Transfer Family mengontrol logika AND saat Anda menggunakan beberapa metode otentikasi. Transfer Family memperlakukan ini sebagai dua permintaan terpisah ke penyedia identitas kustom Anda: namun, efeknya digabungkan.

Kedua permintaan harus berhasil dikembalikan dengan respons yang benar untuk memungkinkan otentikasi selesai. Transfer Family mengharuskan dua tanggapan lengkap, artinya berisi semua elemen yang diperlukan (peran, direktori beranda, kebijakan, dan profil POSIX jika Anda menggunakan Amazon EFS untuk penyimpanan). Transfer Family juga mensyaratkan bahwa respons kata sandi tidak boleh menyertakan kunci publik.

Permintaan kunci publik harus memiliki respons terpisah dari penyedia identitas. Perilaku itu tidak berubah saat menggunakan Kata Sandi ATAU Kunci atau Kata Sandi DAN Kunci.

Protokol SSH/SFTP menantang klien perangkat lunak terlebih dahulu dengan otentikasi kunci publik, kemudian meminta otentikasi kata sandi. Operasi ini mengamanatkan keduanya berhasil sebelum pengguna diizinkan untuk menyelesaikan otentikasi.

Topik

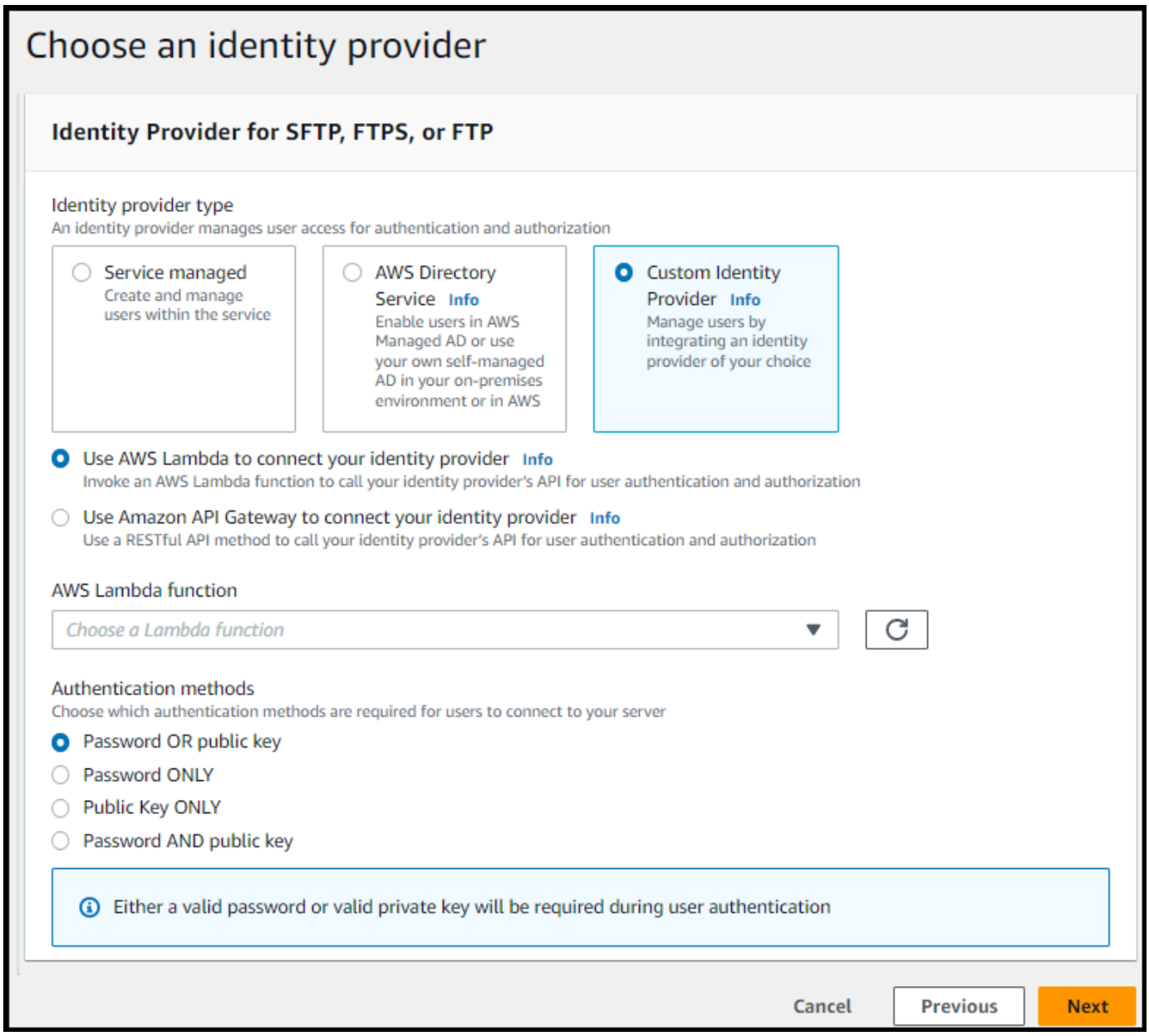
- Menggunakan AWS Lambda untuk mengintegrasikan penyedia identitas Anda
- Menggunakan Amazon API Gateway untuk mengintegrasikan penyedia identitas Anda

Menggunakan AWS Lambda untuk mengintegrasikan penyedia identitas Anda

Buat AWS Lambda fungsi yang terhubung ke penyedia identitas kustom Anda. Anda dapat menggunakan penyedia identitas kustom apa pun, seperti Okta, Secrets Manager OneLogin, atau penyimpanan data khusus yang menyertakan logika otorisasi dan otentikasi.

> **Note**
>
> Sebelum membuat server Transfer Family yang menggunakan Lambda sebagai penyedia identitas, Anda harus membuat fungsinya. Untuk contoh fungsi Lambda, lihat. Contoh fungsi Lambda Atau, Anda dapat menerapkan CloudFormation tumpukan yang menggunakan salah satu. Template fungsi Lambda Selain itu, pastikan fungsi Lambda Anda menggunakan kebijakan berbasis sumber daya yang mempercayai Transfer Family. Untuk contoh kebijakan, lihat Kebijakan berbasis sumber daya Lambda.

1. Buka konsol AWS Transfer Family.
2. Pilih Buat server untuk membuka halaman Buat server. Untuk Pilih penyedia identitas, pilih Penyedia Identitas Kustom, seperti yang ditunjukkan pada gambar berikut.



> **Note**
>
> Pilihan metode otentikasi hanya tersedia jika Anda mengaktifkan SFTP sebagai salah satu protokol untuk server Transfer Family Anda.

3. Pastikan nilai default, Gunakan AWS Lambda untuk menghubungkan penyedia identitas Anda, dipilih.
4. Untuk AWS Lambda fungsi, pilih nama fungsi Lambda Anda.

5. Isi kotak yang tersisa, lalu pilih Buat server. Untuk detail tentang langkah-langkah yang tersisa untuk membuat server, lihatMengkonfigurasi titik akhir server SFTP, FTPS, atau FTP.

Kebijakan berbasis sumber daya Lambda

Anda harus memiliki kebijakan yang mereferensikan server Transfer Family dan Lambda ARN. Misalnya, Anda dapat menggunakan kebijakan berikut dengan fungsi Lambda yang terhubung ke penyedia identitas Anda. Kebijakan ini lolos dari JSON sebagai string.

```
"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**
>
> Dalam contoh kebijakan di atas, ganti setiap *placeholder masukan pengguna dengan informasi* Anda sendiri.

Struktur pesan peristiwa

Struktur pesan acara dari server SFTP yang dikirim ke fungsi Lambda otorisasi untuk IDP kustom adalah sebagai berikut.

```
{
    'username': 'value',
    'password': 'value',
    'protocol': 'SFTP',
    'serverId': 's-abcd123456',
    'sourceIp': '192.168.0.100'
}
```

Di mana `username` dan `password` merupakan nilai untuk kredenal masuk yang dikirim ke server.

Misalnya, Anda memasukkan perintah berikut untuk menghubungkan:

```
sftp bobusa@server_hostname
```

Anda kemudian diminta untuk memasukkan kata sandi Anda:

```
Enter password:
    mysecretpassword
```

Anda dapat memeriksa ini dari fungsi Lambda Anda dengan mencetak peristiwa yang diteruskan dari dalam fungsi Lambda. Seharusnya terlihat mirip dengan blok teks berikut.

```
{
    'username': 'bobusa',
    'password': 'mysecretpassword',
    'protocol': 'SFTP',
    'serverId': 's-abcd123456',
    'sourceIp': '192.168.0.100'
}
```

Struktur acara serupa untuk FTP dan FTPS: satu-satunya perbedaan adalah nilai-nilai tersebut digunakan untuk `protocol` parameter, bukan SFTP.

Fungsi Lambda untuk otentikasi

Untuk menerapkan strategi otentikasi yang berbeda, edit fungsi Lambda. Untuk membantu Anda memenuhi kebutuhan aplikasi Anda, Anda dapat menerapkan CloudFormation tumpukan.

Untuk informasi selengkapnya tentang Lambda, lihat Panduan AWS Lambda Pengembang atau Membangun fungsi Lambda dengan Node.js.

Topik

- Template fungsi Lambda
- Nilai Lambda yang valid
- Contoh fungsi Lambda
- Menguji konfigurasi Anda

Template fungsi Lambda

Anda dapat menerapkan AWS CloudFormation tumpukan yang menggunakan fungsi Lambda untuk otentikasi. Kami menyediakan beberapa templat yang mengautentikasi dan mengotorisasi pengguna Anda menggunakan kredensi masuk. Anda dapat memodifikasi template atau AWS Lambda kode ini untuk lebih menyesuaikan akses pengguna.

> **Note**
>
> Anda dapat membuat AWS Transfer Family server berkemampuan FIPS AWS CloudFormation dengan menentukan kebijakan keamanan berkemampuan FIPS di template Anda. Kebijakan keamanan yang tersedia dijelaskan dalam Kebijakan keamanan untuk AWS Transfer Family server

Untuk membuat AWS CloudFormation tumpukan yang akan digunakan untuk otentikasi

1. Buka AWS CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
2. Ikuti petunjuk untuk menerapkan AWS CloudFormation tumpukan dari template yang ada di Memilih template tumpukan di Panduan AWS CloudFormation Pengguna.
3. Gunakan salah satu templat berikut untuk membuat fungsi Lambda yang akan digunakan untuk otentikasi di Transfer Family.

   - Templat tumpukan klasik (Amazon Cognito)

     Template dasar untuk membuat AWS Lambda untuk digunakan sebagai penyedia identitas kustom di AWS Transfer Family. Ini mengautentikasi terhadap Amazon Cognito untuk otentikasi berbasis kata sandi dan kunci publik dikembalikan dari bucket Amazon S3 jika

otentikasi berbasis kunci publik digunakan. Setelah penerapan, Anda dapat memodifikasi kode fungsi Lambda untuk melakukan sesuatu yang berbeda.

- AWS Secrets Manager template tumpukan

  Template dasar yang digunakan AWS Lambda dengan AWS Transfer Family server untuk mengintegrasikan Secrets Manager sebagai penyedia identitas. Ini mengotentikasi terhadap entri dalam AWS Secrets Manager format`aws/transfer/`*`server-id`*`/`*`username`*. Selain itu, secret harus menyimpan pasangan kunci-nilai untuk semua properti pengguna yang dikembalikan ke Transfer Family. Setelah penerapan, Anda dapat memodifikasi kode fungsi Lambda untuk melakukan sesuatu yang berbeda.
- Template tumpukan Okta: Template dasar yang digunakan AWS Lambda dengan AWS Transfer Family server untuk mengintegrasikan Okta sebagai penyedia identitas khusus.
- Template tumpukan Okta-MFA: Template dasar yang digunakan AWS Lambda dengan AWS Transfer Family server untuk mengintegrasikan Okta, dengan MultiFactor Otentikasi, sebagai penyedia identitas khusus.
- Template Azure Active Directory: detail untuk tumpukan ini dijelaskan dalam posting blog Mengautentikasi AWS Transfer Family dengan Azure Active Directory dan. AWS Lambda

Setelah tumpukan digunakan, Anda dapat melihat detailnya di tab Output di CloudFormation konsol.

Menerapkan salah satu tumpukan ini adalah cara termudah untuk mengintegrasikan penyedia identitas kustom ke dalam alur kerja Transfer Family.

Nilai Lambda yang valid

Tabel berikut menjelaskan detail nilai yang diterima Transfer Family untuk fungsi Lambda yang digunakan untuk penyedia identitas kustom.

Nilai	Deskripsi	Wajib
`Role`	Menentukan Nama Sumber Daya Amazon (ARN) peran IAM yang mengontrol akses pengguna ke bucket Amazon S3 atau sistem file Amazon	Wajib

Nilai	Deskripsi	Wajib
	EFS. Kebijakan yang dilampirkan pada peran ini menentukan tingkat akses yang ingin Anda berikan kepada pengguna saat mentransfer file masuk dan keluar dari sistem file Amazon S3 atau Amazon EFS Anda. IAM role juga harus berisi hubungan kepercayaan yang mengizinkan server untuk mengakses sumber daya Anda saat melayani permintaan transfer pengguna. Untuk detail tentang membangun hubungan kepercayaan, lihatUntuk membangun hubungan kepercayaan.	
`PosixProfile`	Identitas POSIX lengkap, termasuk ID pengguna (`Uid`), ID grup (`Gid`), dan ID grup sekunder (`SecondaryGids`) apa pun, yang mengontrol akses pengguna ke sistem file Amazon EFS Anda. POSIX izin yang ditetapkan pada file dan direktori dalam sistem file Anda menentukan tingkat akses pengguna Anda mendapatkan ketika mentransfer file ke dalam dan keluar dari sistem file Amazon EFS Anda.	Diperlukan untuk penyimpanan dukungan Amazon EFS

Nilai	Deskripsi	Wajib
`PublicKeys`	Daftar nilai kunci publik SSH yang valid untuk pengguna ini. Daftar kosong menyiratkan bahwa ini bukan login yang valid. Tidak boleh dikembalikan selama otentikasi kata sandi.	Opsional
`Policy`	Kebijakan sesi untuk pengguna Anda sehingga Anda dapat menggunakan peran IAM yang sama di beberapa pengguna. Kebijakan ini mencakup bawah akses pengguna ke bagian dari bucket Amazon S3 mereka.	Opsional

Nilai	Deskripsi	Wajib
`HomeDirectoryType`	Jenis direktori pendaratan (folder) yang Anda inginkan direktori home pengguna Anda ketika mereka masuk ke server. • Jika Anda mengaturn yaPATH, pengguna akan melihat bucket Amazon S3 absolut atau jalur Amazon EFS seperti pada klien protokol transfer file mereka. • Jika Anda menyeteln yaLOGICAL, Anda harus menyediakan pemetaan dalam HomeDirec toryDetails parameter agar jalur Amazon S3 atau Amazon EFS terlihat oleh pengguna Anda.	Opsional

Nilai	Deskripsi	Wajib
`HomeDirectoryDetails`	Pemetaan direktori logis yang menentukan jalur dan kunci Amazon S3 atau Amazon EFS mana yang harus terlihat oleh pengguna Anda dan bagaimana Anda ingin membuatnya terlihat. Anda harus menentukan `Entry` dan `Target` memasangkan, di mana `Entry` menunjukkan bagaimana jalur dibuat terlihat dan `Target` merupakan jalur Amazon S3 atau Amazon EFS yang sebenarnya.	Diperlukan jika `HomeDirectoryType` memiliki nilai `LOGICAL`
`HomeDirectory`	Direktori pendaratan untuk pengguna ketika mereka masuk ke server menggunakan klien.	Opsional

> **Note**
>
> `HomeDirectoryDetails`adalah representasi string dari peta JSON. Hal ini berbeda dengan`PosixProfile`, yang merupakan objek peta JSON yang sebenarnya, dan `PublicKeys` yang merupakan array JSON string. Lihat contoh kode untuk detail khusus bahasa.

Contoh fungsi Lambda

Bagian ini menyajikan beberapa contoh fungsi Lambda, baik di NodeJS maupun Python.

> **Note**
>
> Dalam contoh ini, detail direktori pengguna, peran, profil POSIX, kata sandi, dan home directory adalah contoh, dan harus diganti dengan nilai aktual Anda.

Logical home directory, NodeJS

Fungsi contoh NodeJS berikut memberikan rincian untuk pengguna yang memiliki direktori home logis.

```
// 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

Fungsi contoh NodeJS berikut memberikan rincian untuk pengguna yang memiliki direktori home berbasis jalur.

```
// 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

Contoh fungsi Python berikut memberikan rincian untuk pengguna yang memiliki direktori home logis.

```
# 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

Contoh fungsi Python berikut memberikan rincian untuk pengguna yang memiliki direktori home berbasis jalur.

```
# 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
```

Menguji konfigurasi Anda

Setelah Anda membuat penyedia identitas kustom Anda, Anda harus menguji konfigurasi Anda.

Console

Untuk menguji konfigurasi Anda dengan menggunakan AWS Transfer Family konsol

1. Buka konsol AWS Transfer Family.
2. Pada halaman Server, pilih server baru Anda, pilih Tindakan, lalu pilih Uji.
3. Masukkan teks untuk Nama Pengguna dan Kata Sandi yang Anda atur saat Anda menerapkan AWS CloudFormation tumpukan. Jika Anda menyimpan opsi default, nama pengguna adalah `myuser` dan kata sandinya`MySuperSecretPassword`.
4. Pilih protokol Server dan masukkan alamat IP untuk IP Sumber, jika Anda mengaturnya saat Anda menerapkan AWS CloudFormation tumpukan.

CLI

Untuk menguji konfigurasi Anda dengan menggunakan AWS CLI

1. Jalankan perintah test-identity-provider. Ganti masing-masing *`user input placeholder`* dengan informasi Anda sendiri, seperti yang dijelaskan dalam langkah-langkah selanjutnya.

```
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. Masukkan ID server.
3. Masukkan nama pengguna dan kata sandi yang Anda tetapkan saat Anda menerapkan AWS CloudFormation tumpukan. Jika Anda menyimpan opsi default, nama pengguna adalah `myuser` dan kata sandinya`MySuperSecretPassword`.
4. Masukkan protokol server dan alamat IP sumber, jika Anda mengaturnya saat Anda menerapkan AWS CloudFormation tumpukan.

Jika otentikasi pengguna berhasil, pengujian mengembalikan respons `StatusCode: 200` HTTP, string kosong `Message: ""` (yang akan berisi alasan kegagalan jika tidak), dan bidang. `Response`

> **Note**
>
> Dalam contoh respons di bawah ini, `Response` bidang adalah objek JSON yang telah "dirangkai" (diubah menjadi string JSON datar yang dapat digunakan di dalam program), dan berisi rincian peran dan izin pengguna.


```
{
    "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": ""
}
```


Menggunakan Amazon API Gateway untuk mengintegrasikan penyedia identitas Anda

Topik ini menjelaskan cara menggunakan AWS Lambda fungsi untuk mendukung metode API Gateway. Gunakan opsi ini jika Anda memerlukan RESTful API untuk mengintegrasikan penyedia identitas Anda atau jika Anda ingin menggunakannya untuk memanfaatkan kemampuannya AWS WAF untuk permintaan pemblokiran geografis atau pembatasan laju.

Batasan jika menggunakan API Gateway untuk mengintegrasikan penyedia identitas Anda

- Konfigurasi ini tidak mendukung domain kustom.
- Konfigurasi ini tidak mendukung URL API Gateway pribadi.

Jika Anda membutuhkan salah satu dari ini, Anda dapat menggunakan Lambda sebagai penyedia identitas, tanpa API Gateway. Untuk detailnya, lihat Menggunakan AWS Lambda untuk mengintegrasikan penyedia identitas Anda.

Mengautentikasi menggunakan metode API Gateway

Anda dapat membuat metode API Gateway untuk digunakan sebagai penyedia identitas untuk Transfer Family. Pendekatan ini menyediakan cara yang sangat aman bagi Anda untuk membuat dan menyediakan API. Dengan API Gateway, Anda dapat membuat titik akhir HTTPS sehingga semua panggilan API yang masuk ditransmisikan dengan keamanan yang lebih besar. Untuk detail selengkapnya tentang layanan API Gateway, lihat Panduan Pengembang API Gateway.

API Gateway menawarkan metode otorisasi bernama`AWS_IAM`, yang memberi Anda autentikasi yang sama berdasarkan AWS Identity and Access Management (IAM) yang AWS digunakan secara internal. Jika Anda mengaktifkan autentikasi`AWS_IAM`, hanya penelepon dengan izin eksplisit untuk memanggil API yang dapat mencapai metode API Gateway API tersebut.

Untuk menggunakan metode API Gateway Anda sebagai penyedia identitas khusus untuk Transfer Family, aktifkan IAM untuk metode API Gateway Anda. Sebagai bagian dari proses ini, Anda memberikan peran IAM dengan izin untuk Transfer Family untuk menggunakan gateway Anda.

> **Note**
>
> Untuk meningkatkan keamanan, Anda dapat mengkonfigurasi firewall aplikasi web. AWS WAF adalah firewall aplikasi web yang memungkinkan Anda memantau permintaan HTTP dan HTTPS yang diteruskan ke Amazon API Gateway. Untuk detailnya, lihat Tambahkan firewall aplikasi web.

Untuk menggunakan metode API Gateway Anda untuk autentikasi kustom dengan Transfer Family

1. Buat AWS CloudFormation tumpukan. Untuk melakukannya:

> **Note**
>
> Templat tumpukan telah diperbarui untuk menggunakan kata sandi yang disandikan Base64: untuk detailnya, lihat. Perbaikan AWS CloudFormation template

a. Buka AWS CloudFormation konsol di https://console.aws.amazon.com/cloudformation.

b. Ikuti petunjuk untuk menerapkan AWS CloudFormation tumpukan dari template yang ada di Memilih template tumpukan di Panduan AWS CloudFormation Pengguna.

c. Gunakan salah satu templat dasar berikut untuk membuat metode API Gateway yang AWS Lambda didukung untuk digunakan sebagai penyedia identitas kustom di Transfer Family.

   - Template tumpukan dasar

     Secara default, metode API Gateway Anda digunakan sebagai penyedia identitas khusus untuk mengautentikasi satu pengguna dalam satu server menggunakan kunci atau kata sandi SSH (Secure Shell) dengan kode keras. Setelah penerapan, Anda dapat memodifikasi kode fungsi Lambda untuk melakukan sesuatu yang berbeda.

   - AWS Secrets Manager template tumpukan

     Secara default, metode API Gateway Anda melakukan autentikasi terhadap entri di Secrets Manager format`aws/transfer/`*`server-id`*`/`*`username`*. Selain itu, secret harus menyimpan pasangan kunci-nilai untuk semua properti pengguna yang dikembalikan ke Transfer Family. Setelah penerapan, Anda dapat memodifikasi kode fungsi Lambda untuk melakukan sesuatu yang berbeda. Untuk informasi selengkapnya, lihat posting blog Aktifkan otentikasi kata sandi untuk AWS Transfer Family digunakan AWS Secrets Manager.

   - Templat tumpukan Okta

     Metode API Gateway Anda terintegrasi dengan Okta sebagai penyedia identitas khusus di Transfer Family. Untuk informasi lebih lanjut, lihat posting blog Menggunakan Okta sebagai penyedia identitas dengan AWS Transfer Family.

Menerapkan salah satu tumpukan ini adalah cara termudah untuk mengintegrasikan penyedia identitas kustom ke dalam alur kerja Transfer Family. Setiap tumpukan menggunakan fungsi Lambda untuk mendukung metode API Anda berdasarkan API Gateway. Anda kemudian dapat

menggunakan metode API Anda sebagai penyedia identitas kustom di Transfer Family. Secara default, fungsi Lambda mengautentikasi satu pengguna yang dipanggil `myuser` dengan kata sandi. `MySuperSecretPassword` Setelah penerapan, Anda dapat mengedit kredensyal ini atau memperbarui kode fungsi Lambda untuk melakukan sesuatu yang berbeda.

> **Important**
>
> Kami menyarankan Anda mengedit kredensyal pengguna dan kata sandi default.

Setelah tumpukan digunakan, Anda dapat melihat detailnya di tab Output di CloudFormation konsol. Detail ini termasuk Amazon Resource Name (ARN) stack, ARN dari peran IAM yang dibuat stack, dan URL untuk gateway baru Anda.

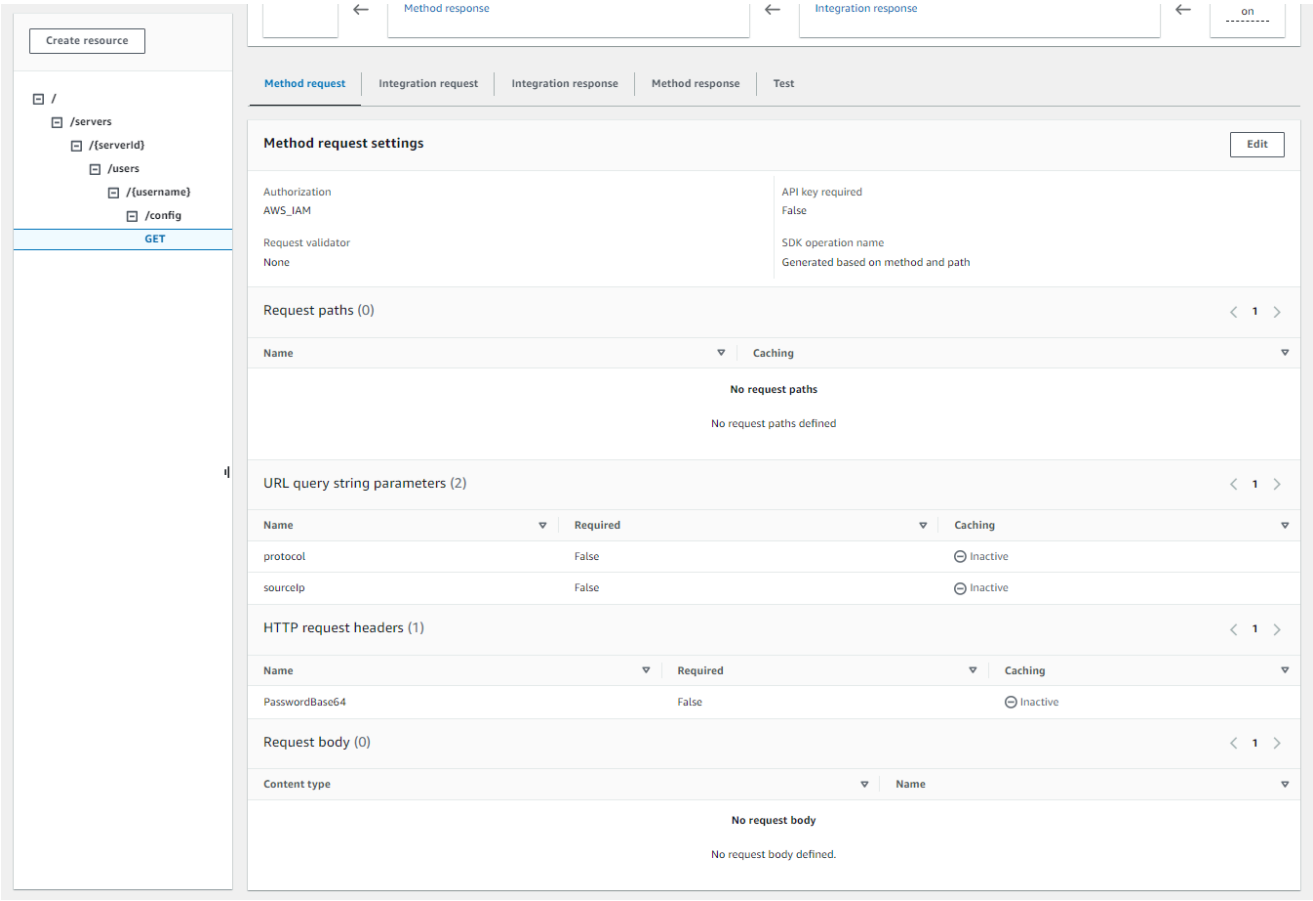
> **Note**
>
> Jika Anda menggunakan opsi penyedia identitas khusus untuk mengaktifkan autentikasi berbasis kata sandi bagi pengguna Anda, dan Anda mengaktifkan pencatatan permintaan dan respons yang disediakan oleh API Gateway, API Gateway mencatat kata sandi pengguna Anda ke Log Amazon Anda. CloudWatch Kami tidak menyarankan menggunakan log ini di lingkungan produksi Anda. Untuk informasi selengkapnya, lihat Menyiapkan pencatatan CloudWatch API di API Gateway di Panduan Pengembang API Gateway.

2. Periksa konfigurasi metode API Gateway untuk server Anda. Untuk melakukannya:

   a. Buka konsol API Gateway di https://console.aws.amazon.com/apigateway/.

   b. Pilih API template dasar Transfer Custom Identity Provider yang dihasilkan AWS CloudFormation template. Anda mungkin perlu memilih wilayah Anda untuk melihat gateway Anda.

   c. Di panel Resources, pilih GET. Tangkapan layar berikut menunjukkan konfigurasi metode yang benar.

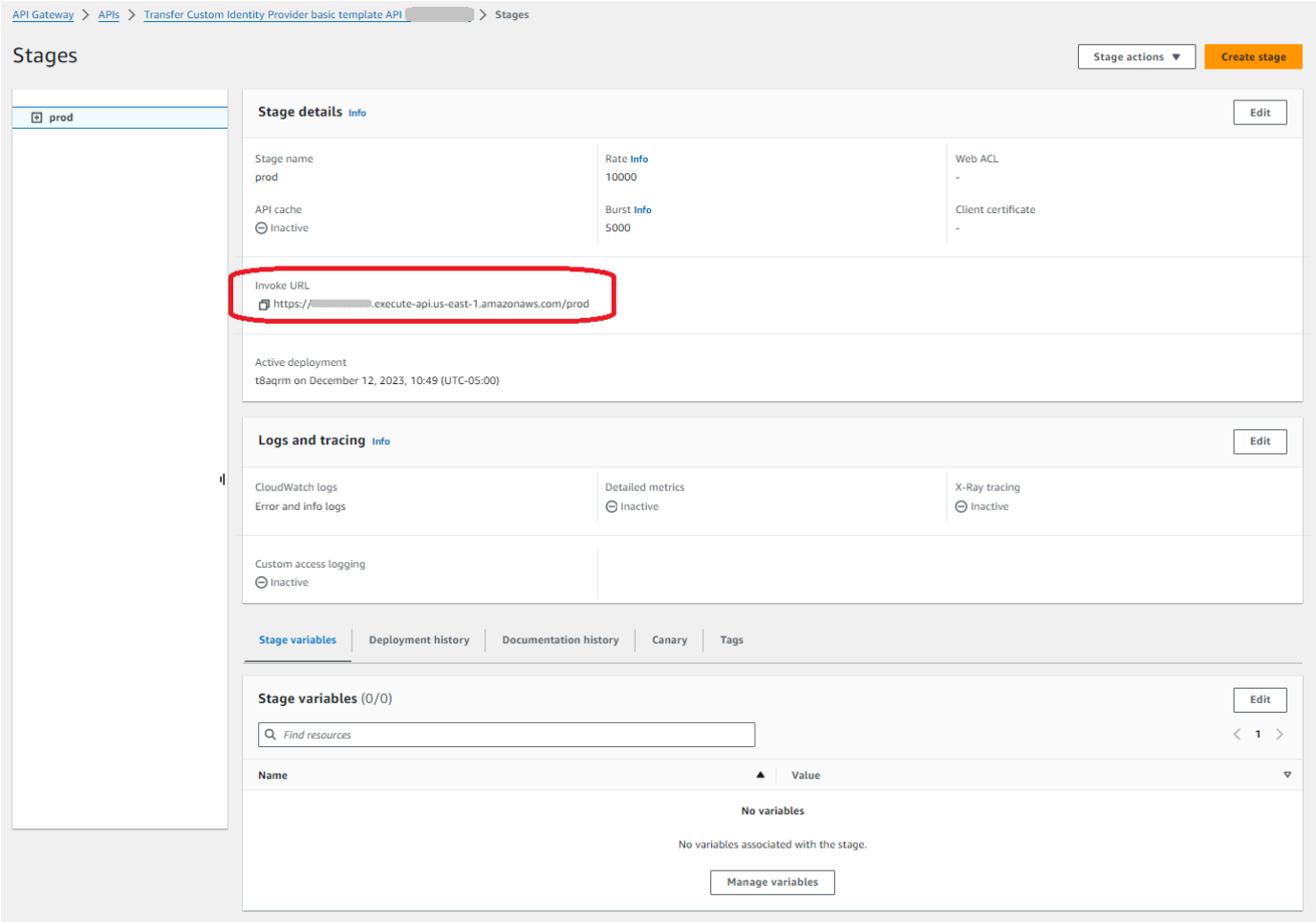


Pada titik ini, gateway API Anda siap digunakan.

3. Untuk Tindakan, pilih Deploy API. Untuk tahap Deployment, pilih prod, lalu pilih Deploy.

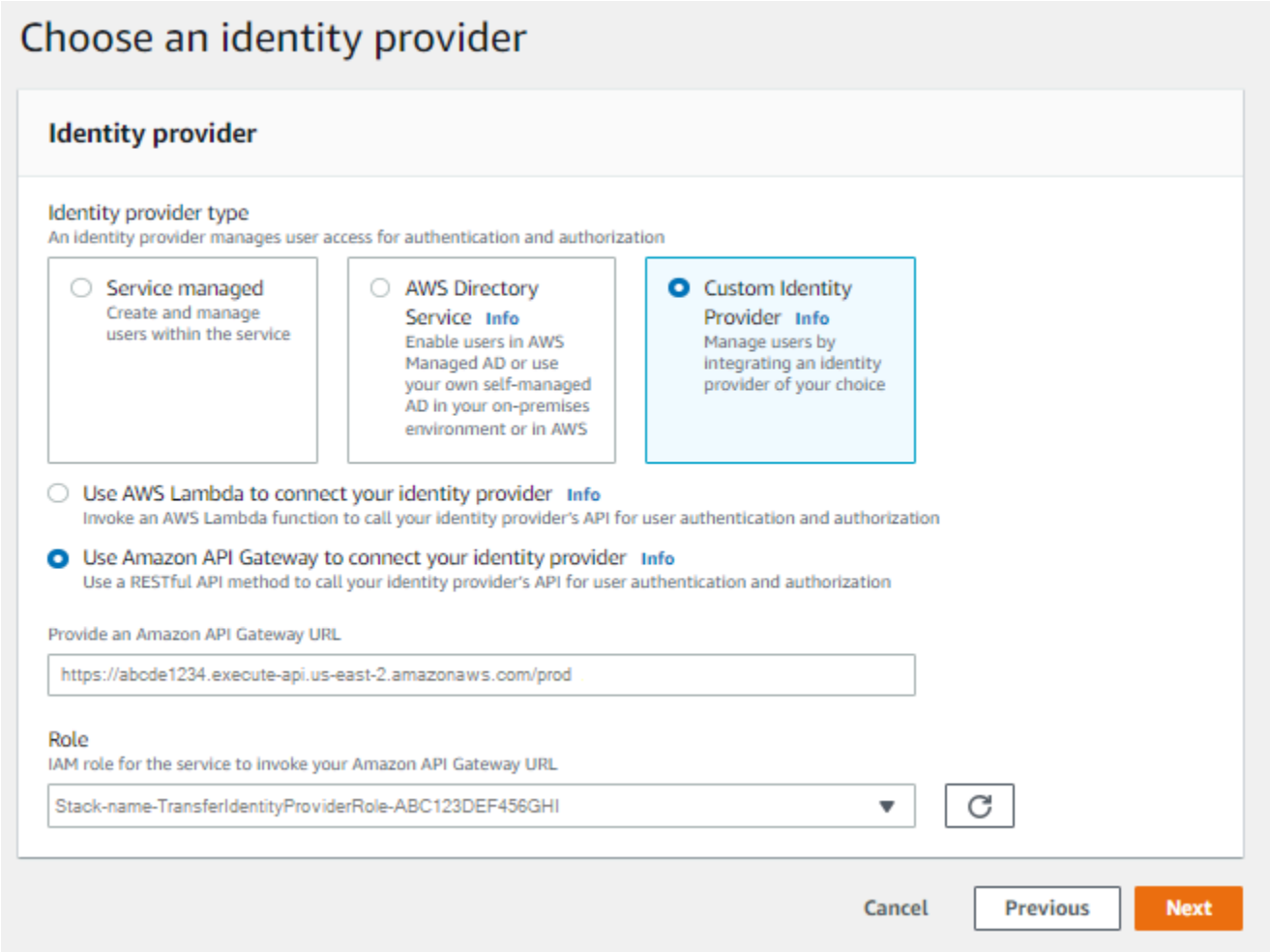
   Setelah metode API Gateway berhasil diterapkan, lihat kinerjanya di Tahapan> Detail tahap, seperti yang ditunjukkan pada gambar berikut.

> **Note**
>
> Salin alamat URL Invoke yang muncul di bagian atas layar. Anda mungkin membutuhkannya untuk langkah selanjutnya.



4. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
5. Transfer Family seharusnya dibuat untuk Anda, saat Anda membuat tumpukan. Jika tidak, konfigurasikan server Anda menggunakan langkah-langkah ini.

   a. Pilih Buat server untuk membuka halaman Buat server. Untuk Pilih penyedia identitas, pilih Kustom, lalu pilih Gunakan Amazon API Gateway untuk terhubung ke penyedia identitas Anda, seperti yang ditunjukkan pada gambar berikut.



b. Di kotak teks Berikan URL Amazon API Gateway, tempelkan alamat URL Panggilan titik akhir API Gateway yang Anda buat di langkah 3 prosedur ini.

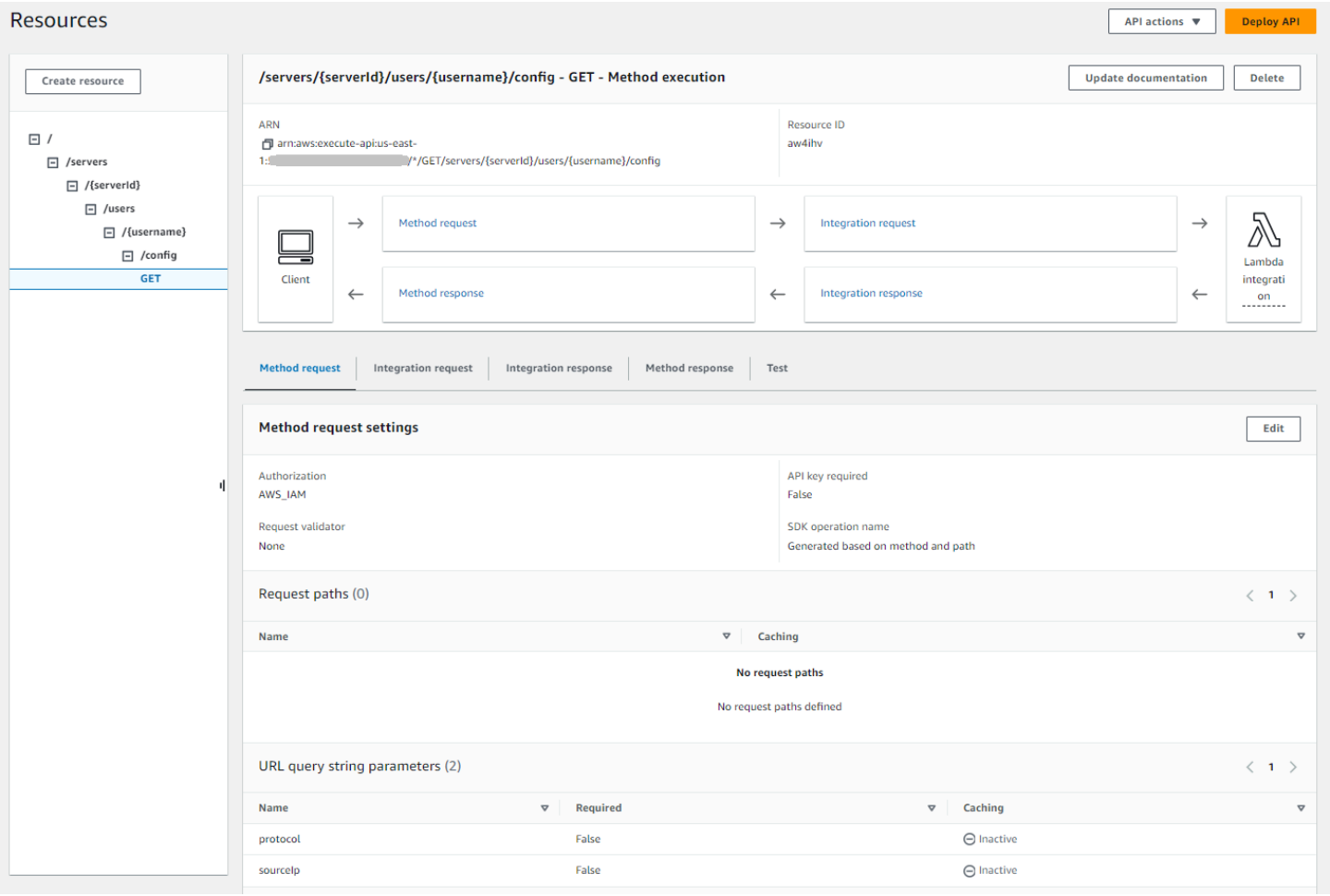
c. Untuk Peran, pilih peran IAM yang dibuat oleh AWS CloudFormation template. Peran ini memungkinkan Transfer Family untuk menjalankan metode gateway API Anda.

   Peran pemanggilan berisi nama AWS CloudFormation tumpukan yang Anda pilih untuk tumpukan yang Anda buat di langkah 1. Ini memiliki format berikut:*`CloudFormation-stack-name`*`-TransferIdentityProviderRole-`*`ABC123DEF456GHI`*.

d. Isi kotak yang tersisa, lalu pilih Buat server. Untuk detail tentang langkah-langkah yang tersisa untuk membuat server, lihatMengkonfigurasi titik akhir server SFTP, FTPS, atau FTP.

Menerapkan metode API Gateway

Untuk membuat penyedia identitas khusus untuk Transfer Family, metode API Gateway Anda harus mengimplementasikan satu metode yang memiliki jalur sumber daya`/servers/`*`serverId`*`/users/`*`username`*`/config`. *`username`*Nilai *`serverId`* dan berasal dari jalur sumber daya RESTful. Juga, tambahkan `sourceIp` dan `protocol` sebagai Parameter String Kueri URL dalam Permintaan Metode, seperti yang ditunjukkan pada gambar berikut.



> **Note**
>
> Nama pengguna harus minimal 3 dan maksimal 100 karakter. Anda dapat menggunakan karakter berikut dalam nama pengguna: a—z, A-Z, 0—9, garis bawah (_), tanda hubung (-), titik (.), dan di tanda (@). Namun, nama pengguna tidak dapat memulai dengan tanda hubung (-), titik (.), atau di tanda (@).

Jika Transfer Family mencoba otentikasi kata sandi untuk pengguna Anda, layanan akan menyediakan bidang `Password:` header. Jika tidak ada `Password:` header, Transfer Family mencoba otentikasi kunci publik untuk mengautentikasi pengguna Anda.

Saat Anda menggunakan penyedia identitas untuk mengautentikasi dan mengotorisasi pengguna akhir, selain memvalidasi kredensialnya, Anda dapat mengizinkan atau menolak permintaan akses berdasarkan alamat IP klien yang digunakan oleh pengguna akhir Anda. Anda dapat menggunakan fitur ini untuk memastikan bahwa data yang disimpan di bucket S3 atau sistem file Amazon EFS Anda dapat diakses melalui protokol yang didukung hanya dari alamat IP yang telah Anda tentukan sebagai tepercaya. Untuk mengaktifkan fitur ini, Anda harus menyertakan `sourceIp` dalam string Query.

Jika Anda memiliki beberapa protokol yang diaktifkan untuk server Anda dan ingin memberikan akses menggunakan nama pengguna yang sama melalui beberapa protokol, Anda dapat melakukannya selama kredensyal khusus untuk setiap protokol telah diatur di penyedia identitas Anda. Untuk mengaktifkan fitur ini, Anda harus menyertakan *`protocol`* nilai di jalur sumber daya RESTful.

Metode API Gateway Anda harus selalu menampilkan kode status HTTP`200`. Kode status HTTP lainnya berarti ada kesalahan saat mengakses API.

Contoh respons Amazon S3

Contoh badan respons adalah dokumen JSON dari formulir berikut untuk Amazon S3.

```
{
 "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

Kebijakan ini lolos dari JSON sebagai string. Sebagai contoh:

```
"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\"}]
}"
```

Contoh respon berikut menunjukkan bahwa pengguna memiliki tipe direktori home logis.

```
{
   "Role": "arn:aws:iam::123456789012:role/transfer-access-role-s3",
   "HomeDirectoryType":"LOGICAL",
   "HomeDirectoryDetails":"[{\"Entry\":\"/\",\"Target\":\"/MY-HOME-BUCKET\"}]",
   "PublicKeys":[""]
}
```

Contoh respons Amazon EFS

Contoh badan respons adalah dokumen JSON dari formulir berikut untuk Amazon EFS.

```
{
 "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`Bidang menunjukkan bahwa otentikasi berhasil terjadi. Saat melakukan otentikasi kata sandi (saat Anda menyediakan `Password:` header), Anda tidak perlu memberikan kunci publik SSH. Jika pengguna tidak dapat diautentikasi, misalnya, jika kata sandi salah, metode Anda harus mengembalikan respons tanpa `Role` disetel. Contoh dari respon tersebut adalah objek JSON kosong.

Contoh respon berikut menunjukkan pengguna yang memiliki tipe direktori home logis.

```
{
    "Role": "arn:aws:iam::123456789012:role/transfer-access-role-efs",
    "HomeDirectoryType": "LOGICAL",
    "HomeDirectoryDetails":"[{\"Entry\":\"/\",\"Target\":\"/faa1a123\"}]",
    "PublicKeys":[""],
    "PosixProfile":{"Uid":65534,"Gid":65534}
}
```

Anda dapat menyertakan kebijakan pengguna dalam fungsi Lambda dalam format JSON. Untuk informasi selengkapnya tentang mengonfigurasi kebijakan pengguna di Transfer Family, lihatMengelola kontrol akses.

Fungsi Lambda default

Untuk menerapkan strategi otentikasi yang berbeda, edit fungsi Lambda yang digunakan gateway Anda. Untuk membantu Anda memenuhi kebutuhan aplikasi Anda, Anda dapat menggunakan contoh fungsi Lambda berikut di Node.js. Untuk informasi selengkapnya tentang Lambda, lihat Panduan AWS Lambda Pengembang atau Membangun fungsi Lambda dengan Node.js.

Contoh fungsi Lambda berikut mengambil nama pengguna, kata sandi (jika Anda melakukan otentikasi kata sandi), ID server, protokol, dan alamat IP klien. Anda dapat menggunakan kombinasi input ini untuk mencari penyedia identitas Anda dan menentukan apakah login harus diterima.

> **Note**
>
> Jika Anda memiliki beberapa protokol yang diaktifkan untuk server Anda dan ingin memberikan akses menggunakan nama pengguna yang sama melalui beberapa protokol, Anda dapat melakukannya selama kredensyal khusus untuk protokol telah diatur di penyedia identitas Anda.
> Untuk File Transfer Protocol (FTP), kami sarankan untuk mempertahankan kredensyal terpisah dari Secure Shell (SSH) File Transfer Protocol (SFTP) dan File Transfer Protocol melalui SSL (FTPS). Sebaiknya pertahankan kredensyal terpisah untuk FTP karena, tidak seperti SFTP dan FTPS, FTP mentransmisikan kredensyal dalam teks yang jelas. Dengan mengisolasi kredensyal FTP dari SFTP atau FTPS, jika kredensyal FTP dibagikan atau diekspos, beban kerja Anda menggunakan SFTP atau FTPS tetap aman.

Fungsi contoh ini mengembalikan peran dan rincian direktori home logis, bersama dengan kunci publik (jika melakukan otentikasi kunci publik).

Saat Anda membuat pengguna yang dikelola layanan, Anda mengatur direktori home mereka, baik logis maupun fisik. Demikian pula, kita membutuhkan hasil fungsi Lambda untuk menyampaikan struktur direktori fisik atau logis pengguna yang diinginkan. Parameter yang Anda tetapkan bergantung pada nilai untuk HomeDirectoryTypebidang tersebut.

- `HomeDirectoryType`disetel ke `PATH` — `HomeDirectory` bidang tersebut kemudian harus berupa awalan bucket Amazon S3 absolut atau jalur absolut Amazon EFS yang dapat dilihat oleh pengguna Anda.
- `HomeDirectoryType`set ke `LOGICAL` - Jangan mengatur `HomeDirectory` bidang. Sebagai gantinya, kami menetapkan `HomeDirectoryDetails` bidang yang menyediakan pemetaan Entry/Target yang diinginkan, mirip dengan nilai yang dijelaskan dalam HomeDirectoryDetailsparameter untuk pengguna yang dikelola layanan.

Contoh fungsi tercantum dalamContoh fungsi Lambda.

Fungsi Lambda untuk digunakan dengan AWS Secrets Manager

Untuk digunakan AWS Secrets Manager sebagai penyedia identitas Anda, Anda dapat bekerja dengan fungsi Lambda di template sampel AWS CloudFormation . Fungsi Lambda menanyakan layanan Secrets Manager dengan kredensyal Anda dan, jika berhasil, mengembalikan rahasia yang

ditentukan. Untuk informasi selengkapnya tentang Secrets Manager, lihat Panduan Pengguna AWS Secrets Manager.

Untuk mengunduh contoh AWS CloudFormation template yang menggunakan fungsi Lambda ini, buka bucket Amazon S3 yang disediakan oleh. AWS Transfer Family

Perbaikan AWS CloudFormation template

Perbaikan antarmuka API Gateway telah dilakukan pada CloudFormation template yang diterbitkan. Template sekarang menggunakan kata sandi yang dienkode Base64 dengan API Gateway. Penerapan Anda yang ada terus berfungsi tanpa peningkatan ini, tetapi jangan izinkan kata sandi dengan karakter di luar set karakter AS-ASCII dasar.

Perubahan dalam template yang mengaktifkan kemampuan ini adalah sebagai berikut:

- `GetUserConfigRequest AWS::ApiGateway::Method`Sumber daya harus memiliki `RequestTemplates` kode ini (baris miring adalah baris yang diperbarui)

```
RequestTemplates:
   application/json: |
   {
      "username": "$util.urlDecode($input.params('username'))",
      "password":
 "$util.escapeJavaScript($util.base64Decode($input.params('PasswordBase64'))).replaceAll("\
\'","'")",
      "protocol": "$input.params('protocol')",
      "serverId": "$input.params('serverId')",
      "sourceIp": "$input.params('sourceIp')"
}
```

- `GetUserConfig`Sumber daya harus diubah untuk menggunakan `PasswordBase64` header (baris miring adalah baris yang diperbarui): `RequestParameters`

```
RequestParameters:
   method.request.header.PasswordBase64: false
   method.request.querystring.protocol: false
   method.request.querystring.sourceIp: false
```

Untuk memeriksa apakah template untuk tumpukan Anda adalah yang terbaru

1. Buka AWS CloudFormation konsol di https://console.aws.amazon.com/cloudformation.

2. Dari daftar tumpukan, pilih tumpukan Anda.
3. Dari panel detail, pilih tab Template.
4. Cari yang berikut ini:

   - Cari`RequestTemplates`, dan pastikan Anda memiliki baris ini:

     ```
     "password":
      "$util.escapeJavaScript($util.base64Decode($input.params('PasswordBase64'))).replaceAll(
     \'","'")",
     ```

   - Cari`RequestParameters`, dan pastikan Anda memiliki baris ini:

     ```
     method.request.header.PasswordBase64: false
     ```

Jika Anda tidak melihat baris yang diperbarui, edit tumpukan Anda. Untuk detail tentang cara memperbarui AWS CloudFormation tumpukan Anda, lihat Memodifikasi template tumpukan di AWS CloudFormation; Panduan Pengguna.

Menggunakan direktori logis untuk menyederhanakan struktur direktori Transfer Family Anda

Untuk menyederhanakan struktur direktori AWS Transfer Family server Anda, Anda dapat menggunakan direktori logis. Dengan direktori logis, Anda dapat membuat struktur direktori virtual yang menggunakan nama yang mudah digunakan yang dinavigasi pengguna saat mereka terhubung ke bucket Amazon S3 atau sistem file Amazon EFS Anda. Saat menggunakan direktori logis, Anda dapat menghindari pengungkapan jalur direktori absolut, nama bucket Amazon S3, dan nama sistem file EFS kepada pengguna akhir Anda.

> **Note**
>
> Anda harus menggunakan kebijakan sesi sehingga pengguna akhir Anda hanya dapat melakukan operasi yang Anda izinkan untuk mereka lakukan.
> Anda harus menggunakan direktori logis untuk membuat direktori virtual yang ramah pengguna untuk pengguna akhir Anda dan mengabstraksi nama bucket jauh. Pemetaan direktori logis hanya memungkinkan pengguna untuk mengakses jalur logis dan subdirektori yang ditunjuk, dan melarang jalur relatif yang melintasi akar logis.

Transfer Family memvalidasi setiap jalur yang mungkin menyertakan elemen relatif dan secara aktif memblokir jalur ini agar tidak diselesaikan sebelum kami meneruskan jalur ini ke Amazon S3; ini mencegah pengguna Anda bergerak melampaui pemetaan logisnya. Meskipun Transfer Family mencegah pengguna akhir mengakses direktori di luar direktori logisnya, kami sarankan Anda juga menggunakan peran unik atau kebijakan sesi untuk menerapkan hak istimewa paling sedikit di tingkat penyimpanan.

Anda dapat menggunakan direktori logis untuk mengatur direktori root pengguna ke lokasi yang diinginkan dalam hierarki penyimpanan Anda, dengan melakukan apa yang dikenal sebagai chroot operasi. Dalam mode ini, pengguna tidak dapat menavigasi ke direktori di luar direktori home atau root yang telah Anda konfigurasi untuk mereka.

Misalnya, meskipun pengguna Amazon S3 telah dicakup untuk mengakses saja`/mybucket/home/${transfer:UserName}`, beberapa klien mengizinkan pengguna untuk melintasi folder ke. `/mybucket/home` Dalam situasi ini, pengguna kembali ke direktori home yang dimaksudkan hanya setelah keluar dari dan kembali ke server Transfer Family lagi. Melakukan chroot operasi dapat mencegah situasi ini terjadi.

Anda dapat membuat struktur direktori Anda sendiri di seluruh bucket dan awalan. Fitur ini berguna jika Anda memiliki alur kerja yang mengharapkan struktur direktori tertentu yang tidak dapat Anda tiru melalui awalan bucket. Anda juga dapat menautkan ke beberapa lokasi yang tidak berdekatan dalam Amazon S3, mirip dengan membuat tautan simbolis dalam sistem file Linux di mana jalur direktori Anda mereferensikan lokasi yang berbeda dalam sistem file.

Pemetaan FILE direktori logis

Tipe `HomeDirectoryMapEntry` data sekarang menyertakan `Type` parameter. Sebelum parameter ini ada, Anda bisa membuat pemetaan direktori logis di mana targetnya adalah file. Jika sebelumnya Anda telah membuat salah satu dari jenis pemetaan direktori logis ini, Anda harus secara eksplisit menyetelnya `Type` ke`FILE`, atau pemetaan ini tidak akan berfungsi dengan benar di masa mendatang.
Salah satu cara untuk melakukannya adalah dengan memanggil `UpdateUser` API, dan mengatur `Type` ke `FILE` untuk pemetaan yang ada.

Aturan untuk menggunakan direktori logis

Sebelum Anda membangun pemetaan direktori logis Anda, Anda harus memahami aturan berikut:

- `Entry`Kapan`"/"`, Anda hanya dapat memiliki satu pemetaan karena jalur yang tumpang tindih tidak diperbolehkan.
- Direktori logis mendukung pemetaan hingga 2,1 MB (untuk pengguna yang dikelola layanan, batas ini adalah 2.000 entri). Artinya, struktur data yang berisi pemetaan memiliki ukuran maksimum 2,1 MB. Jika Anda memiliki banyak pemetaan, Anda dapat menghitung ukuran pemetaan Anda sebagai berikut:

  1. Tuliskan pemetaan khas dalam format`{"Entry":"/`*`entry-path`*`","Target":"/`*`target-path`*`"}`, di mana *`entry-path`* dan *`target-path`* merupakan nilai aktual yang akan Anda gunakan.
  2. Hitung karakter dalam string itu, lalu tambahkan satu (1).
  3. Kalikan angka itu dengan perkiraan jumlah pemetaan yang Anda miliki untuk server Anda.

  Jika jumlah yang Anda perkirakan pada langkah 3 kurang dari 2,1 MB, maka pemetaan Anda berada dalam batas yang dapat diterima.
- Target dapat menggunakan `${transfer:UserName}` variabel jika bucket atau jalur sistem file telah diparameterisasi berdasarkan nama pengguna.
- Target dapat berupa jalur dalam bucket atau sistem file yang berbeda, tetapi Anda harus memastikan bahwa peran yang dipetakan AWS Identity and Access Management (IAM) (`Role`parameter dalam respons) menyediakan akses ke bucket atau sistem file tersebut.
- Jangan tentukan `HomeDirectory` parameternya, karena nilai ini tersirat oleh `Entry` `Target` pasangan saat Anda menggunakan `LOGICAL` nilai untuk `HomeDirectoryType` parameter tersebut.
- Target harus dimulai dengan karakter garis miring (`/`), tetapi jangan gunakan garis miring ke depan (`/`) saat Anda menentukan. `Target` Misalnya, dapat `/`*`DOC-EXAMPLE-BUCKET`*`/images` diterima, tetapi *`DOC-EXAMPLE-BUCKET`*`/images` dan `/`*`DOC-EXAMPLE-BUCKET`*`/images/` tidak.
- Amazon S3 adalah toko objek, yang berarti folder adalah konsep virtual, dan tidak ada hierarki direktori yang sebenarnya. Jika aplikasi Anda mengeluarkan `stat` operasi dari klien, semuanya diklasifikasikan sebagai file saat Anda menggunakan Amazon S3 untuk penyimpanan. Perilaku ini dijelaskan dalam Mengatur objek di konsol Amazon S3 menggunakan folder di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon. Jika aplikasi Anda mengharuskan untuk menunjukkan

`stat` secara akurat apakah sesuatu adalah file atau folder, Anda dapat menggunakan Amazon Elastic File System (Amazon EFS) sebagai opsi penyimpanan untuk server Transfer Family Anda.

- Jika Anda menentukan nilai direktori logis untuk pengguna Anda, parameter yang Anda gunakan bergantung pada jenis pengguna:
  - Untuk pengguna yang dikelola layanan, berikan nilai direktori logis di. `HomeDirectoryMappings`
  - Untuk pengguna penyedia identitas kustom, berikan nilai direktori logis di`HomeDirectoryDetails`.

> **Important**
>
> Kecuali Anda memilih untuk mengoptimalkan kinerja untuk direktori Amazon S3 Anda (saat Anda membuat atau memperbarui server), direktori root harus ada saat startup. Untuk Amazon S3, ini berarti Anda harus sudah membuat objek zero-byte yang diakhiri dengan garis miring maju (/) untuk membuat folder root. Menghindari masalah ini adalah alasan untuk mempertimbangkan mengoptimalkan kinerja Amazon S3.

Menerapkan direktori logis dan **`chroot`**

Untuk menggunakan direktori dan chroot fitur logis, Anda harus melakukan hal berikut:

Aktifkan direktori logis untuk setiap pengguna. Lakukan ini dengan mengatur `HomeDirectoryType` parameter `LOGICAL` saat Anda membuat atau memperbarui pengguna Anda.

```
"HomeDirectoryType": "LOGICAL"
```

`chroot`

Untukchroot, buat struktur direktori yang terdiri dari satu `Entry` dan `Target` pasangan untuk setiap pengguna. Folder root adalah `Entry` intinya, dan lokasi di bucket atau sistem file Anda untuk dipetakan. `Target`

Example for Amazon S3

```
[{"Entry": "/", "Target": "/mybucket/jane"}]
```

Example for Amazon EFS

```
[{"Entry": "/", "Target": "/fs-faa1a123/jane"}]
```

Anda dapat menggunakan jalur absolut seperti pada contoh sebelumnya, atau Anda dapat menggunakan substitusi dinamis untuk nama pengguna dengan`${transfer:UserName}`, seperti pada contoh berikut.

```
[{"Entry": "/", "Target":
"/mybucket/${transfer:UserName}"}]
```

Dalam contoh sebelumnya, pengguna dikunci ke direktori root mereka dan tidak dapat melintasi lebih tinggi dalam hierarki.

Struktur direktori virtual

Untuk struktur direktori virtual, Anda dapat membuat beberapa `Entry` `Target` pasangan, dengan target di mana saja di bucket S3 atau sistem file EFS Anda, termasuk di beberapa bucket atau sistem file, selama pemetaan peran IAM pengguna memiliki izin untuk mengaksesnya.

Dalam contoh struktur virtual berikut, ketika pengguna login ke AWS SFTP, mereka berada di direktori root dengan sub-direktori`/pics`,,, `/doc` dan. `/reporting` `/anotherpath/subpath/financials`

> **Note**
>
> Kecuali Anda memilih untuk mengoptimalkan kinerja untuk direktori Amazon S3 Anda (saat Anda membuat atau memperbarui server), baik pengguna atau administrator perlu membuat direktori jika belum ada. Menghindari masalah ini adalah alasan untuk mempertimbangkan mengoptimalkan kinerja Amazon S3.
> Untuk Amazon EFS, Anda masih memerlukan administrator untuk membuat pemetaan logis atau direktori. `/`

```
[
{"Entry": "/pics", "Target": "/bucket1/pics"},
{"Entry": "/doc", "Target": "/bucket1/anotherpath/docs"},
```

```
{"Entry": "/reporting", "Target": "/reportingbucket/Q1"},
{"Entry": "/anotherpath/subpath/financials", "Target": "/reportingbucket/financials"}]
```

Note

Anda hanya dapat mengunggah file ke folder tertentu yang Anda petakan. Ini berarti bahwa dalam contoh sebelumnya, Anda tidak dapat mengunggah ke `/anotherpath` atau `anotherpath/subpath` direktori; hanya`anotherpath/subpath/financials`. Anda juga tidak dapat memetakan ke jalur tersebut secara langsung, karena jalur yang tumpang tindih tidak diperbolehkan.
Misalnya, asumsikan Anda membuat pemetaan berikut:

```
{
   "Entry": "/pics",
   "Target": "/mybucket/pics"
},
{
   "Entry": "/doc",
   "Target": "/mybucket/mydocs"
},
{
   "Entry": "/temp",
   "Target": "/mybucket"
}
```

Anda hanya dapat mengunggah file ke bucket tersebut. Ketika Anda pertama kali terhubung`sftp`, Anda dijatuhkan ke direktori root,`/`. Jika Anda mencoba mengunggah file ke direktori itu, unggahan gagal. Perintah berikut menunjukkan urutan contoh:

```
sftp> pwd
Remote working directory: /
sftp> put file
Uploading file to /file
remote open("/file"): No such file or directory
```

Untuk mengunggah ke salah satu`directory/sub-directory`, Anda harus secara eksplisit memetakan jalur ke. `sub-directory`

Untuk informasi selengkapnya tentang mengonfigurasi direktori logis dan chroot untuk pengguna Anda, termasuk AWS CloudFormation templat yang dapat Anda unduh dan gunakan, lihat Menyederhanakan Struktur AWS SFTP Anda dengan direktori chroot dan logis di Blog Penyimpanan. AWS

Konfigurasikan contoh direktori logis

Dalam contoh ini, kami membuat pengguna dan menetapkan dua direktori logis. Perintah berikut membuat pengguna baru (untuk server Transfer Family yang sudah ada) dengan direktori logis `pics` dan`doc`.

```
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
```

Jika **`marymajor`** adalah pengguna yang sudah ada dan jenis direktori home nyaPATH, Anda dapat mengubahnya `LOGICAL` dengan perintah yang sama seperti yang sebelumnya.

```
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\"}]"
```

Perhatikan hal berikut:

- Jika direktori */DOC-EXAMPLE-BUCKET1*`/pics` dan */DOC-EXAMPLE-BUCKET2*`/test/mydocs` belum ada, pengguna (atau administrator) perlu membuatnya.
- Ketika **`marymajor`** terhubung ke server, dan menjalankan `ls -l` perintah, dia melihat yang berikut:

  ```
  drwxr--r--   1        -        -        0 Mar 17 15:42 doc
  drwxr--r--   1        -        -        0 Mar 17 16:04 pics
  ```

- **`marymajor`**tidak dapat membuat file atau direktori apa pun pada level ini. Namun, di dalam `pics` dan`doc`, dia dapat menambahkan sub-direktori.

- File yang dia tambahkan `pics` dan `doc` ditambahkan ke jalur Amazon S3 */DOC-EXAMPLE-BUCKET1*`/pics` dan */DOC-EXAMPLE-BUCKET2*`/test/mydocs` masing-masing.
- Dalam contoh ini, kami menentukan dua ember berbeda untuk menggambarkan kemungkinan itu. Namun, Anda dapat menggunakan bucket yang sama untuk beberapa atau semua direktori logis yang Anda tentukan untuk pengguna.

Konfigurasikan direktori logis untuk Amazon EFS

Jika server Transfer Family Anda menggunakan Amazon EFS, direktori home untuk pengguna harus dibuat dengan akses baca dan tulis sebelum pengguna dapat bekerja di direktori home logisnya. Pengguna tidak dapat membuat direktori ini sendiri, karena mereka akan kekurangan izin untuk `mkdir` direktori home logis mereka.

Jika direktori home pengguna tidak ada, dan mereka menjalankan `ls` perintah, sistem merespons sebagai berikut:

```
sftp> ls
remote readdir ("/"): No such file or directory
```

Seorang pengguna dengan akses administratif ke direktori induk perlu membuat direktori home logis pengguna.

AWS Lambda Tanggapan khusus

Anda dapat menggunakan direktori logis dengan fungsi Lambda yang terhubung ke penyedia identitas kustom Anda. Untuk melakukannya, dalam fungsi Lambda Anda, Anda menentukan `HomeDirectoryType` sebagai**`LOGICAL`**, dan menambahkan `Entry` dan `Target` nilai untuk parameter. `HomeDirectoryDetails` Sebagai contoh:

```
HomeDirectoryType: "LOGICAL"
HomeDirectoryDetails: "[{\"Entry\": \"/\", \"Target\": \"/DOC-EXAMPLE-BUCKET/
theRealFolder"}]"
```

Kode berikut adalah contoh respons yang berhasil dari panggilan otentikasi Lambda kustom.

```
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**
>
> `"Url":` Baris dikembalikan hanya jika Anda menggunakan metode API Gateway sebagai penyedia identitas kustom Anda.

AWS Transfer Family Konektor SFTP

AWS Transfer Family Konektor SFTP membangun hubungan untuk mengirim file dan pesan antara penyimpanan Amazon dan mitra eksternal, menggunakan protokol SFTP. Anda dapat mengirim file dari Amazon S3 ke tujuan eksternal milik mitra. Anda juga dapat menggunakan konektor SFTP untuk mengambil file dari server SFTP mitra.

> **Note**
>
> Saat ini, konektor SFTP hanya dapat digunakan untuk terhubung ke server SFTP jarak jauh yang menawarkan titik akhir yang dapat diakses internet.

Lihat konektor AWS Transfer Family SFTP untuk pengenalan singkat tentang konektor SFTP Transfer Family.

Topik

- Konfigurasikan konektor SFTP
- Mengirim dan mengambil file dengan menggunakan konektor SFTP
- Kelola konektor SFTP

Konfigurasikan konektor SFTP

Topik ini menjelaskan cara membuat konektor SFTP, algoritma keamanan yang terkait dengannya, cara menyimpan rahasia untuk menyimpan kredensil, detail tentang memformat kunci pribadi, dan instruksi untuk menguji konektor Anda.

Topik

- Buat konektor SFTP
- Simpan rahasia untuk digunakan dengan konektor SFTP
- Hasilkan dan format kunci pribadi konektor SFTP
- Uji konektor SFTP

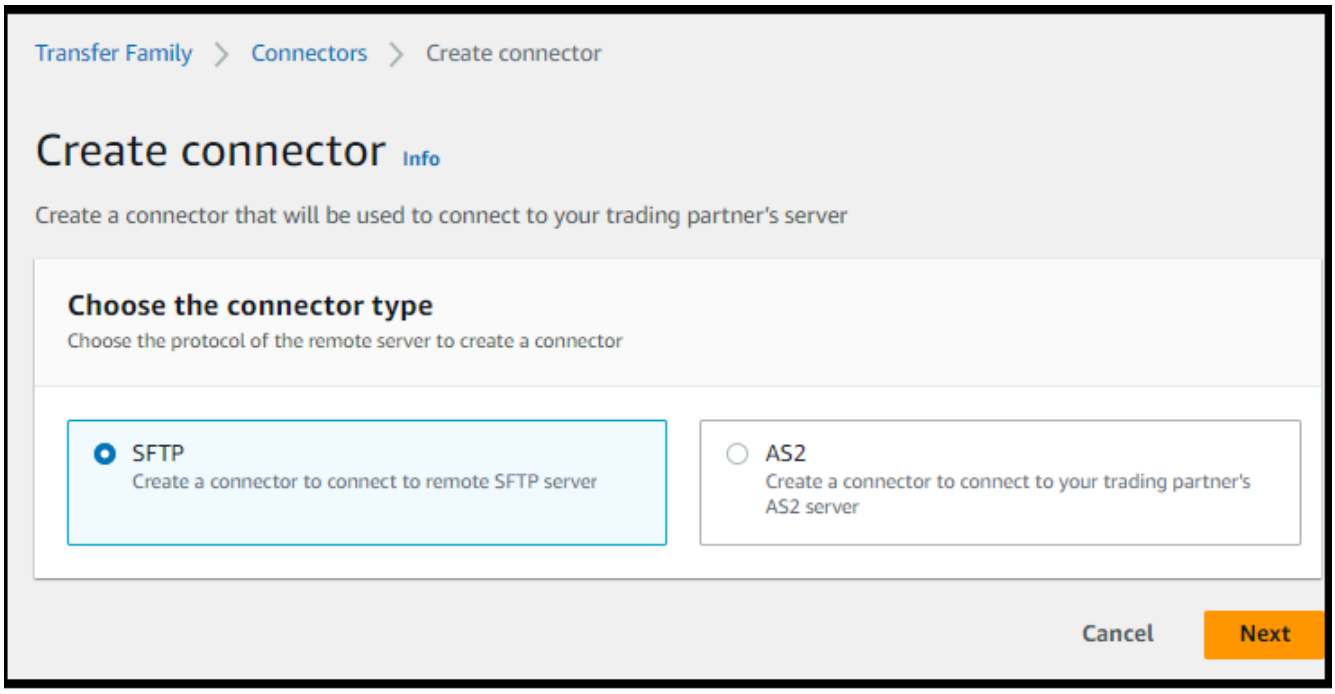
Buat konektor SFTP

Prosedur ini menjelaskan cara membuat konektor SFTP dengan menggunakan AWS Transfer Family konsol atau. AWS CLI

Console

Untuk membuat konektor SFTP

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi kiri, pilih Konektor, lalu pilih Buat konektor.
3. Pilih SFTP untuk jenis konektor untuk membuat konektor SFTP, lalu pilih Berikutnya.



4. Di bagian Konfigurasi konektor, berikan informasi berikut:

   - Untuk URL, masukkan URL untuk server SFTP jarak jauh. URL ini harus diformat sebagai`sftp://`*`partner-SFTP-server-url`*, misalnya`sftp://AnyCompany.com`.

     > **Note**
     >
     > Secara opsional, Anda dapat memberikan nomor port di URL Anda. Formatnya adalah `sftp://`*`partner-SFTP-server-url`*`:`*`port-number`*. Nomor port default (bila tidak ada port yang ditentukan) adalah port 22.

- Untuk peran Access, pilih Amazon Resource Name (ARN) dari peran AWS Identity and Access Management (IAM) yang akan digunakan.
  - Pastikan peran ini menyediakan akses baca dan tulis ke direktori induk lokasi file yang digunakan dalam `StartFileTransfer` permintaan.
  - Pastikan bahwa peran ini memberikan izin `secretsmanager:GetSecretValue` untuk mengakses rahasia.

    > **Note**
    >
    > Dalam kebijakan, Anda harus menentukan ARN untuk rahasianya. ARN berisi nama rahasia, tetapi menambahkan nama dengan enam, acak, karakter alfanumerik. ARN untuk rahasia memiliki format berikut.
    >
    > ```
    > arn:aws:secretsmanager:region:account-id:secret:aws/
    > transfer/SecretName-6RandomCharacters
    > ```

  - Pastikan peran ini berisi hubungan kepercayaan yang memungkinkan konektor mengakses sumber daya Anda saat melayani permintaan transfer pengguna Anda. Untuk detail tentang membangun hubungan kepercayaan, lihatUntuk membangun hubungan kepercayaan.

Contoh berikut memberikan izin yang diperlukan untuk mengakses *DOC-EXAMPLE-BUCKET* di Amazon S3, dan rahasia tertentu yang disimpan di Secrets Manager.

```
{
  "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

Untuk peran akses, contoh memberikan akses ke satu rahasia. Namun, Anda dapat menggunakan karakter wildcard, yang dapat menyimpan pekerjaan jika Anda ingin menggunakan kembali peran IAM yang sama untuk beberapa pengguna dan rahasia. Misalnya, pernyataan sumber daya berikut memberikan izin untuk semua rahasia yang memiliki nama yang dimulai dengan. `aws/transfer`

```
"Resource": "arn:aws:secretsmanager:region:account-id:secret:aws/
transfer/*"
```

Anda juga dapat menyimpan rahasia yang berisi kredensi SFTP Anda di tempat lain. Akun AWS Untuk detail tentang mengaktifkan akses rahasia lintas akun, lihat Izin untuk AWS Secrets Manager rahasia bagi pengguna di akun lain.

- (Opsional) Untuk peran Logging, pilih peran IAM untuk konektor yang akan digunakan untuk mendorong peristiwa ke CloudWatch log Anda. Contoh kebijakan berikut mencantumkan izin yang diperlukan untuk mencatat peristiwa untuk konektor SFTP.

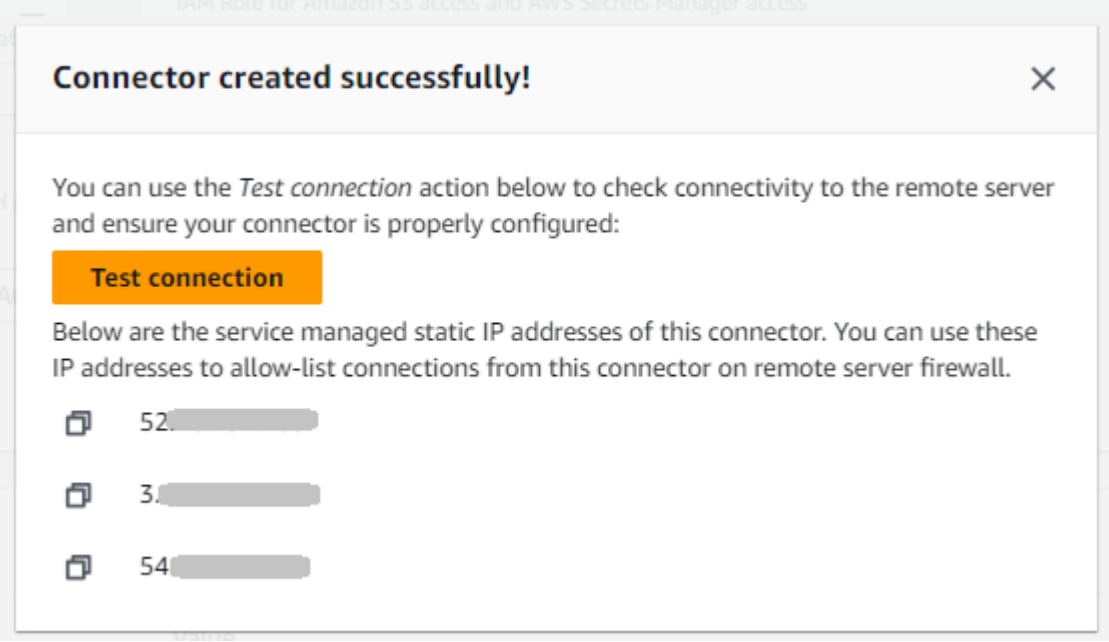
```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Sid": "SFTPConnectorPermissions",
        "Effect": "Allow",
        "Action": [
            "logs:CreateLogStream",
            "logs:DescribeLogStreams",
            "logs:CreateLogGroup",
            "logs:PutLogEvents"
        ],
        "Resource": [
            "arn:aws:logs:*:*:log-group:/aws/transfer/*"
        ]
    }]
}
```

5. Di bagian Konfigurasi SFTP, berikan informasi berikut:

   - Untuk kredensi Connector, dari daftar dropdown, pilih nama rahasia AWS Secrets Manager yang berisi kunci pribadi atau kata sandi pengguna SFTP. Anda harus membuat rahasia dan menyimpannya dengan cara tertentu. Untuk detailnya, lihat Simpan rahasia untuk digunakan dengan konektor SFTP.
   - Untuk kunci host Tepercaya, tempel di bagian publik dari kunci host yang digunakan untuk mengidentifikasi server eksternal. Anda dapat menambahkan lebih dari satu kunci, dengan memilih Tambahkan kunci host tepercaya untuk menambahkan kunci tambahan. Anda dapat menggunakan `ssh-keyscan` perintah terhadap server SFTP untuk mengambil kunci yang diperlukan. Untuk detail tentang format dan jenis kunci host tepercaya yang didukung Transfer Family, lihat SFTPConnectorConfig.

6. Di bagian Opsi algoritma kriptografi, pilih kebijakan keamanan dari daftar dropdown di bidang Kebijakan Keamanan. Kebijakan keamanan memungkinkan Anda untuk memilih algoritma kriptografi yang didukung konektor Anda. Untuk detail tentang kebijakan dan algoritme keamanan yang tersedia, lihatKebijakan keamanan untuk konektor AWS Transfer Family SFTP.

7. (Opsional) Di bagian Tag, untuk Kunci dan Nilai, masukkan satu atau beberapa tag sebagai pasangan nilai kunci.
8. Setelah Anda mengkonfirmasi semua pengaturan Anda, pilih Buat konektor untuk membuat konektor SFTP. Jika konektor berhasil dibuat, layar muncul dengan daftar alamat IP statis yang ditetapkan dan tombol koneksi Uji. Gunakan tombol untuk menguji konfigurasi konektor baru Anda.



Halaman Konektor muncul, dengan ID konektor SFTP baru Anda ditambahkan ke daftar. Untuk melihat detail konektor Anda, lihatLihat detail konektor SFTP.

CLI

Anda menggunakan create-connectorperintah untuk membuat konektor. Untuk menggunakan perintah ini untuk membuat konektor SFTP, Anda harus memberikan informasi berikut.

- URL untuk server SFTP jarak jauh. URL ini harus diformat sebagai`sftp://`*`partner-SFTP-server-url`*, misalnya`sftp://AnyCompany.com`.
- Peran akses. Pilih Nama Sumber Daya Amazon (ARN) dari peran AWS Identity and Access Management (IAM) yang akan digunakan.
  - Pastikan peran ini menyediakan akses baca dan tulis ke direktori induk lokasi file yang digunakan dalam `StartFileTransfer` permintaan.
  - Pastikan bahwa peran ini memberikan izin `secretsmanager:GetSecretValue` untuk mengakses rahasia.

> **Note**
>
> Dalam kebijakan, Anda harus menentukan ARN untuk rahasianya. ARN berisi nama rahasia, tetapi menambahkan nama dengan enam, acak, karakter alfanumerik. ARN untuk rahasia memiliki format berikut.
>
> ```
> arn:aws:secretsmanager:region:account-id:secret:aws/
> transfer/SecretName-6RandomCharacters
> ```

- Pastikan peran ini berisi hubungan kepercayaan yang memungkinkan konektor mengakses sumber daya Anda saat melayani permintaan transfer pengguna Anda. Untuk detail tentang membangun hubungan kepercayaan, lihatUntuk membangun hubungan kepercayaan.

Contoh berikut memberikan izin yang diperlukan untuk mengakses *DOC-EXAMPLE-BUCKET* di Amazon S3, dan rahasia tertentu yang disimpan di Secrets Manager.

```
{
  "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**
>
> Untuk peran akses, contoh memberikan akses ke satu rahasia. Namun, Anda dapat menggunakan karakter wildcard, yang dapat menyimpan pekerjaan jika Anda ingin menggunakan kembali peran IAM yang sama untuk beberapa pengguna dan rahasia. Misalnya, pernyataan sumber daya berikut memberikan izin untuk semua rahasia yang memiliki nama yang dimulai dengan. `aws/transfer`
>
> ```
> "Resource": "arn:aws:secretsmanager:region:account-id:secret:aws/transfer/
> *"
> ```
>
> Anda juga dapat menyimpan rahasia yang berisi kredensi SFTP Anda di tempat lain. Akun AWS Untuk detail tentang mengaktifkan akses rahasia lintas akun, lihat Izin untuk AWS Secrets Manager rahasia bagi pengguna di akun lain.

- (Opsional) Pilih peran IAM untuk konektor yang akan digunakan untuk mendorong peristiwa ke CloudWatch log Anda. Contoh kebijakan berikut mencantumkan izin yang diperlukan untuk mencatat peristiwa untuk konektor SFTP.

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

- Berikan informasi konfigurasi SFTP berikut.
  - ARN rahasia AWS Secrets Manager yang berisi kunci pribadi atau kata sandi pengguna SFTP.
  - Bagian publik dari kunci host yang digunakan untuk mengidentifikasi server eksternal. Anda dapat memberikan beberapa kunci host tepercaya jika Anda mau.

  Cara termudah untuk memberikan informasi SFTP adalah dengan menyimpannya ke file. Misalnya, salin contoh teks berikut ke file bernama`testSFTPConfig.json`.

```
// 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="
   ]
}
```

- Tentukan kebijakan keamanan untuk konektor Anda, masukkan nama kebijakan keamanan.

> **Note**
>
> `SecretId`Dapat berupa seluruh ARN atau nama rahasia (*`example-username-key`*dalam daftar sebelumnya).

Kemudian jalankan perintah berikut untuk membuat konektor.

```
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
```

Simpan rahasia untuk digunakan dengan konektor SFTP

Anda dapat menggunakan Secrets Manager untuk menyimpan kredensi pengguna untuk konektor SFTP Anda. Ketika Anda membuat rahasia Anda, Anda harus memberikan nama pengguna. Selain itu, Anda dapat memberikan kata sandi, kunci pribadi, atau keduanya. Untuk detailnya, lihat Kuota untuk konektor SFTP.

> **Note**
>
> Ketika Anda menyimpan rahasia di Secrets Manager, Anda Akun AWS dikenakan biaya. Untuk informasi tentang harga, lihat AWS Secrets Manager Harga.

Untuk menyimpan kredensi pengguna di Secrets Manager untuk konektor SFTP

1. Masuk ke AWS Management Console dan buka AWS Secrets Manager konsol di https://console.aws.amazon.com/secretsmanager/.
2. Pada panel navigasi kiri, pilih Rahasia.
3. Pada halaman Rahasia, pilih Simpan rahasia baru.
4. Pada halaman Pilih jenis rahasia, untuk tipe Rahasia, pilih Jenis rahasia lainnya.
5. Di bagian pasangan kunci/Nilai, pilih tab kunci/Nilai.
   - Kunci — Masukkan**`Username`**.
   - nilai — Masukkan nama pengguna yang berwenang untuk terhubung ke server mitra.
6. Jika Anda ingin memberikan kata sandi, pilih Tambahkan baris, dan di bagian pasangan kunci/nilai, pilih tab kunci/Nilai.

   Pilih Tambah baris, dan di bagian pasangan kunci/Nilai, pilih tab kunci/Nilai.

   - Kunci — Masukkan**`Password`**.

- nilai — Masukkan kata sandi untuk pengguna.

7. Jika Anda ingin memberikan kunci pribadi, lihatHasilkan dan format kunci pribadi konektor SFTP, yang menjelaskan cara memasukkan data kunci pribadi.

> **Note**
>
> Data kunci pribadi yang Anda masukkan harus sesuai dengan kunci publik yang disimpan untuk pengguna ini di server SFTP jarak jauh.

8. Pilih Berikutnya.
9. Pada halaman Konfigurasi rahasia, masukkan nama dan deskripsi untuk rahasia Anda. Kami menyarankan Anda menggunakan awalan **`aws/transfer/`** untuk nama tersebut. Misalnya, Anda bisa menyebutkan rahasia Anda**`aws/transfer/connector-1`**.
10. Pilih Berikutnya, dan kemudian terima default pada halaman Konfigurasi rotasi. Lalu pilih Selanjutnya.
11. Pada halaman Review, pilih Store untuk membuat dan menyimpan rahasia.

Hasilkan dan format kunci pribadi konektor SFTP

Detail lengkap untuk menghasilkan public/private key pair dijelaskan dalam. Membuat kunci SSH di macOS, Linux, atau Unix

Sebagai contoh, untuk menghasilkan kunci pribadi untuk digunakan dengan konektor SFTP, perintah contoh berikut menghasilkan jenis kunci yang benar (ganti *`key_name dengan nama`* file aktual untuk key pair Anda):

```
ssh-keygen -t rsa -b 4096 -m PEM -f key_name -N ""
```

> **Note**
>
> Saat Anda membuat key pair untuk digunakan dengan konektor SFTP, jangan gunakan frasa sandi. Frasa sandi kosong diperlukan agar konfigurasi SFTP berfungsi dengan benar.

Perintah ini menciptakan key pair RSA, dengan ukuran kunci 4096 bit. Kunci dihasilkan dalam format PEM lama, yang diperlukan oleh Transfer Family untuk digunakan dengan rahasia konektor SFTP.

Kunci disimpan di *key_name* (kunci pribadi) dan *key_name*.pub (kunci publik) di direktori saat ini: yaitu, direktori tempat Anda menjalankan `ssh-keygen` perintah.

> **Note**
>
> Transfer Family tidak mendukung format OpenSSH `-----BEGIN OPENSSH PRIVATE KEY-----` () untuk kunci yang digunakan untuk konektor SFTP Anda. Kuncinya harus dalam format PEM lama (`-----BEGIN RSA PRIVATE KEY-----`atau`-----BEGIN EC PRIVATE KEY-----`). Anda dapat menggunakan ssh-keygen alat ini untuk mengonversi kunci Anda, dengan menyediakan `-m PEM` opsi saat Anda menjalankan perintah.

Setelah Anda membuat kunci, Anda harus memastikan bahwa kunci pribadi diformat dengan karakter baris baru yang disematkan (”\n“) dalam format JSON.

Gunakan perintah untuk mengonversi kunci pribadi Anda yang ada ke format yang benar—format JSON dengan karakter baris baru yang disematkan. Di sini kami memberikan contoh untuk `jq` dan Powershell. Anda dapat menggunakan alat atau perintah apa pun yang ingin Anda ubah kunci pribadi menjadi format JSON dengan karakter baris baru yang disematkan.

jq command

Contoh ini menggunakan `jq` perintah, yang tersedia untuk diunduh dari Download jq.

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

Misalnya, jika file kunci pribadi Anda berada di`~/.ssh/my_private_key`, perintahnya adalah sebagai berikut.

```
jq -sR . ~/.ssh/my_private_key
```

Ini menghasilkan kunci dalam format yang benar (dengan karakter baris baru yang disematkan) ke output standar.

PowerShell

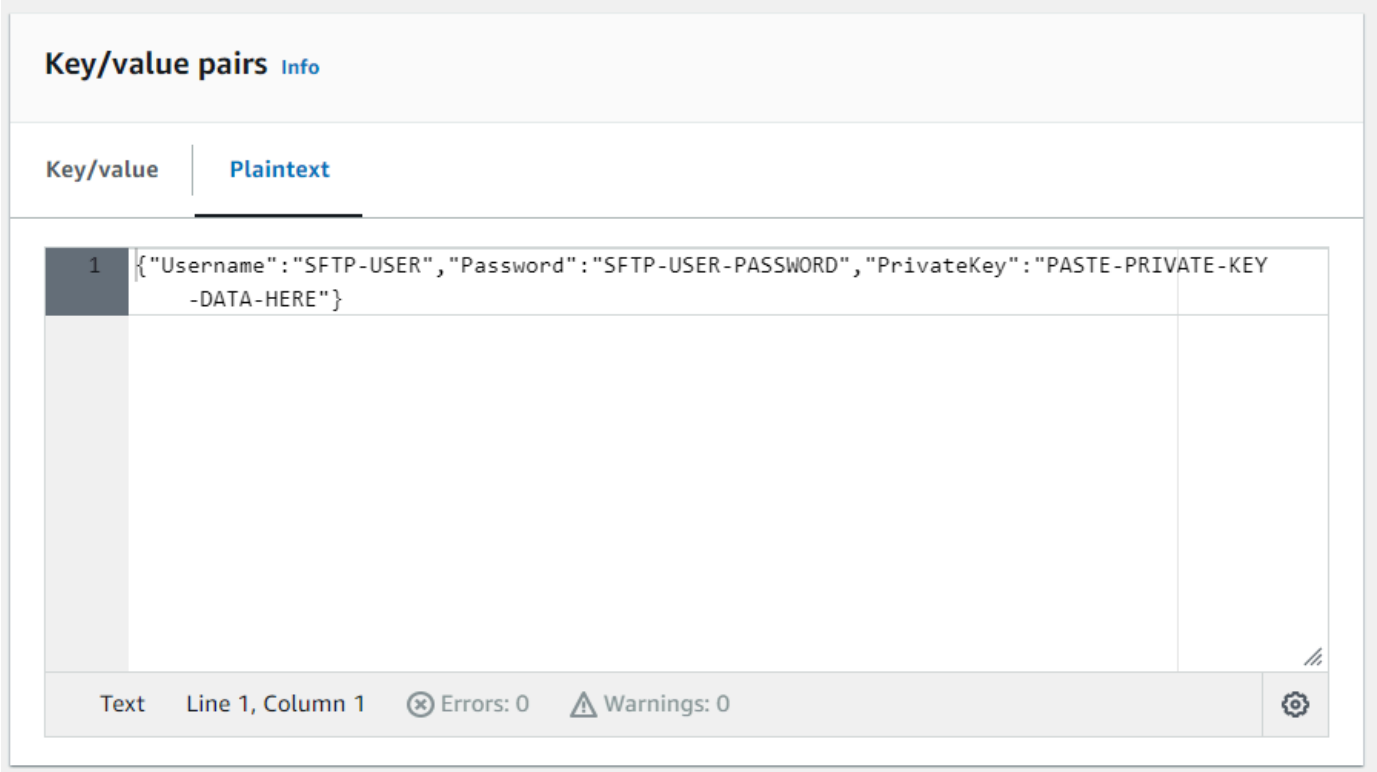
Jika Anda menggunakan Windows, Anda dapat menggunakan PowerShell untuk mengonversi kunci ke format yang benar. Perintah Powershell berikut mengonversi kunci pribadi ke format yang benar.

```
Get-Content -Raw path-to-private-key-file | ConvertTo-Json
```

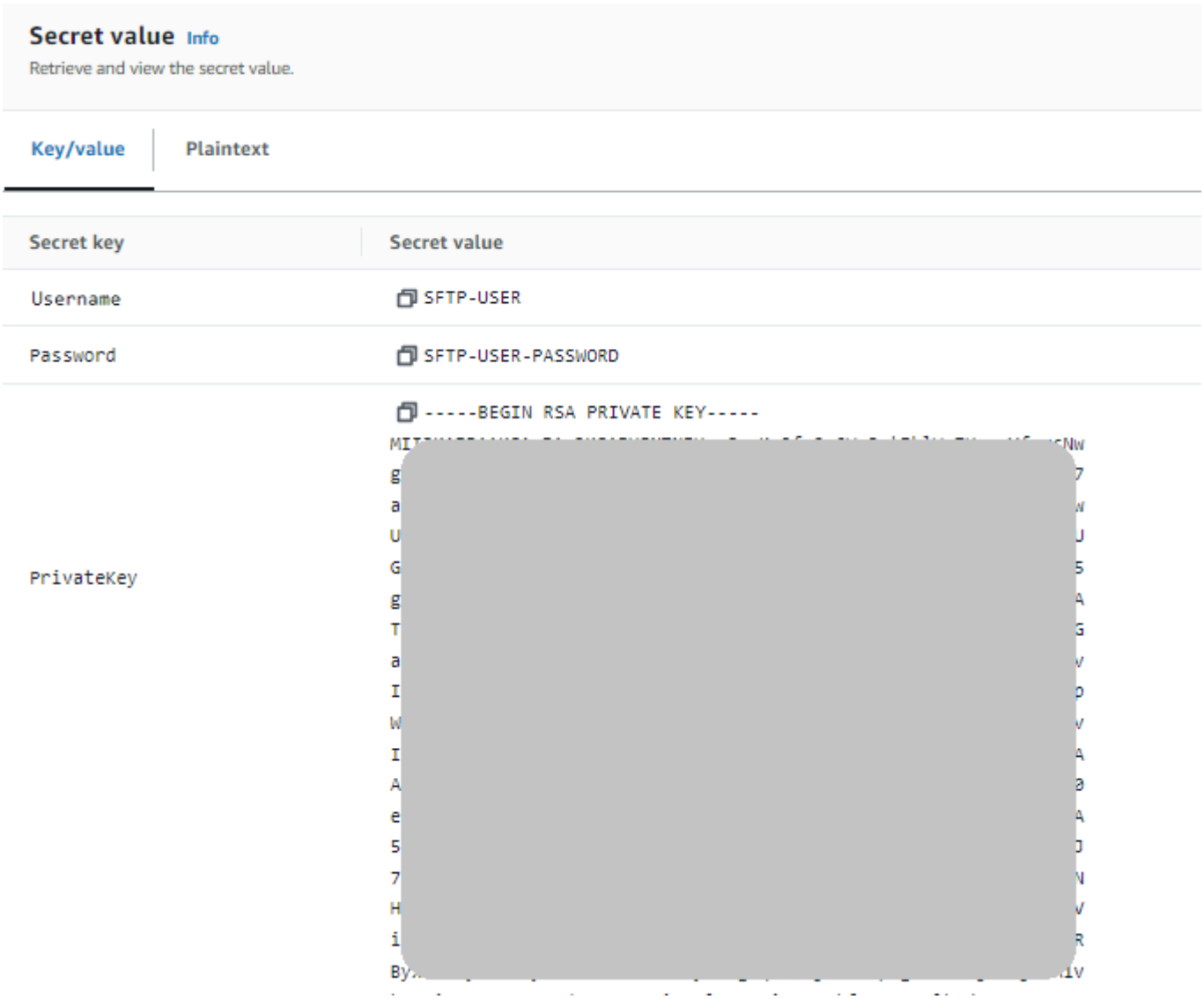
Untuk menambahkan data kunci pribadi ke rahasia untuk digunakan dengan konektor SFTP

1. Di konsol Secrets Manager, saat menyimpan jenis rahasia lainnya, pilih tab Plaintext. Teks harus kosong, dengan hanya tanda kurung pembuka dan penutup, {}.
2. Tempelkan nama pengguna, data kunci pribadi, dan/atau kata sandi Anda menggunakan format berikut. Untuk data kunci pribadi Anda, tempel output dari perintah yang Anda jalankan di langkah 1.

   ```
   {"Username":"SFTP-USER","Password":"SFTP-USER-PASSWORD","PrivateKey":"PASTE-PRIVATE-KEY-DATA-HERE"}
   ```

   Jika Anda menempelkan data kunci pribadi dengan benar, Anda akan melihat yang berikut saat memilih tab kunci/Nilai. Perhatikan bahwa data kunci pribadi ditampilkan line-by-line, bukan sebagai string teks kontinu.



3. Lanjutkan prosedur Simpan rahasia untuk digunakan dengan konektor SFTP di langkah 8, dan ikuti prosedur itu sampai akhir.

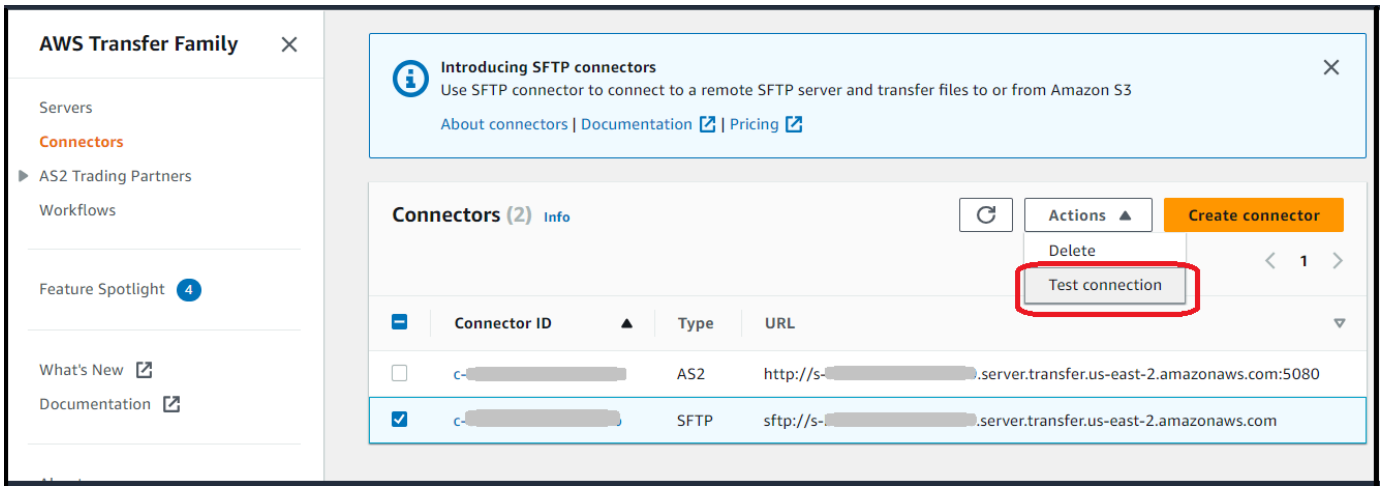
Uji konektor SFTP

Setelah Anda membuat konektor SFTP, kami sarankan Anda mengujinya sebelum mencoba mentransfer file apa pun menggunakan konektor baru Anda.

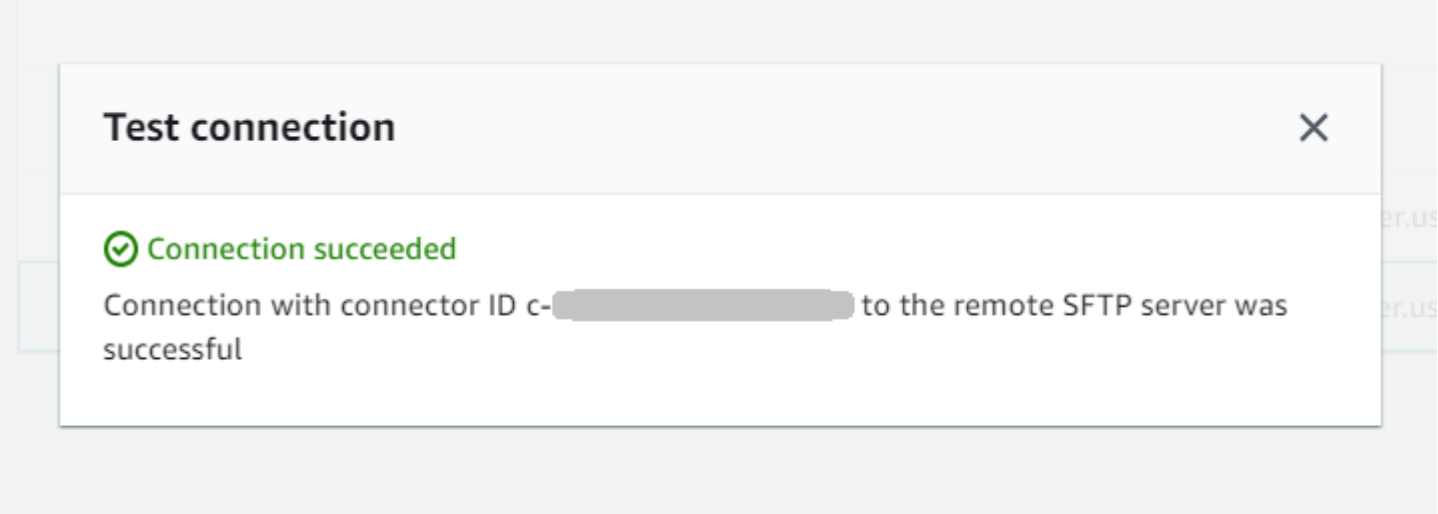
Untuk menguji konektor SFTP

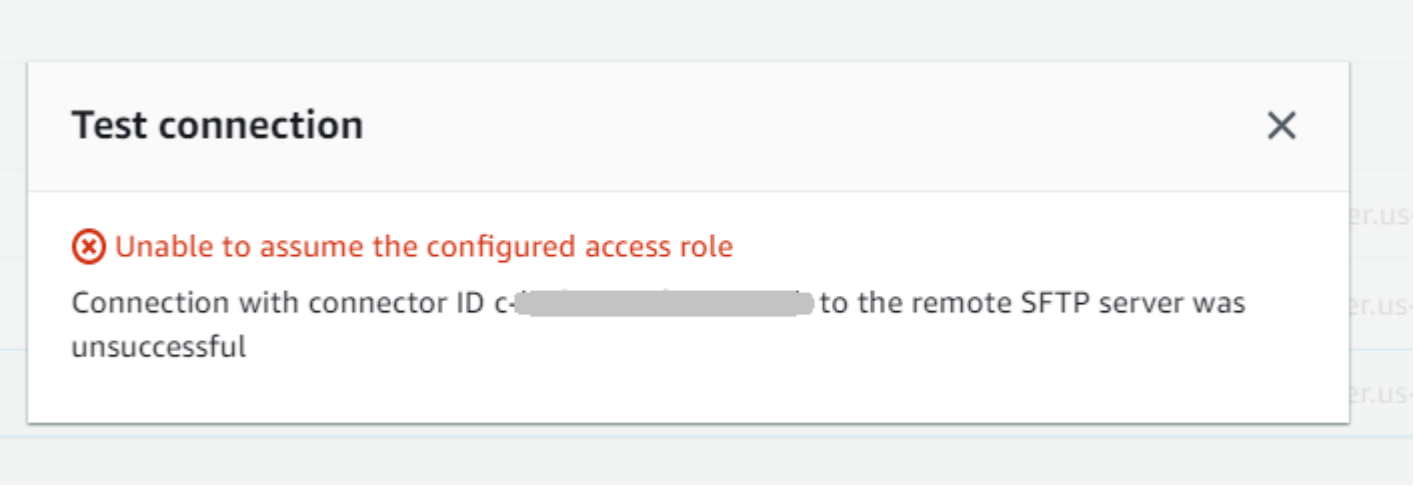
1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi kiri, pilih Konektor, dan pilih konektor.

3. Dari menu Tindakan, pilih Uji koneksi.



Sistem mengembalikan pesan, menunjukkan apakah tes lulus atau gagal. Jika tes gagal, sistem memberikan pesan kesalahan berdasarkan alasan pengujian gagal.





> **Note**
>
> Untuk menggunakan API untuk menguji konektor Anda, lihat dokumentasi TestConnectionAPI.

Mengirim dan mengambil file dengan menggunakan konektor SFTP

Konektor SFTP memperluas kemampuan AWS Transfer Family untuk berkomunikasi dengan server jarak jauh baik di cloud maupun di tempat. Anda dapat mengintegrasikan data yang dihasilkan dan disimpan dalam sumber jarak jauh dengan gudang data yang AWS dihosting untuk analitik, aplikasi bisnis, pelaporan, dan audit. Untuk memulai transfer file ke server SFTP jarak jauh, Anda menggunakan operasi StartFileTransferAPI, yang menggunakan konektor SFTP untuk melakukan transfer. Setiap `StartFileTransfer` permintaan dapat berisi 10 jalur berbeda.

Anda dapat memantau transfer file Anda dengan memeriksa log server Anda. Aktivitas konektor dicatat ke aliran log yang memiliki format`aws/transfer/`*`connector-id`*, misalnya,`aws/transfer/c-1234567890abcdef0`. Jika Anda tidak melihat log apa pun untuk konektor Anda, pastikan Anda telah menentukan peran logging dengan izin yang benar untuk konektor Anda.

Untuk detail tentang membuat konektor, lihatKonfigurasikan konektor SFTP.

Untuk mengirim dan mengambil file dengan menggunakan konektor SFTP, Anda menggunakan perintah `start-file-transfer` AWS Command Line Interface ()AWS CLI. Anda menentukan parameter berikut, tergantung apakah Anda mengirim file (transfer keluar) atau menerima file (transfer masuk).

- Transfer keluar
  - `send-file-paths`berisi dari satu hingga sepuluh jalur file sumber, untuk file yang akan ditransfer ke server SFTP mitra.
  - `remote-directory-path`adalah jalur jarak jauh untuk mengirim file ke server SFTP pelanggan.
- Transfer masuk
  - `retrieve-file-paths`berisi dari satu hingga sepuluh jalur jarak jauh. Setiap jalur menentukan lokasi untuk mentransfer file dari server SFTP mitra ke server Transfer Family Anda.

- `local-directory-path`adalah lokasi Amazon S3 (bucket dan awalan opsional) tempat file Anda disimpan.

Untuk mengirim file, Anda menentukan `remote-directory-path` parameter `send-file-paths` dan. Anda dapat menentukan hingga 10 file untuk `send-file-paths` parameter. Contoh perintah berikut mengirimkan file bernama */DOC-EXAMPLE-SOURCE-BUCKET*`/file1.txt` dan*/DOC-EXAMPLE-SOURCE-BUCKET*`/file2.txt`, terletak di penyimpanan Amazon S3, ke `/tmp` direktori di server SFTP mitra Anda. Untuk menggunakan perintah contoh ini, ganti *DOC-EXAMPLE-SOURCE-BUCKET* dengan bucket Anda sendiri.

```
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
```

Untuk menerima file, Anda menentukan `local-directory-path` parameter `retrieve-file-paths` dan. *Contoh berikut mengambil file /my/remote/file1.txt dan /my/remote/file2.txt di server SFTP mitra, dan menempatkannya di lokasi Amazon S3/awalan DOC-EXAMPLE-BUCKET/.* Untuk menggunakan contoh perintah ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
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
```

Contoh sebelumnya menentukan jalur absolut pada server SFTP. Anda juga dapat menggunakan jalur relatif: yaitu jalur yang relatif terhadap direktori home pengguna SFTP. Misalnya, jika pengguna SFTP `marymajor` dan direktori home mereka di server SFTP adalah`/users/marymajor/`, perintah berikut dikirim ke */DOC-EXAMPLE-SOURCE-BUCKET*`/file1.txt /users/marymajor/test-connectors/file1.txt`

```
aws transfer start-file-transfer --send-file-paths /DOC-EXAMPLE-SOURCE-BUCKET/file1.txt
 \
   --remote-directory-path test-connectors --connector-id c-2222BBBB3333CCCC4 --
region us-east-2
```

Kelola konektor SFTP

Topik ini menjelaskan cara melihat dan memperbarui konektor SFTP, dan mencantumkan kuota yang relevan untuk konektor SFTP.

> **Note**
>
> Setiap konektor secara otomatis diberi alamat IP statis yang tetap tidak berubah selama masa pakai konektor. Ini memungkinkan Anda untuk terhubung dengan server SFTP jarak jauh yang hanya menerima koneksi masuk dari alamat IP yang diketahui. Konektor Anda diberi satu set alamat IP statis yang dibagikan oleh semua konektor menggunakan protokol yang sama (SFTP atau AS2) di konektor Anda. Akun AWS

Topik

- Perbarui konektor SFTP
- Lihat detail konektor SFTP
- Kuota untuk konektor SFTP

Perbarui konektor SFTP

Untuk mengubah nilai parameter yang ada untuk konektor Anda, Anda dapat menjalankan `update-connector` perintah. Perintah berikut memperbarui rahasia untuk konektor*connector-id*, di Wilayah *region-id* ke*secret-ARN*. Untuk menggunakan contoh perintah ini, ganti *user input placeholders* dengan informasi Anda sendiri.

```
aws transfer update-connector --sftp-config '{"UserSecretId":"secret-ARN"}' \
   --connector-id connector-id --region region-id
```

Lihat detail konektor SFTP

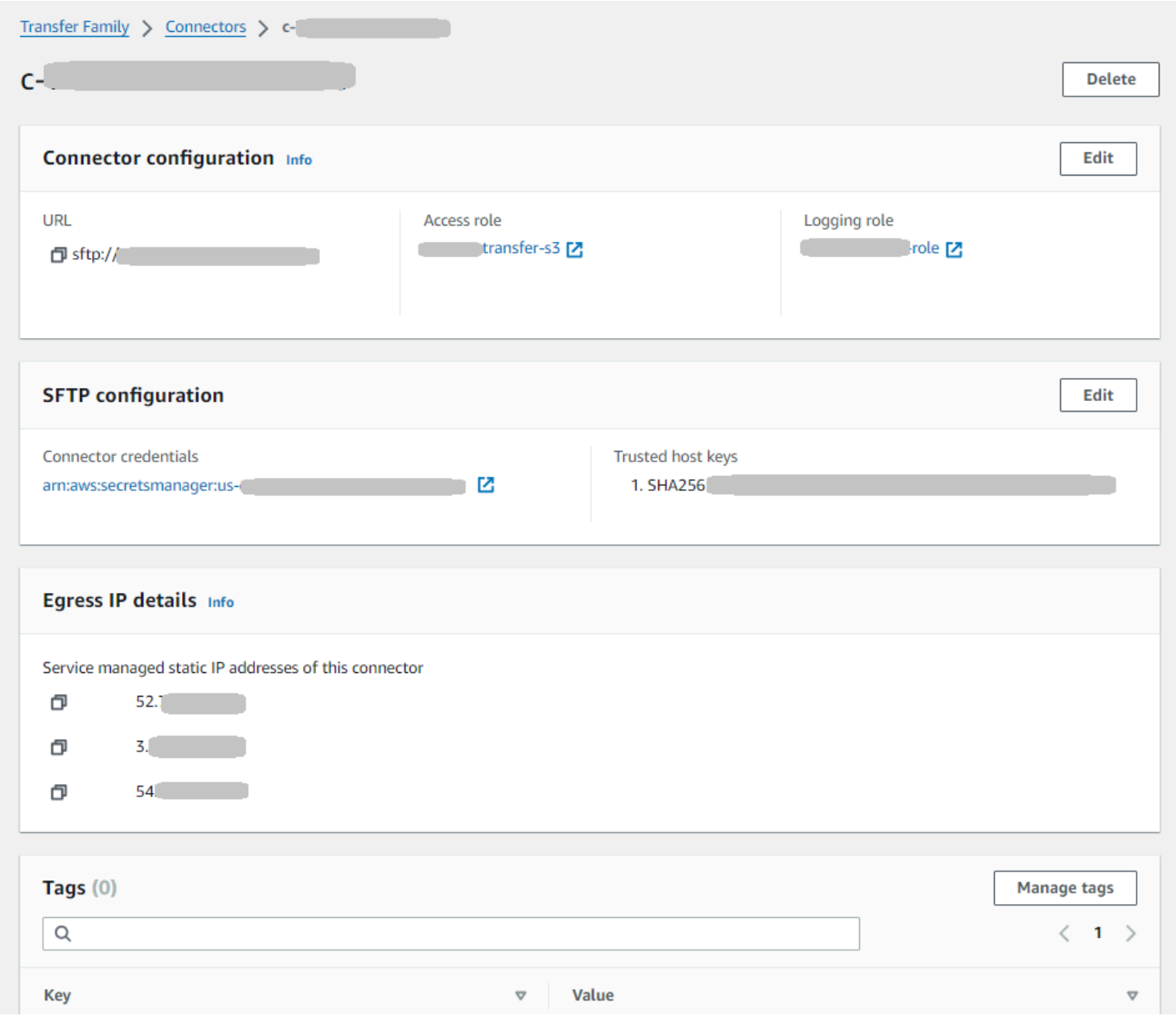
Anda dapat menemukan daftar detail dan properti untuk konektor SFTP di konsol. AWS Transfer Family

Untuk melihat detail konektor

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.

2. Di panel navigasi kiri, pilih Konektor.
3. Pilih pengenal di kolom Connector ID untuk melihat halaman detail untuk konektor yang dipilih.

Anda dapat mengubah properti untuk konektor SFTP dengan memilih Edit pada halaman detail konektor.



> **Note**
>
> Anda bisa mendapatkan banyak informasi ini, meskipun dalam format yang berbeda, dengan menjalankan perintah AWS Command Line Interface (AWS CLI) berikut. Untuk menggunakan contoh perintah ini, ganti *user input placeholders* dengan informasi Anda sendiri.
>
> ```
> aws transfer describe-connector --connector-id your-connector-id
> ```
>
> Untuk informasi selengkapnya, lihat DescribeConnectordi referensi API.

Kuota untuk konektor SFTP

Kuota berikut tersedia untuk konektor SFTP. Kuota untuk konektor AS2 dijelaskan dalam. Kuota dan batasan AS2 Untuk meminta kenaikan kuota yang dapat disesuaikan, lihat Layanan AWS kuota di. Referensi Umum AWS

Kuota konektor SFTP

Nama	Default	Dapat disesuaikan
Transaksi koneksi uji maksimum per detik (TPS)	1 permintaan per detik, per akun	Tidak
TPS `StartFileTransfer` Maksimum	3 permintaan per detik, per akun	Ya
Ukuran antrian maksimum untuk transfer file yang tertunda	1000	Tidak
Ukuran maksimum file	50 gibibyte (GiB)	Tidak
Waktu transfer maksimum per file	6 jam	Tidak
Waktu tunggu permintaan maksimum per file	6 jam	Tidak

Nama	Default	Dapat disesuaikan
Durasi minimum `AccessRole` atau `LoggingRole` sesi	60 menit	Tidak
Transfer file bersamaan maksimum	1 transfer file bersamaan per konektor	Tidak
Jumlah maksimum permintaan transfer file per detik per akun	3	Ya
Jumlah maksimum konektor per akun (konektor SFTP dan AS2 berkontribusi pada hitungan ini)	100	Ya
Bandwidth maksimum untuk konektor per akun (konektor SFTP dan AS2 berkontribusi pada nilai ini)	50 MBps	Tidak

Untuk menyimpan kredensi untuk konektor SFTP, ada kuota yang terkait dengan setiap rahasia Secrets Manager. Jika Anda menggunakan rahasia yang sama untuk menyimpan beberapa jenis kunci, untuk berbagai tujuan, Anda mungkin menemukan kuota ini.

- Total panjang untuk satu rahasia: 12.000 karakter
- Panjang maksimum **`Password`** string: 1024 karakter
- Panjang maksimum **`PrivateKey`** string: 8192 karakter
- Panjang maksimum **`Username`** string: 100 karakter

AWS Transfer Family untuk AS2

Applicability Statement 2 (AS2) adalah spesifikasi transmisi file yang ditentukan RFC yang mencakup perlindungan pesan dan mekanisme verifikasi yang kuat. Protokol AS2 sangat penting untuk alur kerja dengan persyaratan kepatuhan yang bergantung pada perlindungan data dan fitur keamanan yang dibangun ke dalam protokol.

> **Note**
>
> AS2 untuk Transfer Family bersertifikat Drummond.

Pelanggan di industri seperti ritel, ilmu hayati, manufaktur, layanan keuangan, dan utilitas yang mengandalkan AS2 untuk rantai pasokan, logistik, dan alur kerja pembayaran dapat menggunakan titik akhir AWS Transfer Family AS2 untuk bertransaksi dengan aman dengan mitra bisnis mereka. Data yang ditransaksikan dapat diakses secara native AWS untuk pemrosesan, analisis, dan pembelajaran mesin. Data ini juga tersedia untuk integrasi dengan sistem perencanaan sumber daya perusahaan (ERP) dan manajemen hubungan pelanggan (CRM) yang berjalan. AWS Dengan AS2, pelanggan dapat menjalankan transaksi business-to-business (B2B) mereka dalam skala besar AWS sambil mempertahankan integrasi dan kepatuhan mitra bisnis yang ada.

Jika Anda adalah pelanggan Transfer Family yang ingin bertukar file dengan mitra yang memiliki server berkemampuan AS2 yang dikonfigurasi, penyiapannya melibatkan pembuatan satu key pair publik-pribadi untuk enkripsi dan satu lagi untuk menandatangani dan menukar kunci publik dengan mitra.

Kami memiliki lokakarya yang dapat Anda hadiri, di mana Anda dapat mengonfigurasi titik akhir Transfer Family dengan AS2 diaktifkan, dan konektor Transfer Family AS2 Anda dapat melihat detail untuk lokakarya ini di sini.

Melindungi muatan AS2 dalam perjalanan biasanya melibatkan penggunaan Cryptographic Message Syntax (CMS) dan umumnya menggunakan enkripsi dan tanda tangan digital untuk memberikan perlindungan data dan otentikasi rekan. Payload respons Pemberitahuan Disposisi Pesan (MDN) yang ditandatangani memberikan verifikasi (non-penolakan) bahwa pesan diterima dan berhasil didekripsi.

Transportasi muatan CMS ini dan respons MDN terjadi melalui HTTP.

> **Note**
>
> Titik akhir server HTTPS AS2 saat ini tidak didukung. Pengakhiran TLS saat ini menjadi tanggung jawab pelanggan.

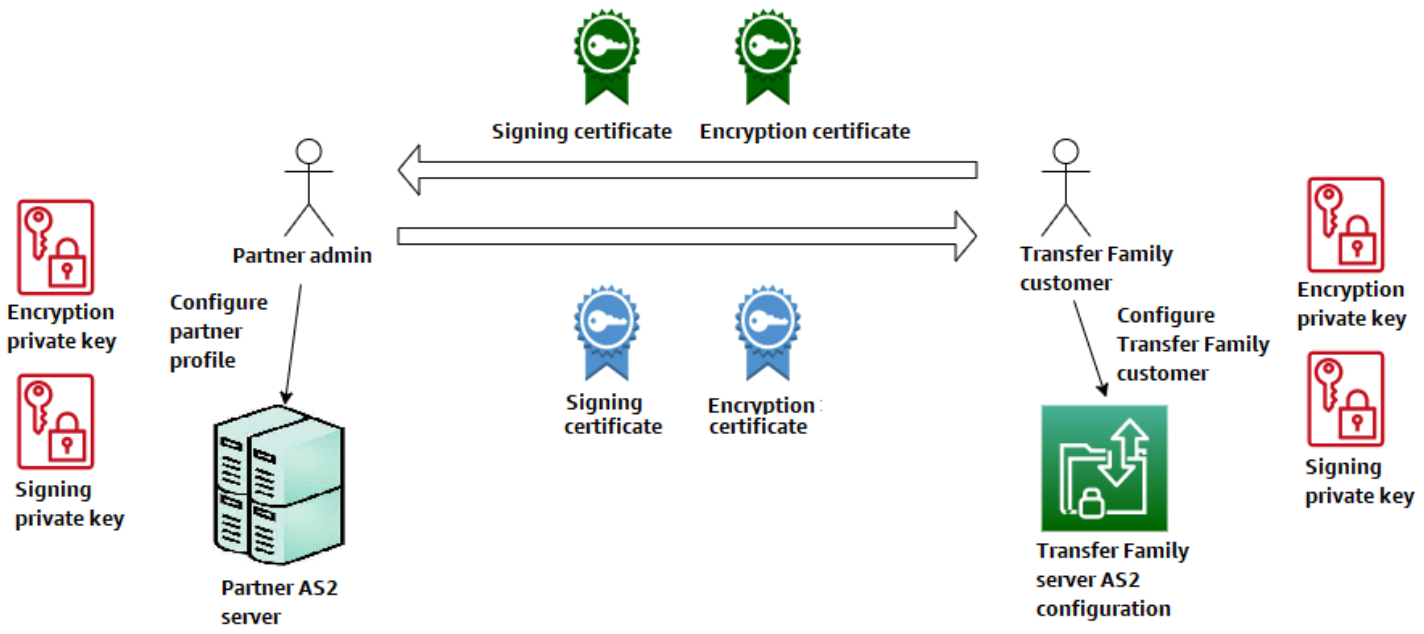
Untuk step-by-step panduan rinci tentang pengaturan konfigurasi Applicability Statement 2 (AS2), lihat tutorialnya,. Menyiapkan konfigurasi AS2

Topik

- Kasus penggunaan AS2
- Mengkonfigurasi AS2
- Konfigurasikan konektor AS2
- Kelola mitra AS2
- Mengirim dan menerima pesan AS2
- Memantau penggunaan AS2

Kasus penggunaan AS2

Jika Anda adalah AWS Transfer Family pelanggan yang ingin bertukar file dengan mitra yang memiliki server AS2 yang dikonfigurasi, bagian paling kompleks dari pengaturan melibatkan pembuatan satu public-private key pair untuk enkripsi dan satu lagi untuk menandatangani dan bertukar kunci publik dengan mitra.



Pertimbangkan variasi berikut untuk digunakan AWS Transfer Family dengan AS2.

> **Note**
>
> Mitra dagang adalah mitra yang terkait dengan profil mitra tersebut.
> Semua penyebutan MDN dalam tabel berikut mengasumsikan mDNS ditandatangani.

Kasus penggunaan AS2

Kasus penggunaan khusus masuk

- Transfer pesan AS2 terenkripsi dari mitra dagang ke server Transfer Family.

  Dalam kasus ini, lakukan hal berikut:

  1. Buat profil untuk mitra dagang Anda dan diri Anda sendiri.
  2. Buat server Transfer Family yang menggunakan protokol AS2.
  3. Buat perjanjian dan tambahkan ke server Anda.
  4. Impor sertifikat dengan kunci pribadi dan tambahkan ke profil Anda, lalu impor kunci publik ke profil mitra Anda untuk enkripsi.
  5. Setelah Anda memiliki barang-barang ini, kirimkan kunci publik untuk sertifikat Anda ke mitra dagang Anda.

Sekarang pasangan Anda dapat mengirimi Anda pesan terenkripsi dan Anda dapat mendekripsi dan menyimpannya di ember Amazon S3 Anda.

- Transfer pesan AS2 terenkripsi dari mitra dagang ke server Transfer Family dan tambahkan penandatanganan.

  Dalam skenario ini, Anda masih hanya melakukan transfer masuk, tetapi sekarang Anda ingin pasangan Anda menandatangani pesan yang mereka kirim. Dalam hal ini, impor kunci publik penandatanganan mitra dagang (sebagai sertifikat penandatanganan yang ditambahkan ke profil mitra Anda).
- Transfer pesan AS2 terenkripsi dari mitra dagang ke server Transfer Family dan tambahkan penandatanganan dan pengiriman respons MDN.

  Dalam skenario ini, Anda masih hanya melakukan transfer masuk, tetapi sekarang, selain menerima muatan yang ditandatangani, mitra dagang Anda ingin menerima tanggapan MDN yang ditandatangani.

  1. Impor kunci penandatanganan publik dan pribadi Anda (sebagai sertifikat penandatanganan ke profil Anda).
  2. Kirim kunci penandatanganan publik ke mitra dagang Anda.

Kasus penggunaan khusus keluar

- Transfer pesan AS2 terenkripsi dari server Transfer Family ke mitra dagang.

  Kasus ini mirip dengan kasus penggunaan transfer inbound-only, kecuali bahwa alih-alih menambahkan perjanjian ke server AS2 Anda, Anda membuat konektor. Dalam hal ini, Anda mengimpor kunci publik mitra dagang Anda ke profil mereka.
- Transfer pesan AS2 terenkripsi dari server Transfer Family ke mitra dagang dan tambahkan penandatanganan.

  Anda masih hanya melakukan transfer keluar, tetapi sekarang mitra dagang Anda ingin Anda menandatangani pesan yang Anda kirim kepada mereka.

  1. Impor kunci pribadi penandatanganan Anda (sebagai sertifikat penandatanganan yang ditambahkan ke profil Anda).
  2. Kirimkan kunci publik kepada mitra dagang Anda.
- Transfer pesan AS2 terenkripsi dari server Transfer Family ke mitra dagang dan tambahkan penandatanganan dan kirim tanggapan MDN.

  Anda masih hanya melakukan transfer keluar, tetapi sekarang, selain mengirim muatan yang ditandatangani, Anda ingin menerima tanggapan MDN yang ditandatangani dari mitra dagang Anda.

  1. Mitra dagang Anda mengirimi Anda kunci penandatanganan publik mereka.
  2. Impor kunci publik mitra dagang Anda (sebagai sertifikat penandatanganan yang ditambahk an ke profil mitra Anda).

Kasus penggunaan masuk dan keluar

- Transfer pesan AS2 terenkripsi di kedua arah antara server Transfer Family dan mitra dagang.

  Dalam kasus ini, lakukan hal berikut:

  1. Buat profil untuk mitra dagang Anda dan diri Anda sendiri.
  2. Buat server Transfer Family yang menggunakan protokol AS2.
  3. Buat perjanjian dan tambahkan ke server Anda.
  4. Buat konektor.
  5. Impor sertifikat dengan kunci pribadi dan tambahkan ke profil Anda, lalu impor kunci publik ke profil mitra Anda untuk enkripsi.
  6. Terima kunci publik dari mitra dagang Anda dan tambahkan ke profil mereka untuk enkripsi.
  7. Setelah Anda memiliki barang-barang ini, kirimkan kunci publik untuk sertifikat Anda ke mitra dagang Anda.

  Sekarang Anda dan mitra dagang Anda dapat bertukar pesan terenkripsi, dan Anda berdua dapat mendekripsi mereka. Anda dapat menyimpan pesan yang Anda terima di bucket Amazon S3 Anda, dan pasangan Anda dapat mendekripsi dan menyimpan pesan yang Anda kirim kepada mereka.
- Transfer pesan AS2 terenkripsi di kedua arah antara server Transfer Family dan mitra dagang dan tambahkan penandatanganan.

  Sekarang Anda dan pasangan Anda ingin pesan yang ditandatangani.

  1. Impor kunci pribadi penandatanganan Anda (sebagai sertifikat penandatanganan yang ditambahkan ke profil Anda).
  2. Kirimkan kunci publik kepada mitra dagang Anda.
  3. Impor kunci publik penandatanganan mitra dagang Anda dan tambahkan ke profil mereka.
- Transfer pesan AS2 terenkripsi di kedua arah antara server Transfer Family dan mitra dagang dan tambahkan penandatanganan dan kirim respons MDN.

  Sekarang, Anda ingin menukar muatan yang ditandatangani, dan Anda dan mitra dagang Anda menginginkan tanggapan MDN.

  1. Mitra dagang Anda mengirimi Anda kunci penandatanganan publik mereka.
  2. Impor kunci publik mitra dagang Anda (sebagai sertifikat penandatanganan ke profil mitra Anda).

3. Kirim kunci publik Anda ke mitra dagang Anda.

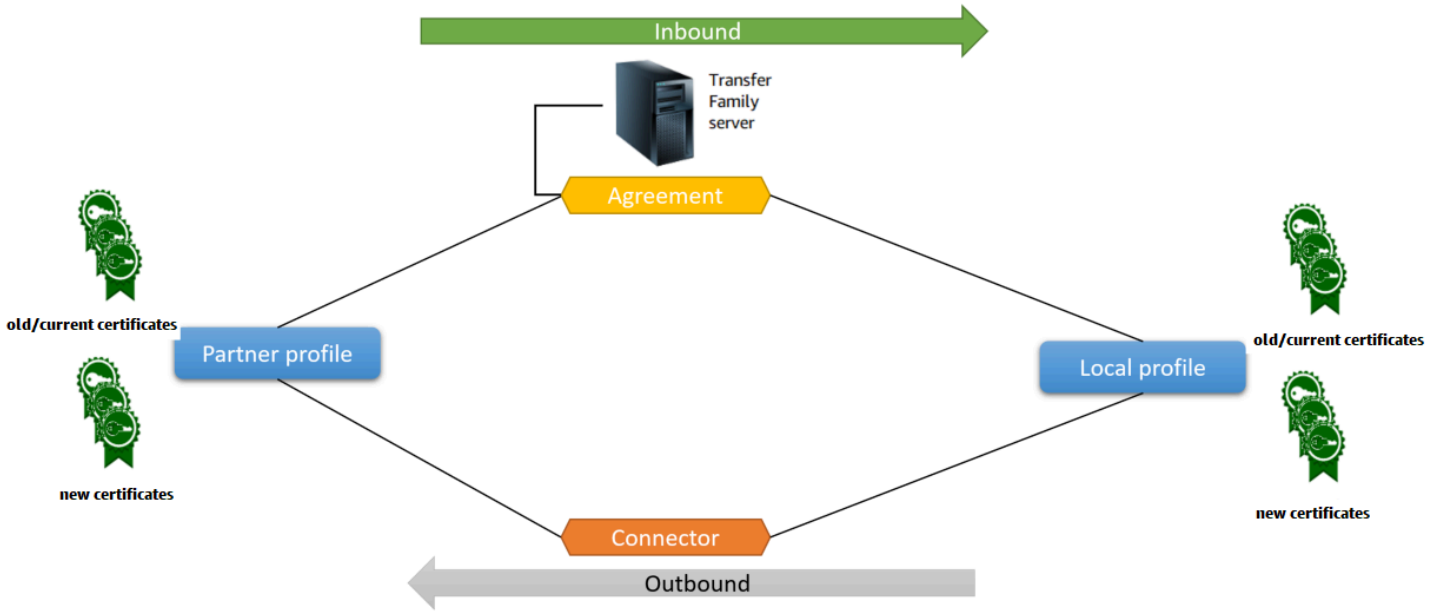
Mengkonfigurasi AS2

Untuk membuat server berkemampuan AS2, Anda juga harus menentukan komponen berikut:

- Perjanjian — Perjanjian mitra dagang bilateral, atau kemitraan, menentukan hubungan antara kedua pihak yang bertukar pesan (file). Untuk menentukan perjanjian, Transfer Family menggabungkan server, profil lokal, profil mitra, dan informasi sertifikat. Transfer Family AS2-Proses masuk menggunakan perjanjian.
- Sertifikat — Sertifikat kunci publik (X.509) digunakan dalam komunikasi AS2 untuk enkripsi dan verifikasi pesan. Sertifikat juga digunakan untuk titik akhir konektor.
- Profil lokal dan profil mitra — Profil lokal mendefinisikan organisasi atau “pesta” lokal (AS2 enabled Transfer Family server). Demikian pula, profil mitra mendefinisikan organisasi mitra jarak jauh, eksternal untuk Transfer Family.

Meskipun tidak diperlukan untuk semua server yang mendukung AS2, untuk transfer keluar, Anda memerlukan konektor. Konektor menangkap parameter untuk koneksi keluar. Konektor diperlukan untuk mengirim file ke eksternal pelanggan, non AWS server.

Diagram berikut menunjukkan hubungan antara objek AS2 yang terlibat dalam proses masuk dan keluar.



Untuk end-to-end contoh konfigurasi AS2, lihatMenyiapkan konfigurasi AS2.

Topik

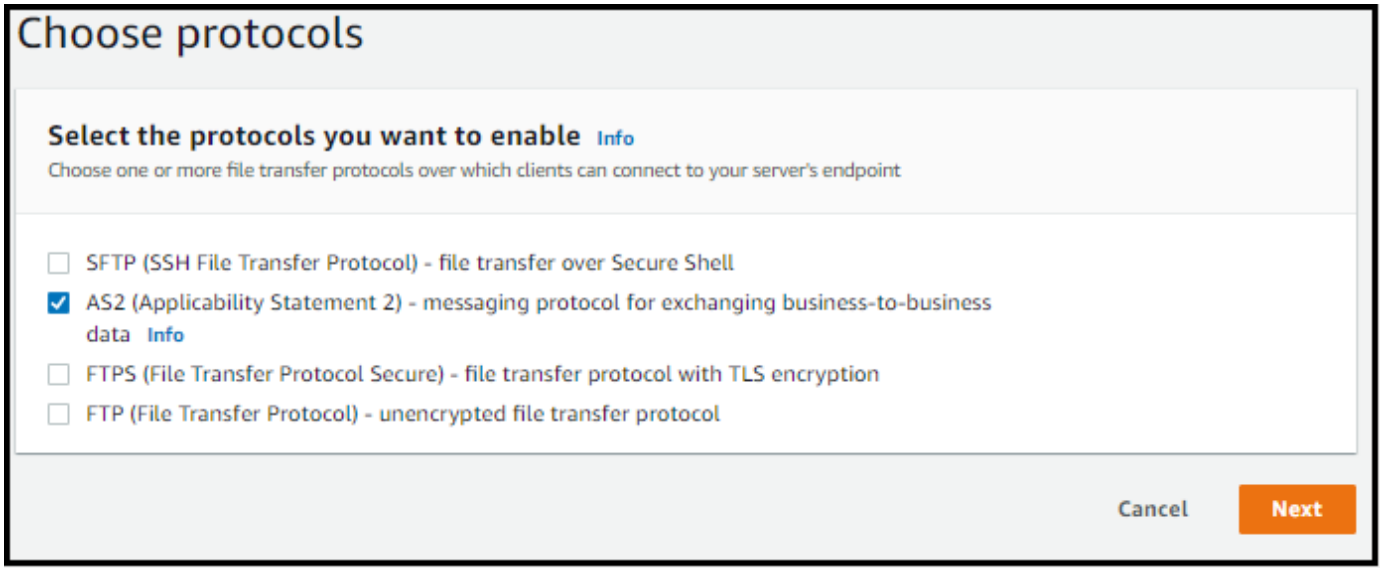
- Membuat server AS2 menggunakan konsol Transfer Family
- Gunakan template untuk membuat demo Transfer Family AS2 stack
- Konfigurasi dan batasan AS2
- Fitur dan kemampuan AS2

Membuat server AS2 menggunakan konsol Transfer Family

Prosedur ini menjelaskan cara membuat server berkemampuan AS2 dengan menggunakan konsol Transfer Family. Jika Anda ingin menggunakan AWS CLI sebagai gantinya, lihatthe section called "Langkah 2: Buat server Transfer Family yang menggunakan protokol AS2".

Untuk membuat server berkemampuan AS2

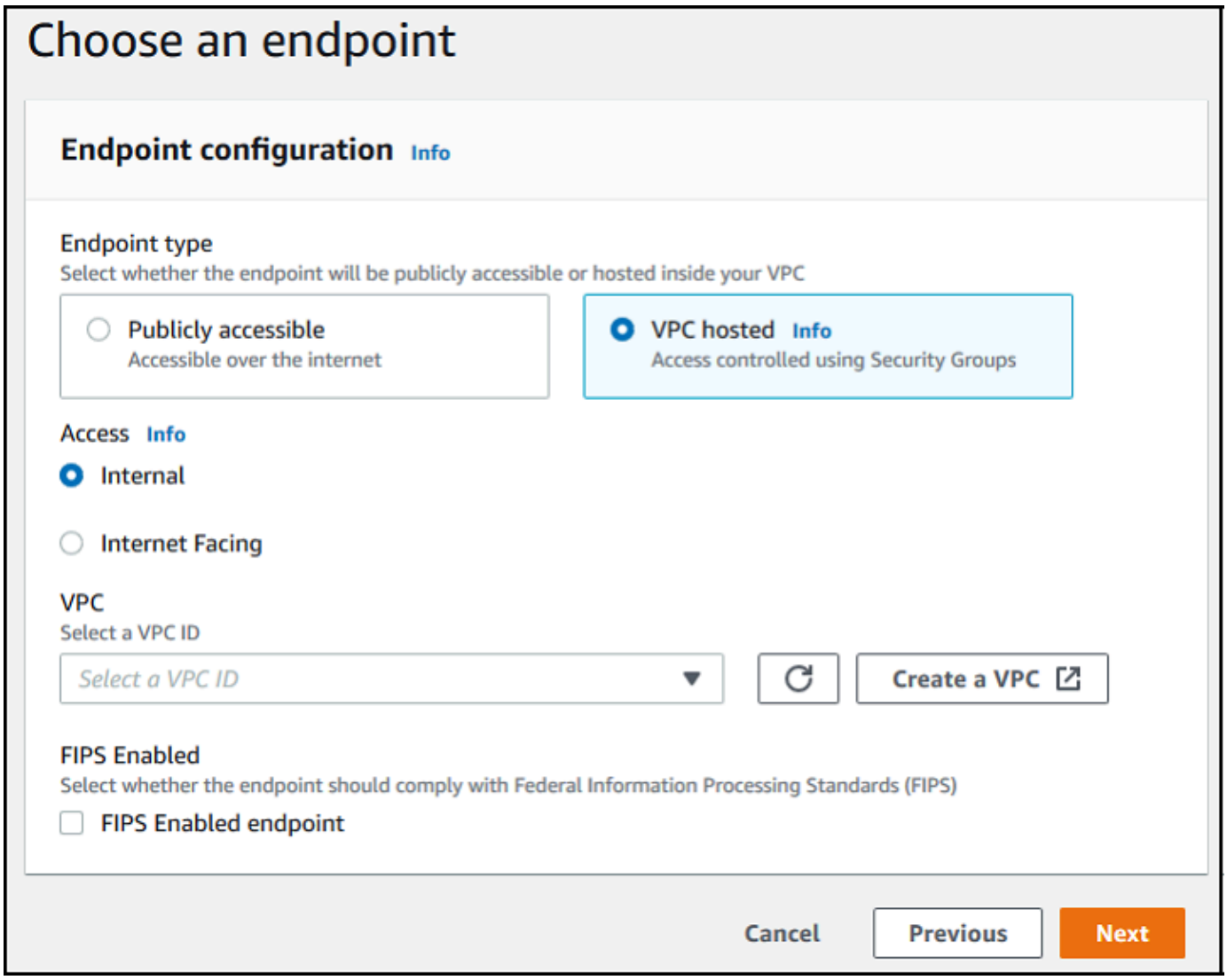
1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi kiri, pilih Server, lalu pilih Buat server.
3. Pada halaman Pilih protokol, pilih AS2 (Pernyataan Penerapan 2), lalu pilih Berikutnya.



4. Pada halaman Pilih penyedia identitas, pilih Berikutnya.

> **Note**
>
> Untuk AS2, Anda tidak dapat memilih penyedia identitas karena otentikasi dasar tidak didukung untuk protokol AS2. Sebagai gantinya, Anda mengontrol akses melalui grup keamanan virtual private cloud (VPC).

5. Pada halaman Pilih titik akhir, lakukan hal berikut:



a. Untuk jenis Endpoint, pilih VPC yang dihosting untuk meng-host endpoint server Anda. Untuk informasi tentang menyiapkan titik akhir yang dihosting VPC, lihat. Buat server di cloud pribadi virtual

> **Note**
>
> Titik akhir yang dapat diakses publik tidak didukung untuk protokol AS2. Untuk membuat titik akhir VPC Anda dapat diakses melalui internet, pilih Internet Facing di bawah Access, lalu berikan alamat IP Elastic Anda.

b. Untuk Access, pilih salah satu opsi berikut:

- Internal — Pilih opsi ini untuk menyediakan akses dari dalam lingkungan yang terhubung dengan VPC dan VPC Anda, seperti pusat data lokal di atas atau VPN. AWS Direct Connect
- Menghadapi Internet — Pilih opsi ini untuk menyediakan akses melalui internet dan dari dalam lingkungan yang terhubung dengan VPC dan VPC Anda, seperti pusat data lokal di atas atau VPN. AWS Direct Connect

  Jika Anda memilih Internet Facing, berikan alamat IP Elastis Anda saat diminta.

c. Untuk VPC, pilih VPC yang sudah ada atau pilih Create VPC untuk membuat VPC baru.

d. Untuk FIPS Diaktifkan, biarkan kotak centang titik akhir FIPS Enabled tetap bersih.

> **Note**
>
> Titik akhir berkemampuan FIPS tidak didukung untuk protokol AS2.

e. Pilih Berikutnya.

6. Pada halaman Pilih domain, pilih Amazon S3 untuk menyimpan dan mengakses file Anda sebagai objek dengan menggunakan protokol yang dipilih.

   Pilih Berikutnya.

7. Pada halaman Konfigurasi detail tambahan, pilih pengaturan yang Anda butuhkan.

> **Note**
>
> Jika Anda mengonfigurasi protokol lain bersama dengan AS2, semua pengaturan detail tambahan berlaku. Namun, untuk protokol AS2, satu-satunya pengaturan yang berlaku adalah yang ada di bagian CloudWatch logging dan Tags.

Meskipun pengaturan peran CloudWatch logging adalah opsional, kami sangat menyarankan untuk mengaturnya sehingga Anda dapat melihat status pesan Anda dan memecahkan masalah konfigurasi.

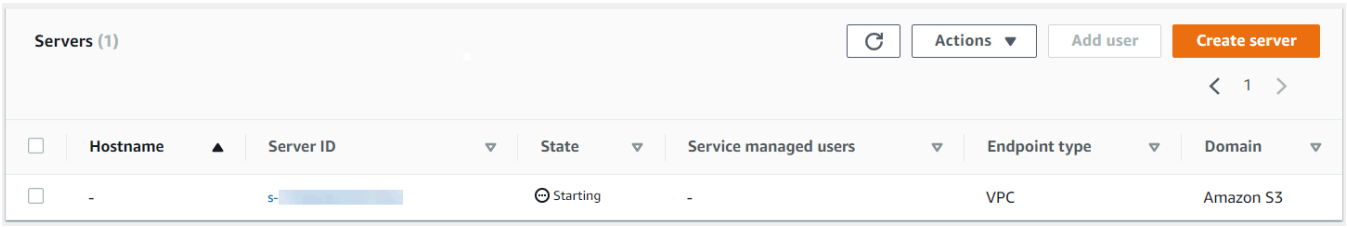
8. Pada halaman Tinjau dan buat, tinjau pilihan Anda untuk memastikannya benar.

   - Jika Anda ingin mengedit pengaturan apa pun, pilih Edit di samping langkah yang ingin Anda ubah.

     Note

     Jika Anda mengedit langkah, kami sarankan Anda meninjau setiap langkah setelah langkah yang Anda pilih untuk diedit.

   - Jika Anda tidak memiliki perubahan, pilih Buat server untuk membuat server Anda. Anda dibawa ke halaman Server, ditampilkan berikut, di mana server baru Anda terdaftar.



   Diperlukan beberapa menit sebelum status server baru Anda berubah menjadi Online. Pada saat itu, server Anda dapat melakukan operasi file untuk pengguna Anda.

Gunakan template untuk membuat demo Transfer Family AS2 stack

Kami menyediakan AWS CloudFormation template mandiri untuk membuat server Transfer Family AS2 dengan cepat. Template mengonfigurasi server dengan titik akhir VPC Amazon publik, sertifikat, profil lokal dan mitra, perjanjian, dan konektor.

Sebelum menggunakan template ini, perhatikan hal berikut:

- Jika Anda membuat tumpukan dari template ini, Anda akan ditagih untuk AWS sumber daya yang digunakan.
- Template membuat beberapa sertifikat dan menempatkannya AWS Secrets Manager untuk menyimpannya dengan aman. Anda dapat menghapus sertifikat ini dari Secrets Manager jika mau, karena Anda dikenakan biaya untuk menggunakan layanan ini. Menghapus sertifikat ini di Secrets

Manager tidak akan menghapusnya dari server Transfer Family. Oleh karena itu, fungsionalitas tumpukan demo tidak terpengaruh. Namun, untuk sertifikat yang akan Anda gunakan dengan server produksi AS2, Anda mungkin ingin menggunakan Secrets Manager untuk mengelola dan memutar sertifikat tersimpan secara berkala.

- Kami menyarankan Anda menggunakan template sebagai basis saja, dan terutama untuk tujuan demonstrasi. Jika Anda ingin menggunakan tumpukan demo ini dalam produksi, kami sarankan Anda memodifikasi kode YAMM template untuk membuat tumpukan yang lebih kuat. Misalnya, buat sertifikat tingkat produksi, dan buat AWS Lambda fungsi yang dapat Anda gunakan dalam produksi.

Untuk membuat server Transfer Family AS2 berkemampuan dari template CloudFormation

1. Buka AWS CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
2. Di panel navigasi sebelah kiri, pilih Tumpukan.
3. Pilih Buat tumpukan, lalu pilih Dengan sumber daya baru (standar).
4. Di bagian Prasyarat - Siapkan templat, pilih Template sudah siap.
5. Salin tautan ini, templat demo AS2, dan tempel ke bidang URL Amazon S3.
6. Pilih Berikutnya.
7. Pada halaman Tentukan detail tumpukan, beri nama tumpukan Anda, lalu tentukan parameter berikut:

   - Di bawah AS2, masukkan nilai untuk ID AS2 Lokal dan ID AS2 Mitra, atau terima default, dan, masing-masing. `local` `partner`
   - Di bawah Jaringan, masukkan nilai untuk masuk grup Keamanan CIDR IP, atau terima default,. `0.0.0.0/0`

     > **Note**
     >
     > Nilai ini, dalam format CIDR, menentukan alamat IP mana yang diizinkan untuk lalu lintas masuk ke server AS2. Nilai default,`0.0.0.0/0`, memungkinkan semua alamat IP.

   - Di bawah Umum, masukkan nilai untuk Awalan, atau terima default,`transfer-as2`. Awalan ini ditempatkan sebelum nama sumber daya apa pun yang dibuat oleh tumpukan. Misalnya, jika Anda menggunakan awalan default, bucket Amazon S3 Anda diberi nama. `transfer-as2-`*`TransferS3BucketName`*

8. Pilih Berikutnya. Pada halaman Configure stack options, pilih Next lagi.
9. Tinjau detail tumpukan yang Anda buat, lalu pilih Buat tumpukan.

> **Note**
>
> Di bagian bawah halaman, di bawah Kemampuan, Anda harus mengakui bahwa AWS CloudFormation mungkin membuat sumber daya AWS Identity and Access Management (IAM).

Setelah tumpukan dibuat, Anda dapat mengirim pesan uji AS2 dari server mitra ke server Transfer Family lokal Anda dengan menggunakan AWS Command Line Interface (AWS CLI). AWS CLIPerintah sampel untuk mengirim pesan pengujian dibuat bersama dengan semua sumber daya lain di tumpukan.

Untuk menggunakan perintah sampel ini, buka tab Output dari tumpukan Anda, dan salin TransferExampleAs2Command. Anda kemudian dapat menjalankan perintah dengan menggunakanAWS CLI. Jika Anda belum menginstalAWS CLI, lihat Menginstal atau memperbarui versi terbaru dari AWS CLI Panduan AWS Command Line Interface Pengguna.

Perintah sampel memiliki format berikut:

```
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**
>
> Versi Anda dari perintah ini berisi nilai aktual untuk *TransferS3BucketName* dan *TransferConnectorId* sumber daya di tumpukan Anda.

Perintah sampel ini terdiri dari dua perintah terpisah yang dirantai bersama dengan menggunakan && string.

Perintah pertama membuat file teks baru yang kosong di bucket Anda:

```
aws s3api put-object --bucket TransferS3BucketName --key test.txt
```

Kemudian, perintah kedua menggunakan konektor untuk mengirim file dari profil mitra ke profil lokal. Server Transfer Family memiliki pengaturan perjanjian yang memungkinkan profil lokal menerima pesan dari profil mitra.

```
aws transfer start-file-transfer --region aws-region --connector-id TransferConnectorId
 --send-file-paths /TransferS3BucketName/test.txt
```

Setelah menjalankan perintah, Anda dapat pergi ke bucket Amazon S3 (*TransferS3BucketName*) dan melihat isinya. Jika perintah berhasil, Anda akan melihat objek berikut di ember Anda:

- `processed/`— Folder ini berisi file JSON yang menjelaskan file yang ditransfer dan respons MDN.
- `processing/`— Folder ini sementara berisi file saat sedang diproses, tetapi setelah transfer selesai, folder ini harus kosong.
- *server-id*`/`— Folder ini diberi nama berdasarkan ID server Transfer Family Anda. Ini berisi `from-`*partner* (folder ini dinamai secara dinamis, berdasarkan ID AS2 mitra), yang berisi, `failed/processed/`, dan `processing/` folder dengan sendirinya. `/`*server-id*`/from-`*partner*`/processed/`Folder berisi salinan file teks yang ditransfer, dan file JSON dan MDN yang sesuai.
- `test.txt`— Objek ini adalah file (kosong) yang ditransfer.

Konfigurasi dan batasan AS2

Topik ini menjelaskan konfigurasi, fitur, dan kemampuan yang didukung untuk transfer yang menggunakan protokol Applicability Statement 2 (AS2), termasuk cipher dan digest yang diterima. Bagian ini juga menjelaskan batasan dan masalah yang diketahui untuk transfer AS2.

Topik

- Konfigurasi yang didukung AS2
- Kuota dan batasan AS2

Konfigurasi yang didukung AS2

Penandatanganan, enkripsi, kompresi, MDN

Untuk transfer masuk dan keluar, item berikut wajib atau opsional:

- Enkripsi — Diperlukan (untuk transportasi HTTP, yang merupakan satu-satunya metode transport yang saat ini didukung). Pesan yang tidak terenkripsi hanya diterima jika diteruskan oleh proxy penghentian TLS seperti Application Load Balancer (ALB) dan header ada. `X-Forwarded-Proto: https`
- Penandatanganan - Opsional
- Kompresi - Opsional (satu-satunya algoritma kompresi yang didukung saat ini adalah ZLIB)
- Pemberitahuan Disposisi Pesan (MDN) — Opsional

Cipher

Cipher berikut didukung untuk transfer masuk dan keluar:

- AES128_CBC
- AES192_CBC
- AES256_CBC
- 3DES (hanya untuk kompatibilitas mundur)

Intisari

Intisari berikut didukung:

- Penandatanganan masuk dan MDN - SHA1, SHA256, SHA384, SHA512
- Penandatanganan keluar dan MDN - SHA1, SHA256, SHA384, SHA512

MDN

Untuk tanggapan MDN, jenis tertentu didukung, sebagai berikut:

- Transfer masuk — Sinkron dan asinkron
- Transfer keluar — Hanya sinkron
- Protokol Transfer Surat Sederhana (SMTP) (email MDN) - Tidak didukung

Transportasi

- Transfer masuk - HTTP adalah satu-satunya transportasi yang didukung saat ini, dan Anda harus menentukannya secara eksplisit.

> **Note**
>
> Jika Anda perlu menggunakan HTTPS untuk transfer masuk, Anda dapat menghentikan TLS pada Application Load Balancer atau Network Load Balancer. Ini dijelaskan dalamTerima pesan AS2 melalui HTTPS.

- Transfer keluar — Jika Anda memberikan URL HTTP, Anda juga harus menentukan algoritma enkripsi. Jika Anda memberikan URL HTTPS, Anda memiliki opsi untuk menentukan NONE untuk algoritme enkripsi Anda.

Kuota dan batasan AS2

Bagian ini membahas kuota dan batasan untuk AS2

Topik

- Kuota AS2
- Kuota untuk menangani rahasia
- Keterbatasan yang Sudah Diketahui

Kuota AS2

Kuota berikut tersedia untuk transfer file AS2. Untuk meminta kenaikan kuota yang dapat disesuaikan, lihat Layanan AWS kuota di. Referensi Umum AWS

Kuota AS2

Nama	Default	Dapat disesuaikan
Permintaan AS2 masuk per server	25 per detik	Tidak
Permintaan AS2 masuk sedang berlangsung per server	100	Tidak
Jumlah maksimum file per permintaan transfer file	10	Tidak

Nama	Default	Dapat disesuaikan
Permintaan AS2 keluar sedang berlangsung per konektor	100	Tidak
Ukuran file maksimum (terkompresi atau tidak terkompresi)	50 MiB	Ya
Batas waktu tidak aktif	350 detik	Tidak
Jumlah maksimum profil mitra per akun	1000 (hingga 10 sertifikat per profil mitra: tidak dapat disesuaikan)	Ya
Jumlah maksimum sertifikat per akun	1000	Ya
Jumlah maksimum permintaan transfer file per detik per akun	3	Ya
Jumlah maksimum konektor per akun (konektor SFTP dan AS2 berkontribusi pada hitungan ini)	100	Ya
Bandwidth maksimum untuk konektor per akun (konektor SFTP dan AS2 berkontribusi pada nilai ini)	50 MBps	Tidak
Jumlah maksimum perjanjian per server	100	Ya

Kuota untuk menangani rahasia

AWS Transfer Family melakukan panggilan ke AWS Secrets Manager atas nama pelanggan AS2 yang menggunakan otentikasi Dasar. Selain itu Secrets Manager membuat panggilan ke AWS KMS.

> **Note**
>
> Kuota ini tidak spesifik untuk penggunaan rahasia Anda untuk Transfer Family: mereka dibagi di antara semua layanan di Anda Akun AWS.

Untuk Secrets Manager`GetSecretValue`, kuota yang berlaku adalah Combined rate of DescribeSecret dan permintaan GetSecretValue API, seperti yang dijelaskan dalam AWS Secrets Manager kuota.

Secrets Manager **`GetSecretValue`**

Nama	Nilai	Deskripsi
Tingkat gabungan permintaan DescribeSecret dan GetSecret Value API	Setiap Wilayah yang didukung: 10.000 per detik	Transaksi maksimum per detik untuk DescribeSecret dan permintaan GetSecretValue API digabungkan.

Untuk AWS KMS, kuota berikut berlaku untuk`Decrypt`. Untuk detailnya, lihat Meminta kuota untuk setiap operasi AWS KMS API

AWS KMS **`Decrypt`**

Nama kuota	Nilai default (permintaan per detik)

Tingkat permintaan operasi kriptografi (simetris)	Kuota bersama ini bervariasi dengan Wilayah AWS dan jenis AWS KMS kunci yang digunakan dalam permintaan. Setiap kuota dihitung secara terpisah.
	• 5.500 (bersama)
	• 10.000 (bersama) di Wilayah berikut:
	  • US East (Ohio) us-east-2
	  • Asia Pacific (Singapore) ap-southeast-1
	  • Asia Pacific (Sydney), ap-southeast-2
	  • Asia Pacific (Tokyo), ap-northeast-1

Nama kuota	Nilai default (permintaan per detik)

	• Europe (Frankfurt), eu-central-1
	• Europe (London), eu-west-2
	• 50.000 (bersama) di Wilayah berikut:
	  • US East (N. Virginia), us-east-1
	  • US West (Oregon), us-west-2
	  • Europe (Ireland), eu-west-1

Kuota permintaan toko kunci kustom **Note** Kuota ini hanya berlaku jika Anda menggunakan toko kunci eksternal.	Kuota permintaan toko kunci khusus dihitung secara terpisah untuk setiap toko kunci khusus. • 1.800 (dibagikan) untuk setiap toko AWS CloudHSM kunci • 1.800 (dibagikan) untuk setiap toko kunci eksternal

Keterbatasan yang Sudah Diketahui

- Keep-alive TCP sisi server tidak didukung. Waktu koneksi habis setelah 350 detik tidak aktif kecuali klien mengirim paket keep-alive.
- Agar perjanjian aktif diterima oleh layanan dan muncul di CloudWatch log Amazon, pesan harus berisi header AS2 yang valid.
- Server yang menerima pesan dari AWS Transfer Family AS2 harus mendukung atribut perlindungan algoritma Cryptographic Message Syntax (CMS) untuk memvalidasi tanda tangan pesan, seperti yang didefinisikan dalam RFC 6211. Atribut ini tidak didukung di beberapa produk IBM Sterling yang lebih lama.
- ID pesan duplikat menghasilkan pesan diproses/peringatan: dokumen duplikat.
- Panjang kunci untuk sertifikat AS2 harus minimal 2048 bit, dan paling banyak 4096.
- Saat mengirim pesan AS2 atau mDNS asinkron ke titik akhir HTTPS mitra dagang, pesan atau mDNS harus menggunakan sertifikat SSL yang valid yang ditandatangani oleh otoritas sertifikat tepercaya publik (CA). Sertifikat yang ditandatangani sendiri saat ini hanya didukung untuk transfer keluar.

- Titik akhir harus mendukung protokol TLS versi 1.2 dan algoritma kriptografi yang diizinkan oleh kebijakan keamanan (seperti yang dijelaskan dalam). Kebijakan keamanan untuk AWS Transfer Family server
- Mutual TLS (mTLS) saat ini tidak didukung.
- Beberapa lampiran dan pesan pertukaran sertifikat (CEM) dari AS2 versi 1.2 saat ini tidak didukung.
- Autentikasi dasar saat ini hanya didukung untuk pesan keluar.

Fitur dan kemampuan AS2

Tabel berikut mencantumkan fitur dan kemampuan yang tersedia untuk sumber daya Transfer Family yang menggunakan AS2.

Fitur AS2

Transfer Family menawarkan fitur-fitur berikut untuk AS2.

Fitur	Didukung oleh AWS Transfer Family
Sertifikasi Drummond	Ya
AWS CloudFormation dukungan	Ya
CloudWatchMetrik Amazon	Ya
Algoritma kriptografi SHA-2	Ya
Dukungan untuk Amazon S3	Ya
Dukungan untuk Amazon EFS	Tidak
Pesan Terjadwal	Ya [1]
AWS Transfer Family Alur Kerja Terkelola	Tidak
Pesan Pertukaran Sertifikat (CEM)	Tidak
TLS Bersama (mTL)	Tidak

Fitur	Didukung oleh AWS Transfer Family
Support untuk sertifikat yang ditandatangani sendiri	Ya

1. Pesan Terjadwal Keluar tersedia dengan menjadwalkan AWS Lambda fungsi menggunakan Amazon EventBridge

AS2 mengirim dan menerima kemampuan

Tabel berikut menyediakan daftar kemampuan mengirim dan menerima AWS Transfer Family AS2.

Kemampuan	Inbound: Menerima dengan server	Keluar: Mengirim dengan konektor
Transportasi Terenkripsi TLS (HTTPS)	Ya [1]	Ya
Transportasi Non-TLS (HTTP)	Ya	Ya [2]
MDN sinkron	Ya	Ya
Kompresi Pesan	Ya	Ya
MDN asinkron	Ya	Tidak
Alamat IP Statis	Ya	Ya
Bawa Alamat IP Anda Sendiri	Ya	Tidak
Beberapa Lampiran File	Tidak	Tidak
Otentikasi Dasar	Tidak	Ya
AS2 Mulai Ulang	Tidak berlaku	Tidak
Subjek Kustom per Pesan	Tidak berlaku	Tidak

1. Transportasi Terenkripsi TLS Masuk tersedia dengan Network Load Balancer (NLB)

2. Transportasi Non-TLS Outbound hanya tersedia saat enkripsi diaktifkan

Konfigurasikan konektor AS2

Tujuan dari konektor adalah untuk membangun hubungan antara mitra dagang untuk transfer keluar — mengirim file AS2 dari server Transfer Family ke tujuan eksternal milik mitra. Untuk konektor, Anda menentukan pihak lokal, mitra jarak jauh, dan sertifikat mereka (dengan membuat profil lokal dan mitra).

Setelah Anda memiliki konektor, Anda dapat mentransfer informasi ke mitra dagang Anda. Setiap server AS2 diberi tiga alamat IP statis. Konektor AS2 menggunakan alamat IP ini untuk mengirim mDNS asinkron ke mitra dagang Anda melalui AS2.

> **Note**
>
> Ukuran pesan yang diterima oleh mitra dagang tidak akan cocok dengan ukuran objek di Amazon S3. Perbedaan ini terjadi karena pesan AS2 membungkus file dalam amplop sebelum dikirim. Jadi, ukuran file mungkin meningkat, bahkan jika file dikirim dengan kompresi. Oleh karena itu, pastikan bahwa ukuran file maksimum mitra dagang lebih besar dari ukuran file yang Anda kirim.

Buat konektor AS2

Prosedur ini menjelaskan cara membuat konektor AS2 dengan menggunakan AWS Transfer Family konsol. Jika Anda ingin menggunakan AWS CLI sebagai gantinya, lihatthe section called “Langkah 6: Buat konektor antara Anda dan pasangan”.

Untuk membuat konektor AS2

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi kiri, pilih Konektor, lalu pilih Buat konektor.
3. Di bagian Konfigurasi konektor, tentukan informasi berikut:
   - URL — Masukkan URL untuk koneksi keluar.
   - Peran akses — Pilih Nama Sumber Daya Amazon (ARN) dari peran AWS Identity and Access Management (IAM) yang akan digunakan. Pastikan bahwa peran ini menyediakan akses

baca dan tulis ke direktori induk dari lokasi file yang digunakan dalam `StartFileTransfer` permintaan. Selain itu, pastikan bahwa peran tersebut menyediakan akses baca dan tulis ke direktori induk dari file yang ingin Anda kirim`StartFileTransfer`.

> **Note**
>
> Jika Anda menggunakan otentikasi Dasar untuk konektor Anda, peran akses memerlukan `secretsmanager:GetSecretValue` izin untuk rahasia tersebut. Jika rahasia dienkripsi dengan menggunakan kunci yang dikelola pelanggan alih-alih Kunci yang dikelola AWS masuk AWS Secrets Manager, maka peran tersebut juga memerlukan `kms:Decrypt` izin untuk kunci itu. Jika Anda memberi nama rahasia Anda dengan awalan`aws/transfer/`, Anda dapat menambahkan izin yang diperlukan dengan karakter wildcard (*), seperti yang ditunjukkan dalam izin Contoh untuk membuat rahasia.

- Peran logging (opsional) - Pilih peran IAM untuk konektor yang akan digunakan untuk mendorong peristiwa ke CloudWatch log Anda.

4. Di bagian konfigurasi AS2, pilih profil lokal dan mitra, algoritma enkripsi dan penandatanganan, dan apakah akan mengompres informasi yang ditransfer. Perhatikan hal berikut:

   - Untuk algoritma enkripsi, jangan memilih `DES_EDE3_CBC` kecuali Anda harus mendukung klien lama yang membutuhkannya, karena ini adalah algoritma enkripsi yang lemah.
   - Subjek digunakan sebagai atribut header `subject` HTTP dalam pesan AS2 yang sedang dikirim dengan konektor.
   - Jika Anda memilih untuk membuat konektor tanpa algoritma enkripsi, Anda harus menentukan `HTTPS` sebagai protokol Anda.

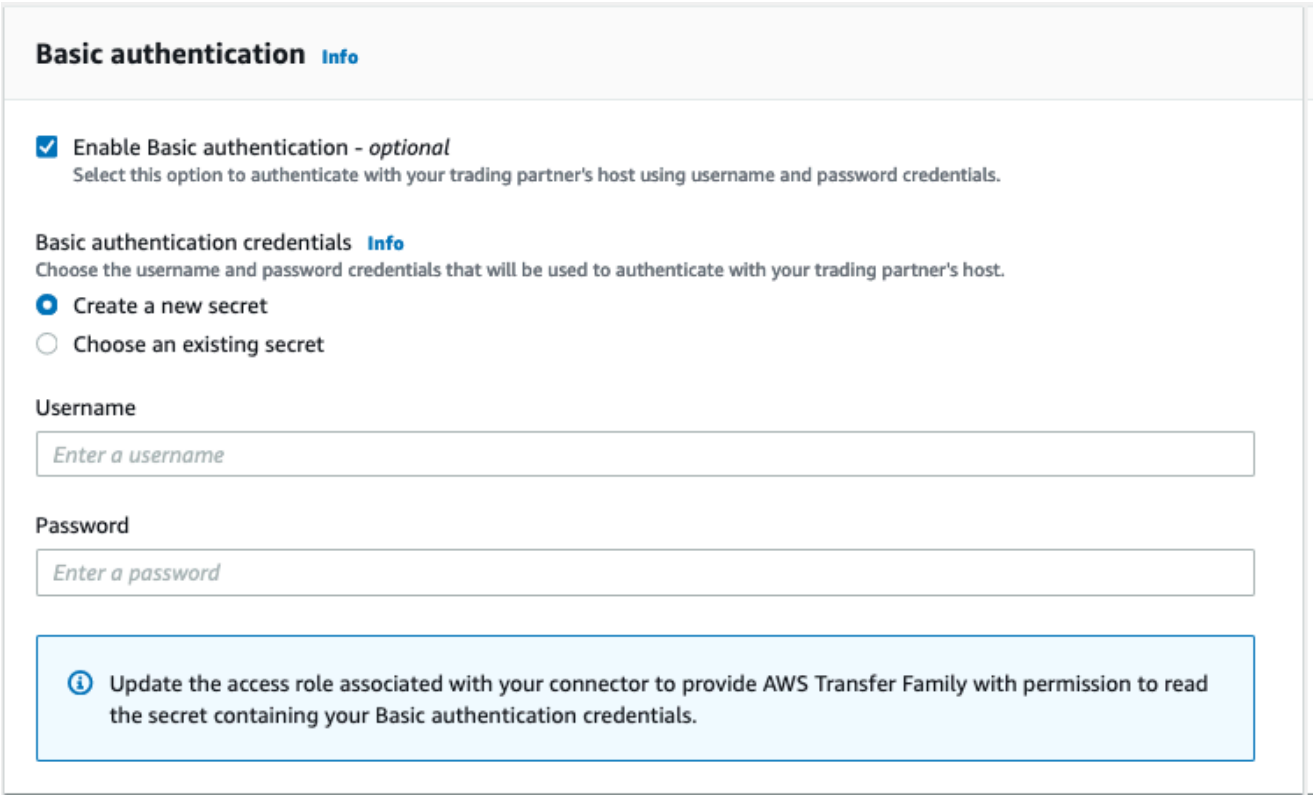
5. Di bagian konfigurasi MDN, tentukan informasi berikut:

   - Minta MDN — Anda memiliki opsi untuk meminta mitra dagang Anda mengirimi Anda MDN setelah mereka berhasil menerima pesan Anda melalui AS2.
   - Ditandatangani MDN — Anda memiliki opsi untuk meminta mDNS ditandatangani. Opsi ini hanya tersedia jika Anda telah memilih Minta MDN.

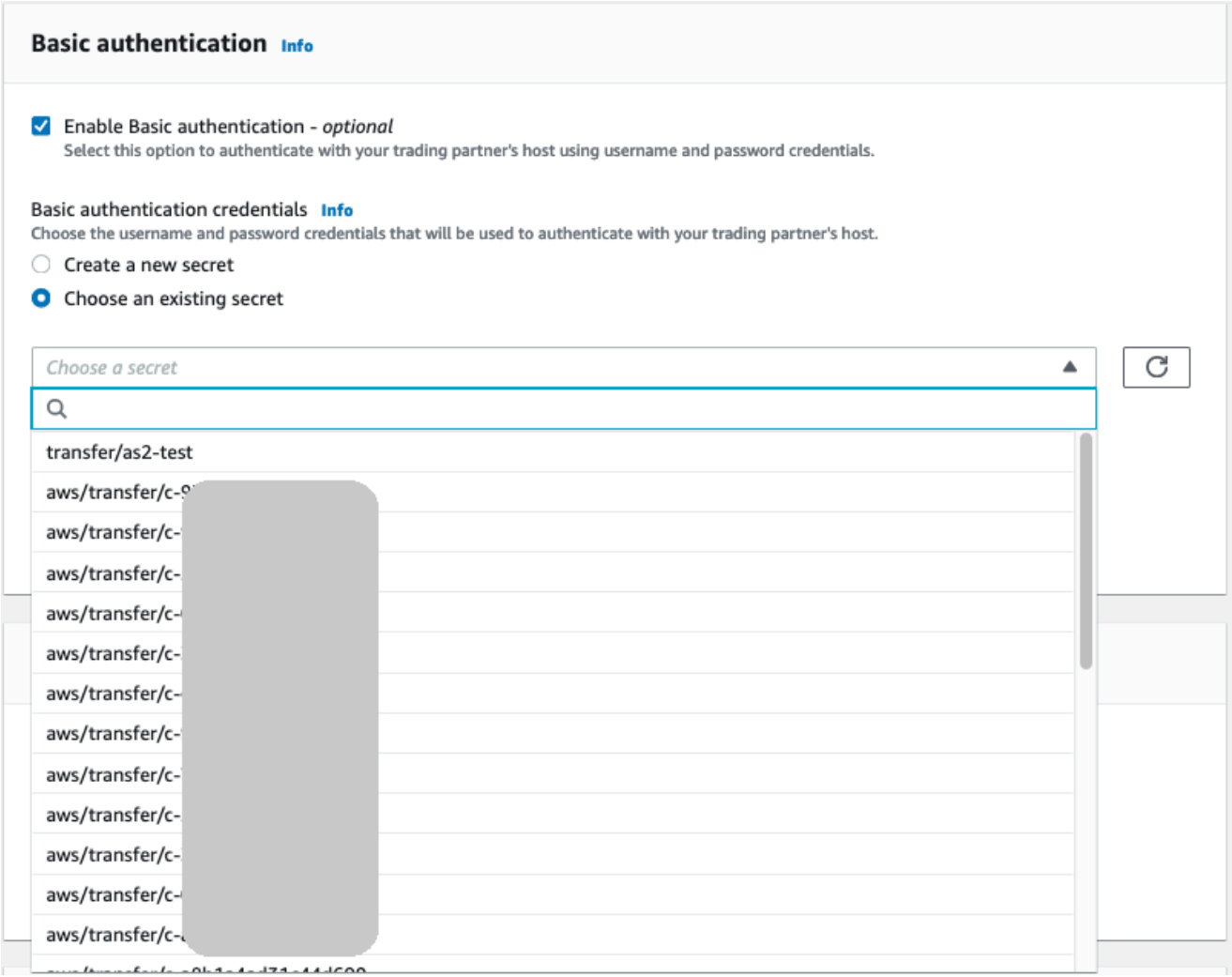
6. Di bagian otentikasi dasar, tentukan informasi berikut.

   - Untuk mengirim kredensyal masuk bersama dengan pesan keluar, pilih Aktifkan otentikasi Dasar. Jika Anda tidak ingin mengirim kredensyal apa pun dengan pesan keluar, pastikan Aktifkan otentikasi Dasar tetap bersih.

- Jika Anda menggunakan otentikasi, pilih atau buat rahasia.
  - Untuk membuat rahasia baru, pilih Buat rahasia baru lalu masukkan nama pengguna dan kata sandi. Kredensyal ini harus sesuai dengan pengguna yang terhubung ke titik akhir mitra.



  - Untuk menggunakan rahasia yang ada, pilih Pilih rahasia yang ada, lalu pilih rahasia dari menu tarik-turun. Untuk detail tentang membuat rahasia yang diformat dengan benar di Secrets Manager, lihat[Aktifkan otentikasi dasar untuk konektor AS2](#).



7. Setelah Anda mengonfirmasi semua pengaturan Anda, pilih Buat konektor untuk membuat konektor.

Halaman Konektor muncul, dengan ID konektor baru Anda ditambahkan ke daftar. Untuk melihat detail konektor Anda, lihatLihat detail konektor AS2.

Algoritma konektor AS2

Saat Anda membuat konektor AS2, algoritma keamanan berikut terpasang ke konektor.

Tipe	Algoritme
Cipher TLS	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256

Tipe	Algoritme
	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

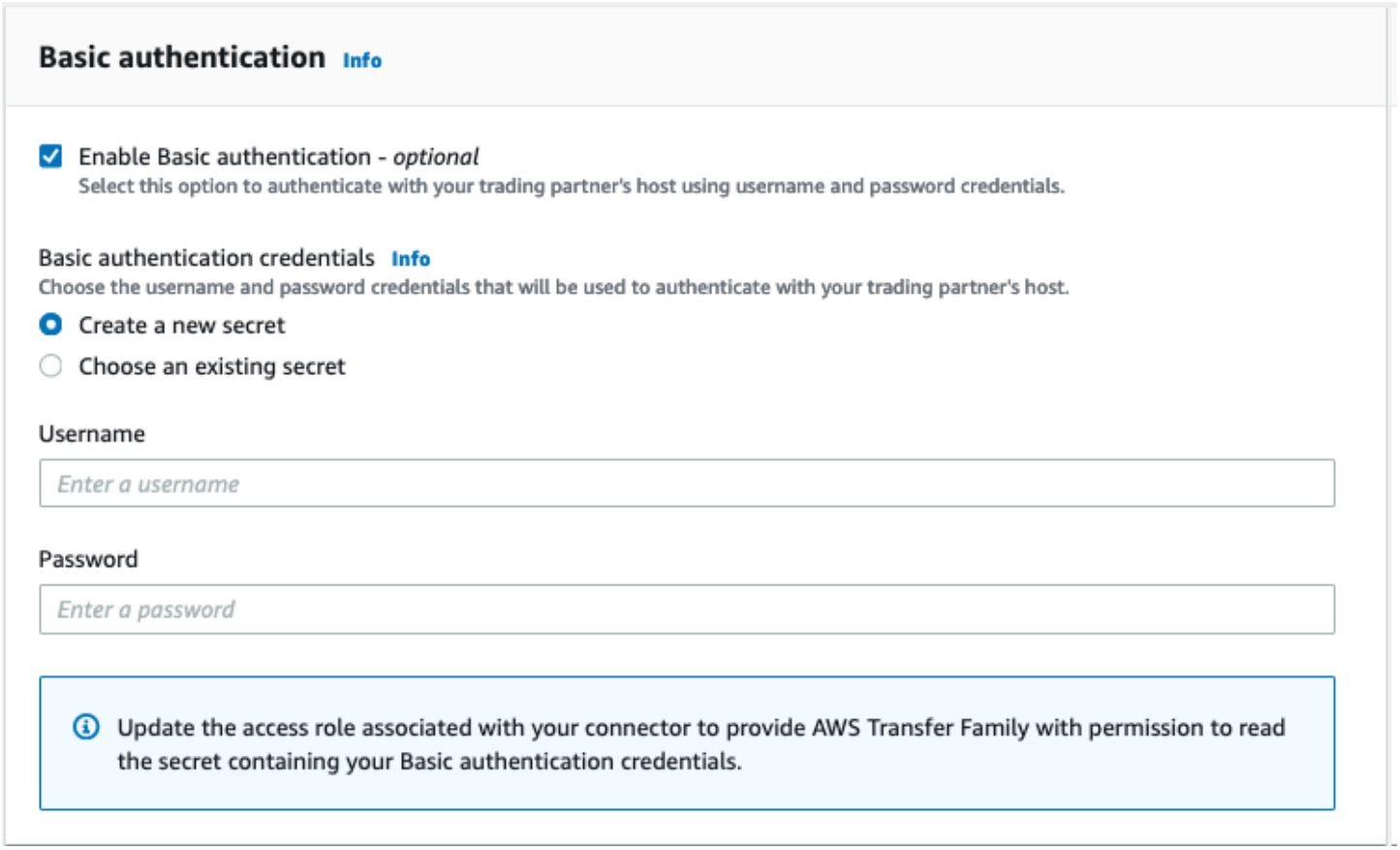
Otentikasi dasar untuk konektor AS2

Saat membuat atau memperbarui server Transfer Family yang menggunakan protokol AS2, Anda dapat menambahkan otentikasi Dasar untuk pesan keluar. Anda melakukan ini dengan menambahkan informasi otentikasi ke konektor.

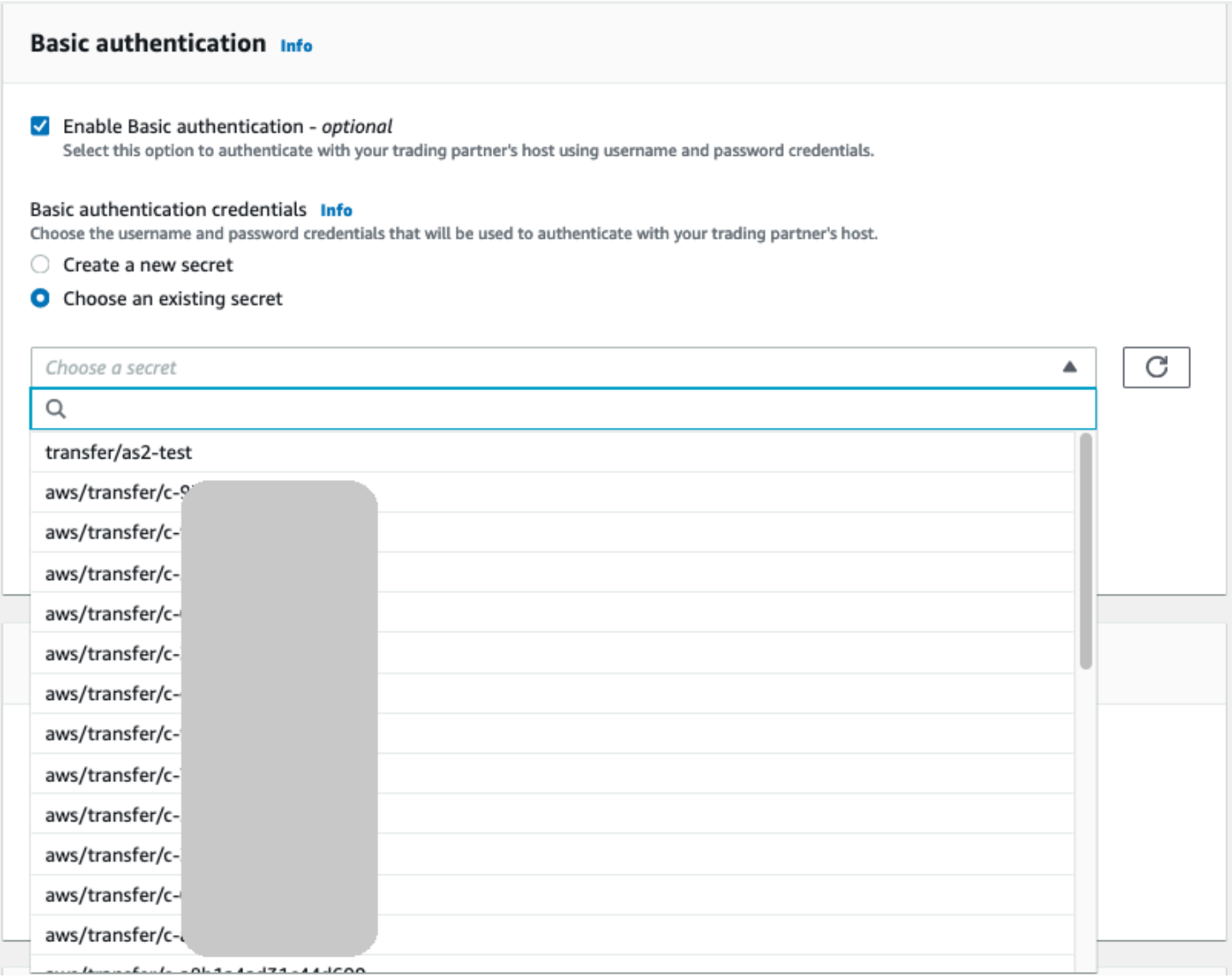
> **Note**
>
> Otentikasi dasar hanya tersedia jika Anda menggunakan HTTPS.

Untuk menggunakan otentikasi untuk konektor Anda, pilih Aktifkan otentikasi Dasar di bagian Autentikasi dasar. Setelah mengaktifkan otentikasi Dasar, Anda dapat memilih untuk membuat rahasia baru, atau menggunakan yang sudah ada. Dalam kedua kasus tersebut, kredensyal dalam rahasia dikirim dengan pesan keluar yang menggunakan konektor ini. Kredensyal harus sesuai dengan pengguna yang mencoba terhubung ke titik akhir jarak jauh mitra dagang.

Tangkapan layar berikut menunjukkan Aktifkan otentikasi Dasar yang dipilih, dan Buat rahasia baru yang dipilih. Setelah membuat pilihan ini, Anda dapat memasukkan nama pengguna dan kata sandi untuk rahasianya.



Tangkapan layar berikut menunjukkan Aktifkan otentikasi dasar dipilih, dan Pilih rahasia yang ada dipilih. Rahasia Anda harus dalam format yang benar, seperti yang dijelaskan dalamAktifkan otentikasi dasar untuk konektor AS2.



Aktifkan otentikasi dasar untuk konektor AS2

Saat Anda mengaktifkan otentikasi Dasar untuk konektor AS2, Anda dapat membuat rahasia baru di konsol Transfer Family, atau Anda dapat menggunakan rahasia yang Anda buat. AWS Secrets Manager Dalam kedua kasus tersebut, rahasia Anda disimpan di Secrets Manager.

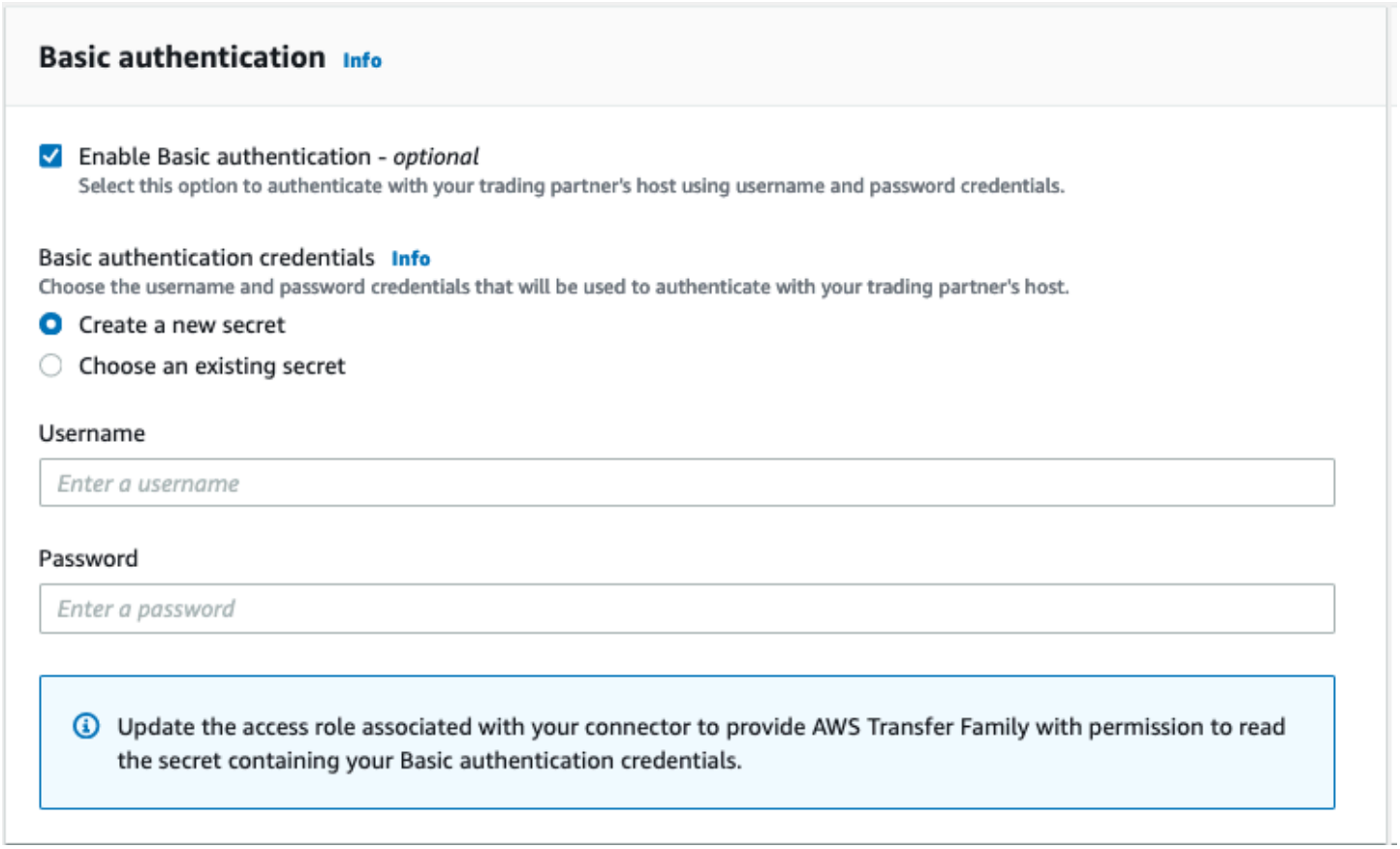
Topik

- Buat rahasia baru di konsol
- Gunakan secret yang sudah ada
- Buat rahasia di AWS Secrets Manager

Buat rahasia baru di konsol

Saat Anda membuat konektor di konsol, Anda dapat membuat rahasia baru.

Untuk membuat rahasia baru, pilih Buat rahasia baru lalu masukkan nama pengguna dan kata sandi. Kredensyal ini harus sesuai dengan pengguna yang terhubung ke titik akhir mitra.

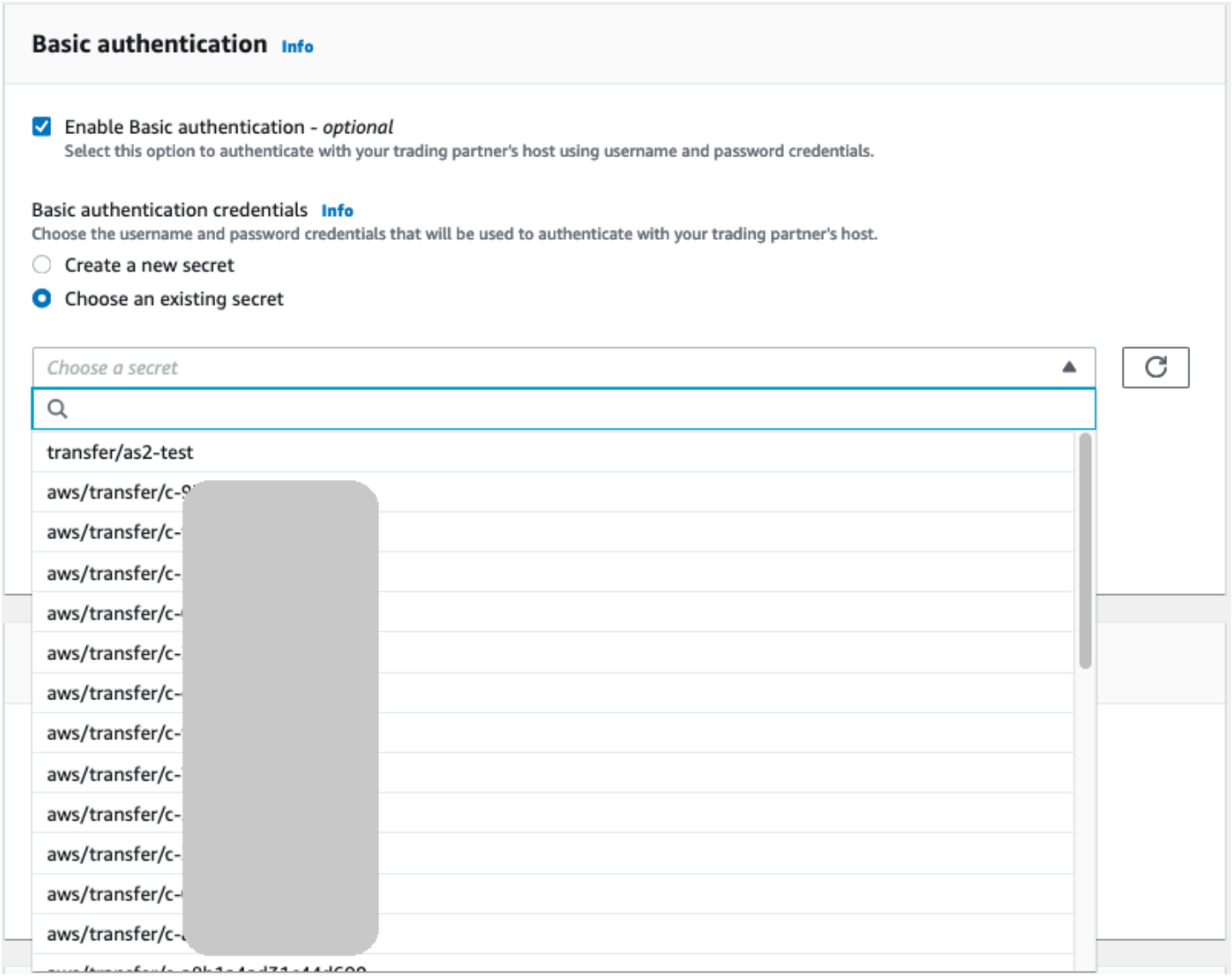


> **Note**
>
> Saat Anda membuat rahasia baru di konsol, nama rahasia mengikuti konvensi penamaan ini:**`/aws/transfer/`*`connector-id`***, di mana *`connector-id`* adalah ID konektor yang Anda buat. Pertimbangkan ini ketika Anda mencoba menemukan rahasianya AWS Secrets Manager.

Gunakan secret yang sudah ada

Saat membuat konektor di konsol, Anda dapat menentukan rahasia yang ada.

Untuk menggunakan rahasia yang ada, pilih Pilih rahasia yang ada, lalu pilih rahasia dari menu tarik-turun. Untuk detail tentang membuat rahasia yang diformat dengan benar di Secrets Manager, lihatBuat rahasia di AWS Secrets Manager.



Buat rahasia di AWS Secrets Manager

Prosedur berikut menjelaskan cara membuat rahasia yang sesuai untuk digunakan dengan konektor AS2 Anda.

> **Note**
>
> Otentikasi dasar hanya tersedia jika Anda menggunakan HTTPS.

Untuk menyimpan kredensyal pengguna di Secrets Manager untuk otentikasi AS2 Basic

1. Masuk ke AWS Management Console dan buka AWS Secrets Manager konsol di https://console.aws.amazon.com/secretsmanager/.
2. Pada panel navigasi kiri, pilih Rahasia.
3. Pada halaman Rahasia, pilih Simpan rahasia baru.
4. Pada halaman Pilih jenis rahasia, untuk tipe Rahasia, pilih Jenis rahasia lainnya.
5. Di bagian pasangan kunci/Nilai, pilih tab kunci/Nilai.
   - Kunci — Masukkan**`Username`**.
   - nilai — Masukkan nama pengguna yang berwenang untuk terhubung ke server mitra.
6. Jika Anda ingin memberikan kata sandi, pilih Tambahkan baris, dan di bagian pasangan kunci/Nilai, pilih tab kunci/Nilai.

   Pilih Tambah baris, dan di bagian pasangan kunci/Nilai, pilih tab kunci/Nilai.
   - Kunci — Masukkan**`Password`**.
   - nilai — Masukkan kata sandi untuk pengguna.
7. Jika Anda ingin memberikan kunci pribadi, pilih Tambah baris, dan di bagian pasangan kunci/nilai, pilih tab kunci/Nilai.
   - Kunci — Masukkan**`PrivateKey`**.
   - nilai — Masukkan kunci pribadi untuk pengguna. Nilai ini harus disimpan dalam format OpenSSH, dan harus sesuai dengan kunci publik yang disimpan untuk pengguna ini di server jarak jauh.
8. Pilih Berikutnya.
9. Pada halaman Konfigurasi rahasia, masukkan nama dan deskripsi untuk rahasia Anda. Kami menyarankan Anda menggunakan awalan **`aws/transfer/`** untuk nama tersebut. Misalnya, Anda bisa menyebutkan rahasia Anda**`aws/transfer/connector-1`**.
10. Pilih Berikutnya, dan kemudian terima default pada halaman Konfigurasi rotasi. Lalu pilih Selanjutnya.
11. Pada halaman Review, pilih Store untuk membuat dan menyimpan rahasia.

Setelah Anda membuat rahasia, Anda dapat memilihnya saat Anda membuat konektor (lihatKonfigurasikan konektor AS2). Pada langkah di mana Anda mengaktifkan otentikasi Dasar, pilih rahasia dari daftar dropdown rahasia yang tersedia.

Lihat detail konektor AS2

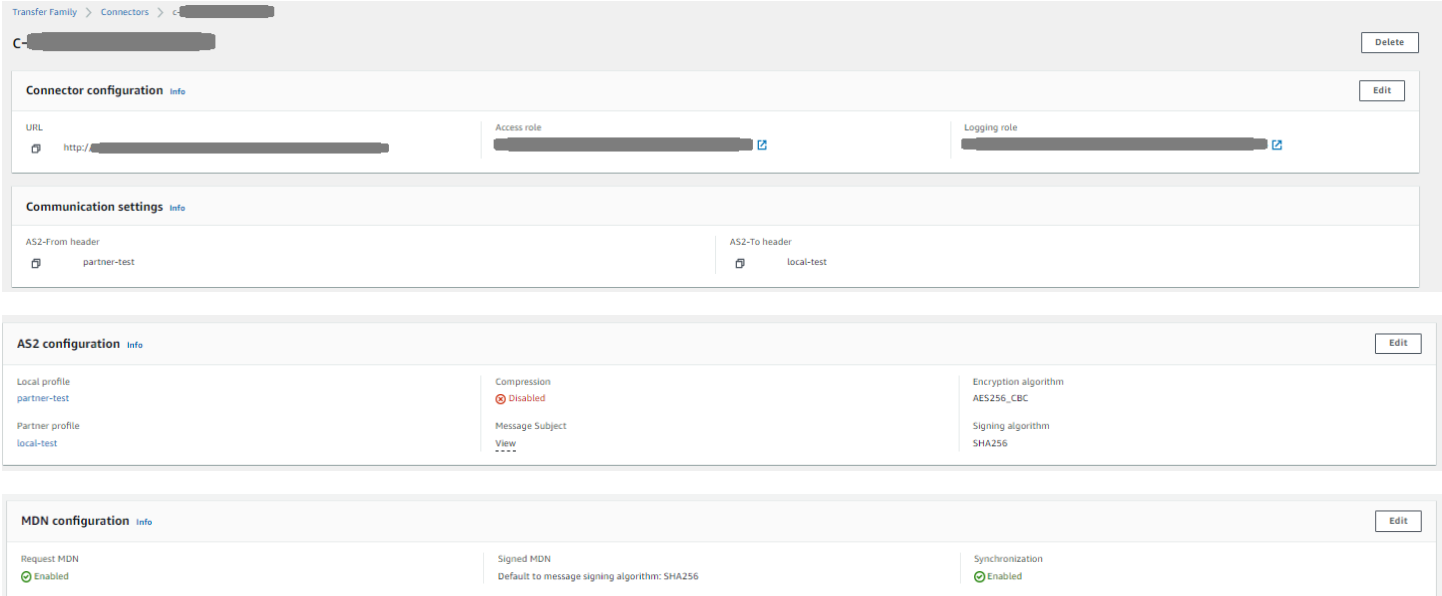
Anda dapat menemukan daftar detail dan properti untuk AWS Transfer Family konektor AS2 di AWS Transfer Family konsol. Properti konektor AS2 mencakup URL, peran, profil, mDNS, tag, dan metrik pemantauannya.

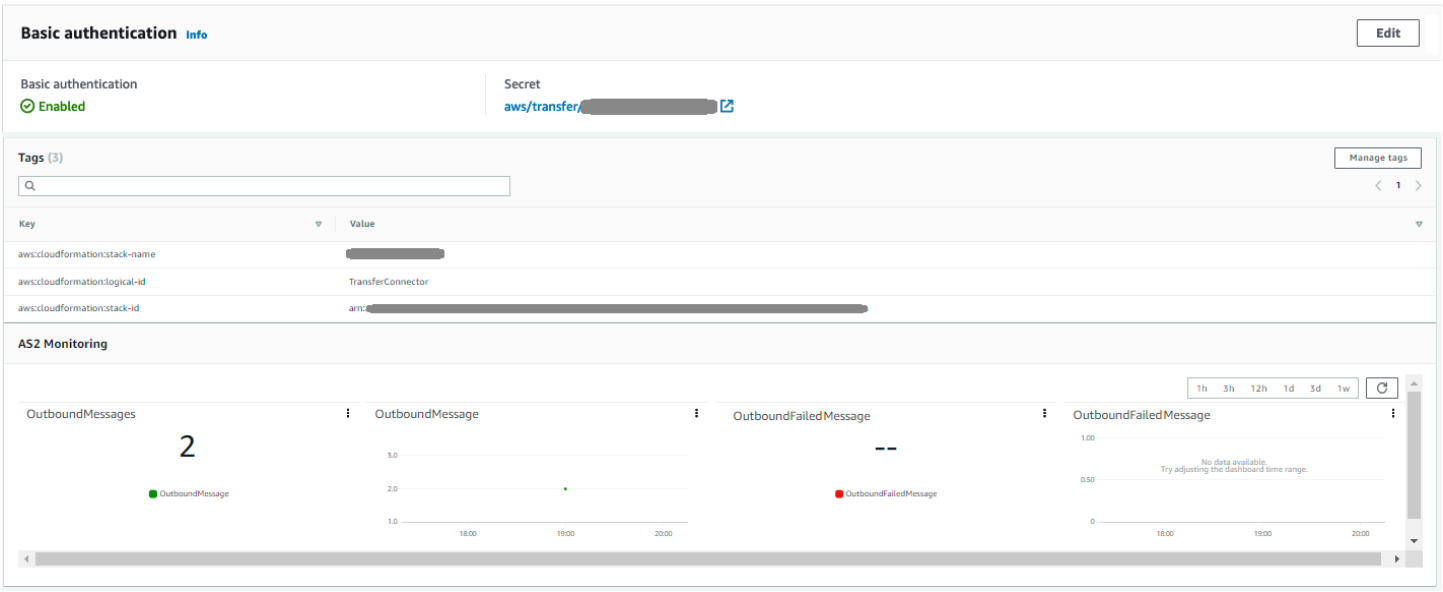
Ini adalah prosedur untuk melihat detail konektor.

Untuk melihat detail konektor

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi kiri, pilih Konektor.
3. Pilih pengenal di kolom Connector ID untuk melihat halaman detail untuk konektor yang dipilih.

Anda dapat mengubah properti untuk konektor AS2 pada halaman detail konektor dengan memilih Edit.





> **Note**
>
> Anda bisa mendapatkan banyak informasi ini, meskipun dalam format yang berbeda, dengan menjalankan perintah berikut AWS Command Line Interface (AWS CLI :
>
> ```
> aws transfer describe-connector --connector-id your-connector-id
> ```
>
> Untuk informasi selengkapnya, lihat DescribeConnectordi referensi API.

Kelola mitra AS2

Topik ini membahas cara mengelola sertifikat, profil, dan perjanjian AS2.

Impor sertifikat AS2

Proses Transfer Family AS2 menggunakan kunci sertifikat untuk enkripsi dan penandatanganan informasi yang ditransfer. Mitra dapat menggunakan kunci yang sama untuk kedua tujuan, atau kunci terpisah untuk masing-masing. Jika Anda memiliki kunci enkripsi umum yang disimpan di escrow oleh pihak ketiga tepercaya sehingga data dapat didekripsi jika terjadi bencana atau pelanggaran keamanan, sebaiknya Anda memiliki kunci penandatanganan terpisah. Dengan menggunakan kunci penandatanganan terpisah (yang tidak Anda escrow), Anda tidak mengkompromikan fitur non-penolakan tanda tangan digital Anda.

> **Note**
>
> Panjang kunci untuk sertifikat AS2 harus minimal 2048 bit, dan paling banyak 4096.

Poin-poin berikut merinci bagaimana sertifikat AS2 digunakan selama proses berlangsung.

- AS2 masuk
  - Mitra dagang mengirimkan kunci publik mereka untuk sertifikat penandatanganan, dan kunci ini diimpor ke profil mitra.
  - Pihak lokal mengirimkan kunci publik untuk enkripsi dan sertifikat penandatanganan mereka. Mitra kemudian mengimpor kunci pribadi atau kunci. Pihak lokal dapat mengirim kunci sertifikat terpisah untuk penandatanganan dan enkripsi, atau dapat memilih untuk menggunakan kunci yang sama untuk kedua tujuan tersebut.
- AS2 Keluar
  - Mitra mengirimkan kunci publik untuk sertifikat enkripsi mereka, dan kunci ini diimpor ke profil mitra.
  - Pihak lokal mengirimkan kunci publik untuk sertifikat untuk ditandatangani, dan mengimpor kunci pribadi sertifikat untuk ditandatangani.
  - Jika Anda menggunakan HTTPS, Anda dapat mengimpor sertifikat Transport Layer Security (TLS) yang ditandatangani sendiri.

Untuk detail tentang cara membuat sertifikat, lihatthe section called “Langkah 1: Buat sertifikat untuk AS2”.

Prosedur ini menjelaskan cara mengimpor sertifikat dengan menggunakan konsol Transfer Family. Jika Anda ingin menggunakan AWS CLI sebagai gantinya, lihatthe section called “Langkah 3: Impor sertifikat sebagai sumber sertifikat Transfer Family”.

Untuk menentukan sertifikat AS2 diaktifkan

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi kiri, di bawah Mitra Dagang AS2, pilih Sertifikat.
3. Pilih Impor sertifikat.

4. Di bagian Deskripsi sertifikat, masukkan nama yang mudah diidentifikasi untuk sertifikat. Pastikan Anda dapat mengidentifikasi tujuan sertifikat dengan deskripsinya. Selain itu, pilih peran untuk sertifikat.
5. Di bagian Konten Sertifikat, berikan sertifikat publik dari mitra dagang, atau kunci publik dan pribadi untuk sertifikat lokal.
6. Di bagian Penggunaan sertifikat, pilih tujuan untuk sertifikat ini. Ini dapat digunakan untuk enkripsi, penandatanganan, atau keduanya.

   > **Note**
   >
   > Jika Anda memilih Enkripsi dan menandatangani untuk penggunaan, Transfer Family membuat dua sertifikat identik (masing-masing memiliki ID sendiri): satu dengan nilai penggunaan `ENCRYPTION` dan satu dengan nilai penggunaan`SIGNING`.

7. Isi bagian Isi sertifikat dengan detail yang sesuai.

   - Jika Anda memilih Sertifikat yang ditandatangani sendiri, Anda tidak memberikan rantai sertifikat.
   - Tempel di isi sertifikat.
   - Jika sertifikat bukan sertifikat yang ditandatangani sendiri, berikan rantai sertifikat.
   - Jika sertifikat ini adalah sertifikat lokal, tempel di kunci pribadinya.

8. Pilih Impor sertifikat untuk menyelesaikan proses dan menyimpan rincian untuk sertifikat yang diimpor.

> **Note**
>
> Sertifikat TLS hanya dapat diimpor sebagai sertifikat publik mitra. Jika Anda memilih sertifikat Publik dari mitra, lalu pilih Transport Layer Security (TLS) untuk penggunaan, Anda akan menerima peringatan. Selain itu, sertifikat TLS harus ditandatangani sendiri (yaitu, Anda harus memilih Self Signed Certificate untuk mengimpor sertifikat TLS).

Rotasi sertifikat AS2

Seringkali, sertifikat berlaku untuk jangka waktu enam bulan hingga satu tahun. Anda mungkin telah menyiapkan profil yang ingin Anda pertahankan untuk durasi yang lebih lama. Untuk memfasilitasi

hal ini, Transfer Family menyediakan rotasi sertifikat. Anda dapat menentukan beberapa sertifikat untuk profil, memungkinkan Anda untuk tetap menggunakan profil selama beberapa tahun. Transfer Family menggunakan sertifikat untuk penandatanganan (opsional) dan enkripsi (wajib). Anda dapat menentukan satu sertifikat untuk kedua tujuan, jika Anda mau.

Rotasi sertifikat adalah proses penggantian sertifikat lama yang kedaluwarsa dengan sertifikat yang lebih baru. Transisi adalah transisi bertahap untuk menghindari gangguan transfer di mana mitra dalam perjanjian belum mengonfigurasi sertifikat baru untuk transfer keluar atau mungkin mengirim muatan yang ditandatangani atau dienkripsi dengan sertifikat lama selama periode ketika sertifikat yang lebih baru mungkin juga digunakan. Periode menengah di mana sertifikat lama dan baru valid disebut sebagai masa tenggang.

Sertifikat X.509 memiliki `Not Before` dan tanggal. `Not After` Namun, parameter ini mungkin tidak memberikan kontrol yang cukup untuk administrator. Transfer Family `Inactive Date` menyediakan `Active Date` dan mengatur untuk mengontrol sertifikat mana yang digunakan untuk muatan keluar dan mana yang diterima untuk muatan masuk.

Pemilihan sertifikat keluar menggunakan nilai maksimum yang sebelum tanggal transfer sebagai`Inactive Date`. Proses masuk menerima sertifikat dalam kisaran `Not Before` dan `Not After` dan dalam kisaran `Active Date` dan`Inactive Date`.

Tabel berikut menjelaskan satu cara yang mungkin untuk mengkonfigurasi dua sertifikat untuk satu profil.

Dua sertifikat dalam rotasi

Nama	NOT BEFORE(dikendalikan oleh otoritas sertifikat)	ACTIVE DATE(diatur oleh Transfer Family)	INACTIVE DATE(diatur oleh Transfer Family)	NOT AFTER(ditetapkan oleh otoritas sertifikat)
Cert1 (sertifikat lama)	2019-11-01	2020-01-01	2020-12-31	2024-01-01
Cert2 (sertifikat yang lebih baru)	01/11/2020	2020-06-01	2021-06-01	2025-01-01

Perhatikan hal berikut:

- Saat Anda menentukan `Active Date` dan `Inactive Date` untuk sertifikat, rentang harus berada di dalam rentang antara `Not Before` dan`Not After`.
- Kami menyarankan Anda mengonfigurasi beberapa sertifikat untuk setiap profil, memastikan bahwa rentang tanggal aktif untuk semua sertifikat yang digabungkan mencakup jumlah waktu yang ingin Anda gunakan profil.
- Kami menyarankan Anda menentukan waktu tenggang antara saat sertifikat lama Anda menjadi tidak aktif dan ketika sertifikat Anda yang lebih baru menjadi aktif. Dalam contoh sebelumnya, sertifikat pertama tidak menjadi tidak aktif hingga 2020-12-31, sedangkan sertifikat kedua menjadi aktif pada 2020-06-01, memberikan masa tenggang 6 bulan. Selama periode 2020-06-01 hingga 2020-12-31, kedua sertifikat aktif.

Buat profil AS2

Gunakan prosedur ini untuk membuat profil lokal dan mitra. Prosedur ini menjelaskan cara membuat profil AS2 dengan menggunakan konsol Transfer Family. Jika Anda ingin menggunakan AWS CLI sebagai gantinya, lihatthe section called “Langkah 4: Buat profil untuk Anda dan mitra dagang Anda”.

Untuk membuat profil AS2

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi kiri, di bawah AS2 Trading Partners, pilih Profil, lalu pilih Buat profil.
3. Di bagian Konfigurasi profil, masukkan ID AS2 untuk profil. Nilai ini digunakan untuk header HTTP khusus protokol AS2 `as2-from` dan `as2-to` untuk mengidentifikasi kemitraan perdagangan, yang menentukan sertifikat yang akan digunakan, dan sebagainya.
4. Di bagian Jenis profil, pilih Profil lokal atau Profil mitra.
5. Di bagian Sertifikat, pilih satu atau beberapa sertifikat dari menu tarik-turun.

   > **Note**
   >
   > Jika Anda ingin mengimpor sertifikat yang tidak tercantum dalam menu tarik-turun, pilih Impor Sertifikat baru. Ini membuka jendela browser baru di layar Impor sertifikat. Untuk prosedur tentang mengimpor sertifikat lihatImpor sertifikat AS2.

6. (Opsional) Di tag bagian, tentukan satu atau beberapa pasangan nilai kunci untuk membantu mengidentifikasi profil ini.
7. Pilih Buat profil untuk menyelesaikan proses dan menyimpan profil baru.

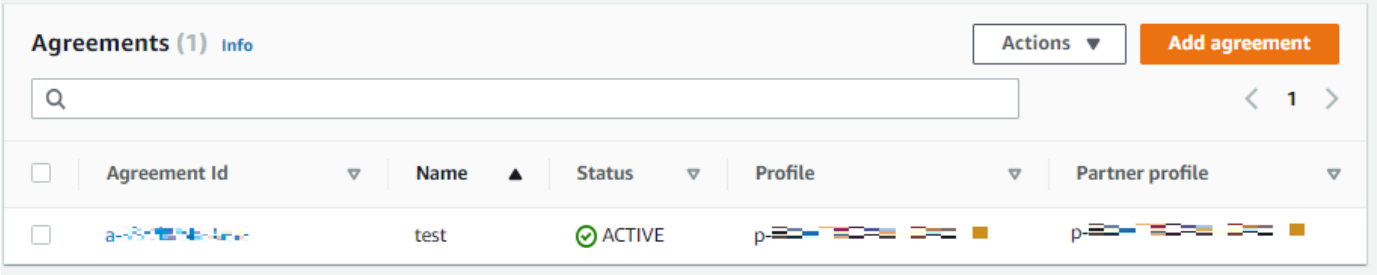
Buat perjanjian AS2

Perjanjian terkait dengan server Transfer Family. Mereka menentukan detail untuk mitra dagang yang menggunakan protokol AS2 untuk bertukar pesan atau file dengan menggunakan Transfer Family, untuk transfer masuk—mengirim file AS2 dari sumber eksternal milik mitra ke server Transfer Family.

Prosedur ini menjelaskan cara membuat perjanjian AS2 dengan menggunakan konsol Transfer Family. Jika Anda ingin menggunakan AWS CLI sebagai gantinya, lihatthe section called “Langkah 5: Buat kesepakatan antara Anda dan pasangan”.

Untuk membuat perjanjian untuk server Transfer Family

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi kiri, pilih Server, lalu pilih server yang menggunakan protokol AS2.
3. Pada halaman detail server, gulir ke bawah ke bagian Perjanjian.



4. Pilih Tambahkan perjanjian.
5. Isi parameter perjanjian, sebagai berikut:

   a. Di bagian Konfigurasi Perjanjian, masukkan nama deskriptif. Pastikan Anda dapat mengidentifikasi tujuan perjanjian dengan namanya. Juga, tetapkan Status untuk perjanjian: Aktif (dipilih secara default) atau Tidak Aktif.

   b. Di bagian Konfigurasi komunikasi, pilih profil lokal dan profil mitra.

   c. Di bagian konfigurasi folder Kotak Masuk, pilih bucket Amazon S3 untuk menyimpan file masuk dan peran IAM yang dapat mengakses bucket. Secara opsional, Anda dapat memasukkan awalan (folder) yang akan digunakan untuk menyimpan file di ember.

      Misalnya, jika Anda memasukkan **`DOC-EXAMPLE-BUCKET`** bucket dan **`incoming`** awalan, file masuk Anda akan disimpan ke folder`/DOC-EXAMPLE-BUCKET/incoming`.

   d. (Opsional) Tambahkan tag di bagian Tag.

   e. Setelah Anda memasukkan semua informasi untuk perjanjian, pilih Buat perjanjian.

Perjanjian baru muncul di bagian Perjanjian pada halaman detail server.

Mengirim dan menerima pesan AS2

Bagian ini menjelaskan proses untuk mengirim dan menerima pesan AS2. Ini juga memberikan rincian tentang nama file dan lokasi yang terkait dengan pesan AS2.

Tabel berikut mencantumkan algoritma enkripsi yang tersedia untuk pesan AS2, dan kapan Anda dapat menggunakannya.

Enkripsi algoritme	HTTP	HTTPS	Catatan
AES128_CBC	Ya	Ya	
AES192_CBC	Ya	Ya	
AES256_CBC	Ya	Ya	
DES_EDE3_CBC	Ya	Ya	Hanya gunakan algoritma ini jika Anda harus mendukung klien lama yang membutuhkannya, karena ini adalah algoritma enkripsi yang lemah.
NONE	Tidak	Ya	Jika Anda mengirim pesan ke server Transfer Family, Anda hanya dapat memilih `NONE` apakah Anda menggunak an Application Load Balancer (ALB).

Topik

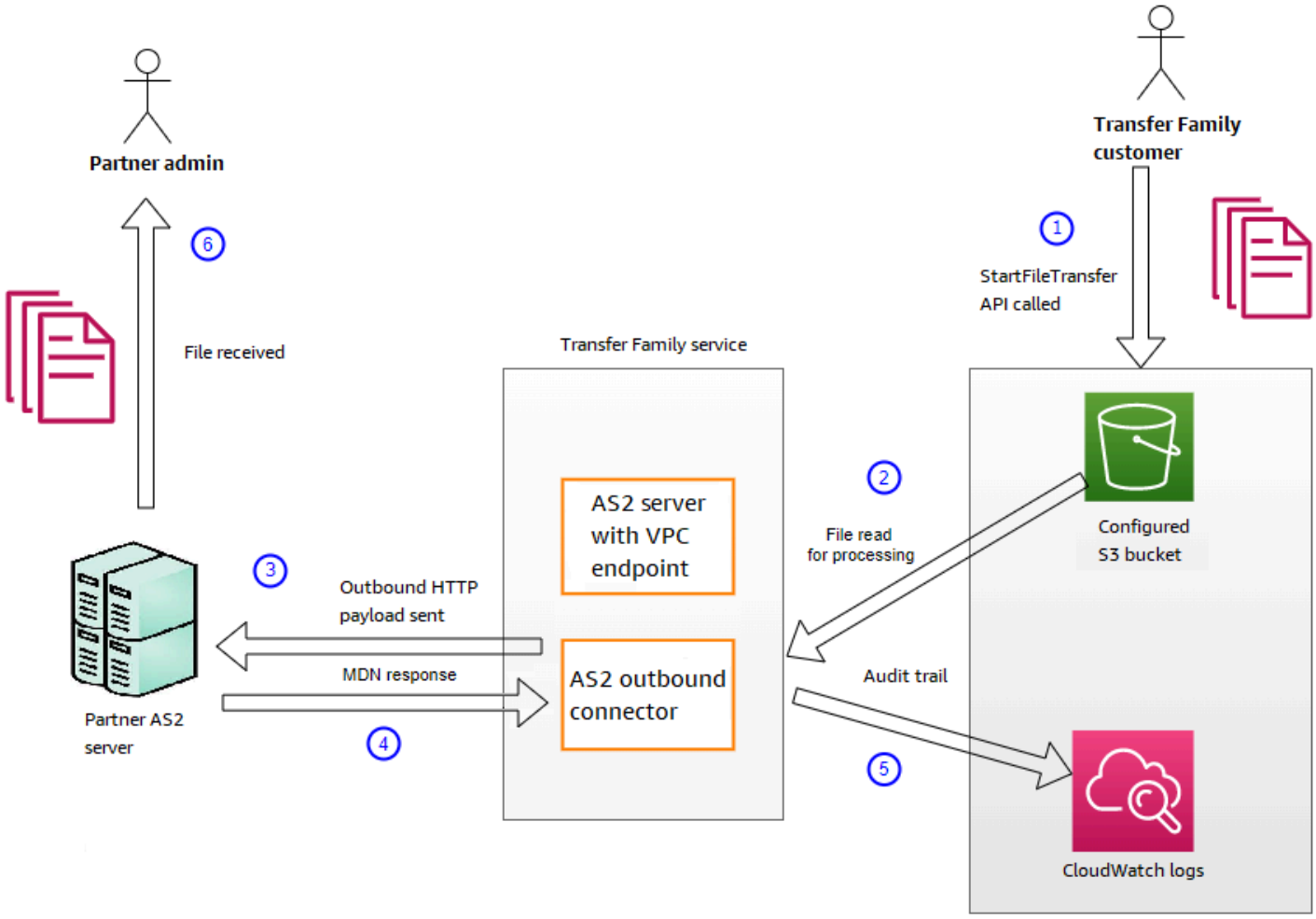
- Kirim proses pesan AS2

- Menerima proses pesan AS2
- Mengirim dan menerima pesan AS2 melalui HTTPS
- Mentransfer file dengan menggunakan konektor AS2
- Nama dan lokasi file
- Kode status
- Contoh file JSON

Kirim proses pesan AS2

Proses keluar didefinisikan sebagai pesan atau file yang dikirim dari AWS ke klien atau layanan eksternal. Urutan pesan keluar adalah sebagai berikut:

1. Admin memanggil perintah `start-file-transfer` AWS Command Line Interface (AWS CLI) atau operasi `StartFileTransfer` API. Operasi ini mereferensikan `connector` konfigurasi.
2. Transfer Family mendeteksi permintaan file baru dan menemukan file. File dikompresi, ditandatangani, dan dienkripsi.
3. Klien HTTP transfer melakukan permintaan HTTP POST untuk mengirimkan muatan ke server AS2 mitra.
4. Proses mengembalikan respons MDN yang ditandatangani, sejalan dengan respons HTTP (MDN sinkron).
5. Saat file bergerak di antara berbagai tahap transmisi, proses memberikan tanda terima respons MDN dan rincian pemrosesan kepada pelanggan.
6. Server AS2 jarak jauh membuat file yang didekripsi dan diverifikasi tersedia untuk admin mitra.



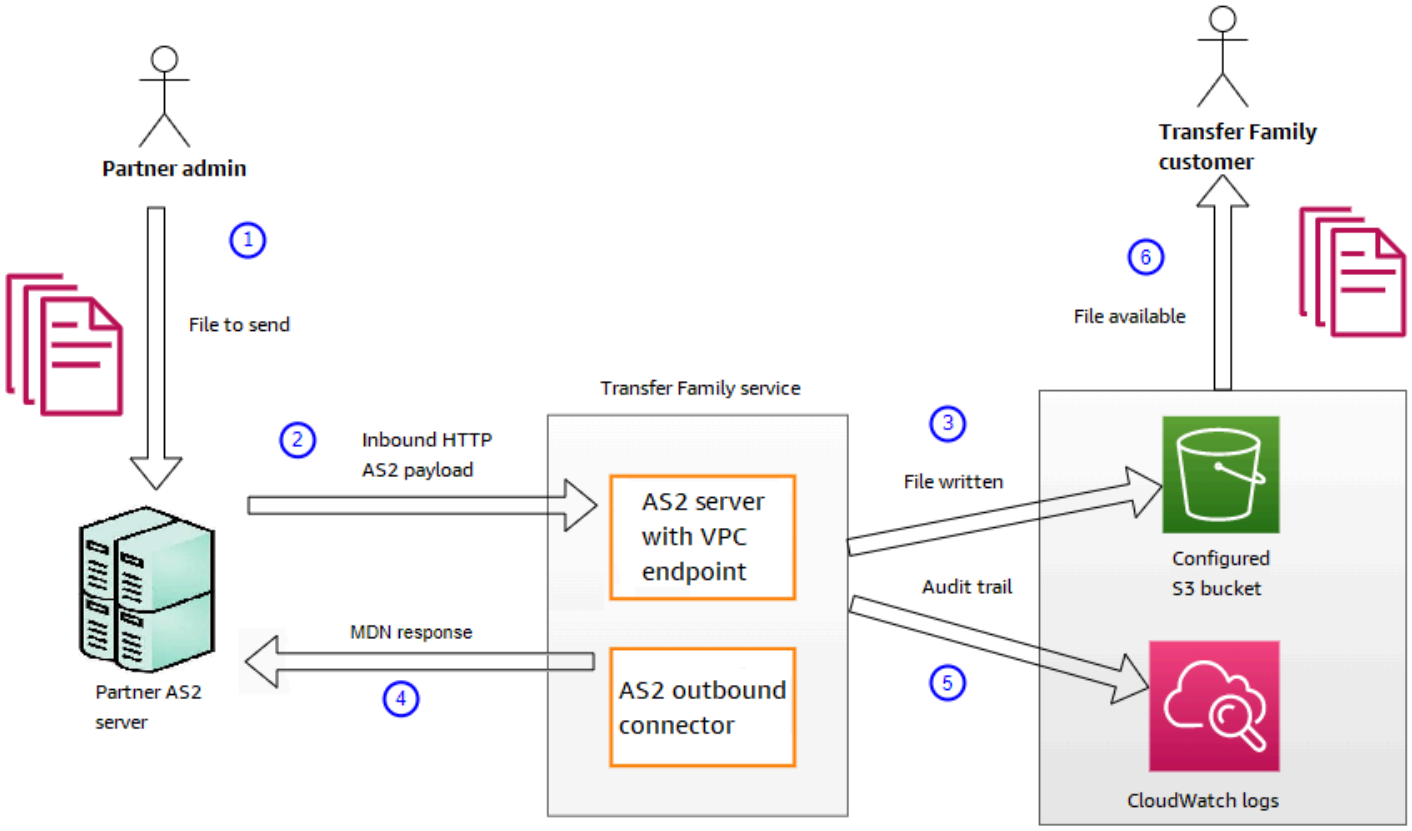
Pemrosesan AS2 mendukung banyak protokol RFC 4130, dengan fokus pada kasus penggunaan umum dan integrasi dengan implementasi server berkemampuan AS2 yang ada. Untuk detail konfigurasi yang didukung, lihatKonfigurasi yang didukung AS2.

Menerima proses pesan AS2

Proses inbound didefinisikan sebagai pesan atau file yang sedang ditransfer ke AWS Transfer Family server Anda. Urutan pesan masuk adalah sebagai berikut:

1. Admin atau proses otomatis memulai transfer file AS2 di server AS2 jarak jauh mitra.
2. Server AS2 jarak jauh mitra menandatangani dan mengenkripsi konten file, lalu mengirimkan permintaan HTTP POST ke titik akhir masuk AS2 yang dihosting di Transfer Family.
3. Menggunakan nilai yang dikonfigurasi untuk server, mitra, sertifikat, dan perjanjian, Transfer Family mendekripsi dan memverifikasi payload AS2. Isi file disimpan di toko file Amazon S3 yang dikonfigurasi.

4. Respons MDN yang ditandatangani dikembalikan baik sebaris dengan respons HTTP, atau secara asinkron melalui permintaan HTTP POST terpisah kembali ke server asal.
5. Jejak audit ditulis ke Amazon CloudWatch dengan rincian tentang pertukaran.
6. File yang didekripsi tersedia dalam folder bernama. `inbox/processed`



Mengirim dan menerima pesan AS2 melalui HTTPS

Bagian ini menjelaskan cara mengonfigurasi server Transfer Family yang menggunakan protokol AS2 untuk mengirim dan menerima pesan melalui HTTPS.

Kirim pesan AS2 melalui HTTPS

Untuk mengirim pesan AS2 menggunakan HTTPS, buat konektor dengan informasi berikut:

- Untuk URL, tentukan URL HTTPS
- Untuk algoritma enkripsi, pilih salah satu algoritma yang tersedia.

> **Note**
>
> Untuk mengirim pesan ke server Transfer Family saat tidak menggunakan enkripsi (yaitu, Anda memilih `NONE` algoritma enkripsi), Anda harus menggunakan Application Load Balancer (ALB).

- Berikan nilai yang tersisa untuk konektor seperti yang dijelaskan dalamKonfigurasikan konektor AS2.

Terima pesan AS2 melalui HTTPS

AWS Transfer Family Server AS2 saat ini hanya menyediakan transportasi HTTP melalui port 5080. Namun, Anda dapat menghentikan TLS pada penyeimbang beban di depan endpoint VPC server Transfer Family Anda dengan menggunakan port dan sertifikat pilihan Anda. Dengan pendekatan ini, Anda dapat memiliki pesan AS2 yang masuk menggunakan HTTPS.

Prasyarat

- VPC harus sama Wilayah AWS dengan server Transfer Family Anda.
- Subnet VPC Anda harus berada dalam Availability Zones tempat Anda ingin menggunakan server Anda.

> **Note**
>
> Setiap server Transfer Family dapat mendukung hingga tiga Availability Zone.

- Alokasikan hingga tiga alamat IP Elastis di Wilayah yang sama dengan server Anda. Atau, Anda dapat memilih untuk membawa rentang alamat IP Anda sendiri (BYOIP).

> **Note**
>
> Jumlah alamat IP Elastis harus sesuai dengan jumlah Availability Zone yang Anda gunakan dengan endpoint server Anda.

Konfigurasikan Network Load Balancer Anda

Siapkan Network Load Balancer (NLB) yang menghadap ke internet di VPC Anda.

Untuk membuat Network Load Balancer dan mendefinisikan titik akhir VPC server sebagai target penyeimbang beban

1. Buka konsol Amazon Elastic Compute Cloud di https://console.aws.amazon.com/ec2/.
2. Dari panel navigasi, pilih Load Balancers, lalu pilih Create load balancer.
3. Di bawah Penyeimbang Beban Jaringan, pilih Buat.
4. Di bagian Konfigurasi dasar, masukkan informasi berikut:
   - Untuk Nama, masukkan nama deskriptif untuk penyeimbang beban.
   - Untuk Skema, pilih Internet-facing.
   - Untuk jenis alamat IP, pilih IPv4.
5. Di bagian Pemetaan jaringan, masukkan informasi berikut:
   - Untuk VPC, pilih virtual private cloud (VPC) yang Anda buat.
   - Di bawah Pemetaan, pilih Availability Zones yang terkait dengan subnet publik yang tersedia di VPC yang sama yang Anda gunakan dengan endpoint server Anda.
   - Untuk alamat IPv4 dari setiap subnet, pilih salah satu alamat IP elastis yang Anda alokasikan.
6. Di bagian Pendengar dan perutean, masukkan informasi berikut:
   - Untuk Protokol, pilih TLS.
   - Untuk Port, masukkan **`5080`**.
   - Untuk tindakan Default, pilih Buat grup target. Untuk detail pembuatan grup target baru, lihatUntuk membuat grup target.

   Setelah Anda membuat grup target, masukkan namanya di bidang Tindakan default.
7. Di bagian Pengaturan pendengar aman, pilih sertifikat Anda di area sertifikat SSL/TLS default.
8. Pilih Buat penyeimbang beban untuk membuat NLB Anda.
9. (Opsional, tetapi disarankan) Aktifkan log akses untuk Network Load Balancer untuk mempertahankan jejak audit penuh, seperti yang dijelaskan dalam log Access untuk Network Load Balancer Anda.

   Kami merekomendasikan langkah ini karena koneksi TLS dihentikan di NLB. Oleh karena itu, alamat IP sumber yang tercermin dalam grup CloudWatch log Transfer Family AS2 Anda adalah alamat IP pribadi NLB, bukan alamat IP eksternal mitra dagang Anda.

Setelah Anda mengatur penyeimbang beban, klien berkomunikasi dengan penyeimbang beban melalui pendengar port kustom. Kemudian, penyeimbang beban berkomunikasi dengan server melalui port 5080.

Untuk membuat grup target

1. Setelah Anda memilih Buat grup target dalam prosedur sebelumnya, Anda akan dibawa ke halaman Tentukan rincian grup untuk grup target baru.
2. Di bagian Konfigurasi dasar, masukkan informasi berikut.
   - Untuk Pilih jenis target, pilih alamat IP.
   - Untuk Name, masukkan nama untuk grup target.
   - Untuk Protokol, pilih TCP.
   - Untuk Port, masukkan **`5080`**.
   - Untuk jenis alamat IP, pilih IPv4.
   - Untuk VPC, pilih VPC yang Anda buat untuk server Transfer Family AS2 Anda.
3. Di bagian Pemeriksaan Kesehatan, pilih TCP untuk protokol Pemeriksaan Kesehatan.
4. Pilih Berikutnya.
5. Pada halaman Daftar target, masukkan informasi berikut:
   - Untuk Jaringan, konfirmasikan bahwa VPC yang Anda buat untuk server Transfer Family AS2 Anda ditentukan.
   - Untuk alamat IPv4, masukkan alamat IPv4 pribadi dari endpoint server Transfer Family AS2 Anda.

     Jika Anda memiliki lebih dari satu titik akhir untuk server Anda, pilih Tambahkan alamat IPv4 untuk menambahkan baris lain untuk memasukkan alamat IPv4 lain. Ulangi proses ini sampai Anda memasukkan alamat IP pribadi untuk semua titik akhir server Anda.
   - Pastikan Port diatur ke**`5080`**.
   - Pilih Sertakan sebagai tertunda di bawah ini untuk menambahkan entri Anda ke bagian Tinjau target.
6. Di bagian Tinjau target, tinjau target IP Anda.
7. Pilih Buat grup target, lalu kembali ke prosedur sebelumnya untuk membuat NLB Anda dan masukkan grup target baru di mana ditunjukkan.

Uji akses ke server dari alamat IP Elastis

Connect ke server melalui port kustom dengan menggunakan alamat IP Elastis atau nama DNS Network Load Balancer.

> **Important**
>
> Kelola akses ke server Anda dari alamat IP klien dengan menggunakan daftar kontrol akses jaringan (ACL jaringan) untuk subnet yang dikonfigurasi pada penyeimbang beban. Izin ACL jaringan ditetapkan pada tingkat subnet, sehingga aturan berlaku untuk semua sumber daya yang menggunakan subnet. Anda tidak dapat mengontrol akses dari alamat IP klien dengan menggunakan grup keamanan, karena jenis target penyeimbang beban disetel ke alamat IP, bukan Instans. Oleh karena itu, penyeimbang beban tidak mempertahankan alamat IP sumber. Jika pemeriksaan kesehatan Network Load Balancer gagal, ini berarti penyeimbang beban tidak dapat terhubung ke titik akhir server. Untuk memecahkan masalah ini, periksa hal berikut:
>
> - Konfirmasikan bahwa grup keamanan terkait titik akhir server memungkinkan koneksi masuk dari subnet yang dikonfigurasi pada penyeimbang beban. Load balancer harus dapat terhubung ke endpoint server melalui port 5080.
> - Konfirmasikan bahwa Status server sedang Online.

Mentransfer file dengan menggunakan konektor AS2

Konektor AS2 membangun hubungan antara mitra dagang untuk transfer pesan AS2 dari server Transfer Family ke tujuan eksternal milik mitra.

Anda dapat menggunakan Transfer Family untuk mengirim pesan AS2 dengan mereferensikan ID konektor dan jalur ke file, seperti yang diilustrasikan dalam perintah `start-file-transfer` AWS Command Line Interface ()AWS CLI berikut:

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

Untuk mendapatkan detail konektor Anda, jalankan perintah berikut:

```
aws transfer list-connectors
```

`list-connectors`Perintah mengembalikan ID konektor, URL, dan Nama Sumber Daya Amazon (ARN) untuk konektor Anda.

Untuk mengembalikan properti konektor tertentu, jalankan perintah berikut dengan ID yang ingin Anda gunakan:

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

`describe-connector`Perintah mengembalikan semua properti untuk konektor, termasuk URL, peran, profil, Pemberitahuan Disposisi Pesan (mDNS), tag, dan metrik pemantauan.

Anda dapat mengonfirmasi bahwa mitra berhasil menerima file dengan melihat file JSON dan MDN. File-file ini diberi nama sesuai dengan konvensi yang dijelaskan dalamNama dan lokasi file. Jika Anda mengonfigurasi peran logging saat membuat konektor, Anda juga dapat memeriksa CloudWatch log Anda untuk status pesan AS2.

Untuk melihat detail konektor AS2, lihatLihat detail konektor AS2. Untuk informasi selengkapnya tentang membuat konektor AS2, lihatKonfigurasikan konektor AS2.

Nama dan lokasi file

Bagian ini membahas konvensi penamaan file untuk transfer AS2.

Untuk transfer file masuk, perhatikan hal berikut:

- Anda menentukan direktori dasar dalam perjanjian. Direktori dasar adalah nama bucket Amazon S3 yang dikombinasikan dengan awalan, jika ada. Misalnya, `/DOC-EXAMPLE-BUCKET/AS2-folder`.
- Jika file yang masuk berhasil diproses, file (dan file JSON yang sesuai) disimpan ke folder/`processed`. Misalnya, `/DOC-EXAMPLE-BUCKET/AS2-folder/processed`.

  File JSON berisi bidang-bidang berikut:

  - `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`
- Jika file yang masuk tidak dapat diproses dengan sukses, file (dan file JSON yang sesuai) disimpan ke folder`/failed`. Misalnya, `/DOC-EXAMPLE-BUCKET/AS2-folder/failed`.
- File yang ditransfer disimpan dalam `processed` folder sebagai*`original_filename`*`.`*`messageId`*`.`*`original_extension`*. Artinya, ID pesan untuk transfer ditambahkan ke nama file, sebelum ekstensi aslinya.
- File JSON dibuat dan disimpan sebagai *`original_filename`*`.`*`messageId`*`.`*`original_extension`*`.json` file. Selain ID pesan yang ditambahkan, string `.json` ditambahkan ke nama file yang ditransfer.
- File Message Disposition Notice (MDN) dibuat dan disimpan sebagai *`original_filename`*`.`*`messageId`*`.`*`original_extension`*`.mdn` file. Selain ID pesan yang ditambahkan, string `.mdn` ditambahkan ke nama file yang ditransfer.
- Jika ada file inbound bernama`ExampleFileInS3Payload.dat`, file berikut dibuat:
  - Berkas — `ExampleFileInS3Payload.c4d6b6c7-23ea-4b8c-9ada-0cb811dc8b35@44313c54b0a46a36.`
  - JSON — `ExampleFileInS3Payload.c4d6b6c7-23ea-4b8c-9ada-0cb811dc8b35@44313c54b0a46a36.`
  - MDN — `ExampleFileInS3Payload.c4d6b6c7-23ea-4b8c-9ada-0cb811dc8b35@44313c54b0a46a36.`

Untuk transfer keluar, penamaannya serupa, dengan perbedaan bahwa tidak ada file pesan masuk, dan juga, ID transfer untuk pesan yang ditransfer ditambahkan ke nama file. ID transfer dikembalikan oleh operasi `StartFileTransfer` API (atau ketika proses atau skrip lain memanggil operasi ini).

- `transfer-id`Ini adalah pengidentifikasi yang terkait dengan transfer file. Semua permintaan yang merupakan bagian dari `StartFileTransfer` panggilan berbagi `atransfer-id`.
- Direktori dasar sama dengan jalur yang Anda gunakan untuk file sumber. Artinya, direktori dasar adalah jalur yang Anda tentukan dalam operasi atau `start-file-transfer` AWS CLI perintah `StartFileTransfer` API. Sebagai contoh:

  ```
  aws transfer start-file-transfer --send-file-paths /DOC-EXAMPLE-BUCKET/AS2-folder/
  file-to-send.txt
  ```

  Jika Anda menjalankan perintah ini, file MDN dan JSON disimpan di `/DOC-EXAMPLE-BUCKET/AS2-folder/processed` (untuk transfer yang berhasil), atau `/DOC-EXAMPLE-BUCKET/AS2-folder/failed` (untuk transfer yang gagal).
- File JSON dibuat dan disimpan sebagai *original_filename*`.`*transferId*`.`*messageId*`.`*original_extension*`.json` file.
- File MDN dibuat dan disimpan sebagai *original_filename*`.`*transferId*`.`*messageId*`.`*original_extension*`.mdn` file.
- Jika ada file keluar bernama`ExampleFileOutTestOutboundSyncMdn.dat`, file berikut dibuat:
  - 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`

Anda juga dapat memeriksa CloudWatch log untuk melihat detail transfer Anda, termasuk yang gagal.

Kode status

Tabel berikut mencantumkan semua kode status yang dapat dicatat ke CloudWatch log saat Anda atau pasangan Anda mengirim pesan AS2. Langkah-langkah pemrosesan pesan yang berbeda berlaku untuk jenis pesan yang berbeda dan dimaksudkan untuk pemantauan saja. Status COMPLETED dan FAILED mewakili langkah terakhir dalam pemrosesan, dan terlihat dalam file JSON.

Kode	Deskripsi	Pemrosesan selesai?
PENGOLAHAN	Pesan sedang dalam proses dikonversi ke format akhirnya. Misalnya, langkah dekompresi dan dekripsi keduanya memiliki status ini.	Tidak
MDN_TRANSMIT	Pemrosesan pesan adalah mengirimkan respons MDN.	Tidak
MDN_RECEIVE	Pemrosesan pesan menerima respons MDN.	Tidak
DISELESAIKAN	Pemrosesan pesan telah selesai dengan sukses. Keadaan ini termasuk ketika MDN dikirim untuk pesan masuk atau untuk verifikasi MDN dari pesan keluar.	Ya
Failed	Pemrosesan pesan telah gagal. Untuk daftar kode kesalahan, lihatKode kesalahan AS2.	Ya

Contoh file JSON

Bagian ini mencantumkan contoh file JSON untuk transfer masuk dan keluar, termasuk file sampel untuk transfer yang berhasil dan transfer yang gagal.

Contoh file keluar yang berhasil ditransfer:

```
{
  "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"
}
```

Contoh file keluar yang tidak berhasil ditransfer:

```
{
  "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"
}
```

Contoh file inbound yang berhasil ditransfer:

```
{
  "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"
}
```

Contoh file masuk yang tidak berhasil ditransfer:

```
{
  "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"
}
```

Memantau penggunaan AS2

Anda dapat memantau aktivitas AS2 menggunakan Amazon CloudWatch danAWS CloudTrail. Untuk melihat metrik server Transfer Family lainnya, lihat CloudWatch Pencatatan Amazon untuk AWS Transfer Family

Metrik AS2

Metrik	Deskripsi
`InboundMessage`	Jumlah total pesan AS2 yang berhasil diterima dari mitra dagang. Unit: Hitungan Periode: 5 menit
`InboundFailedMessage`	Jumlah total pesan AS2 yang tidak berhasil diterima dari mitra dagang. Artinya, mitra dagang mengirim pesan, tetapi server Transfer Family tidak berhasil memprosesnya. Unit: Hitungan Periode: 5 menit
`OutboundMessage`	Jumlah total pesan AS2 yang berhasil dikirim dari server Transfer Family ke mitra dagang. Unit: Hitungan Periode: 5 menit
`OutboundFailedMessage`	Jumlah total pesan AS2 yang tidak berhasil dikirim ke mitra dagang. Artinya, mereka dikirim dari server Transfer Family, tetapi tidak berhasil diterima oleh mitra dagang. Unit: Hitungan Periode: 5 menit

Kode Status AS2

Tabel berikut mencantumkan semua kode status yang dapat dicatat ke CloudWatch log saat Anda atau pasangan Anda mengirim pesan AS2. Langkah-langkah pemrosesan pesan yang

berbeda berlaku untuk jenis pesan yang berbeda dan dimaksudkan untuk pemantauan saja. Status COMPLETED dan FAILED mewakili langkah terakhir dalam pemrosesan, dan terlihat dalam file JSON.

Kode	Deskripsi	Pemrosesan selesai?
PENGOLAHAN	Pesan sedang dalam proses dikonversi ke format akhirnya. Misalnya, langkah dekompresi dan dekripsi keduanya memiliki status ini.	Tidak
MDN_TRANSMIT	Pemrosesan pesan adalah mengirimkan respons MDN.	Tidak
MDN_RECEIVE	Pemrosesan pesan menerima respons MDN.	Tidak
DISELESAIKAN	Pemrosesan pesan telah selesai dengan sukses. Keadaan ini termasuk ketika MDN dikirim untuk pesan masuk atau untuk verifikasi MDN dari pesan keluar.	Ya
Failed	Pemrosesan pesan telah gagal. Untuk daftar kode kesalahan, lihatKode kesalahan AS2.	Ya

Kode kesalahan AS2

Tabel berikut mencantumkan dan menjelaskan kode kesalahan yang mungkin Anda terima dari transfer file AS2.

Kode kesalahan AS2

Kode	Kesalahan	Deskripsi dan resolusi
`ACCESS_DENIED`	• Akses ditolak. Periksa apakah peran akses Anda memiliki izin yang diperlukan. • Jalur berkas tidak valid *`send-file-path`* • *`Gagal mendapatkan kredensil dengan ErrorCode: kode kesalahan`*	Terjadi saat menangani `StartFileTransfer` permintaan di mana salah satu dari `SendFilePaths` yang tidak valid atau cacat. Artinya, jalur tersebut tidak memiliki nama bucket Amazon S3, atau jalurnya menyertakan karakter yang tidak valid. Juga terjadi jika Transfer Family gagal mengambil peran akses atau peran logging. Pastikan jalur berisi nama bucket Amazon S3 dan nama kunci yang valid.
`AGREEMENT_NOT_FOUND`	Kesepakatan tidak ditemukan.	Entah perjanjian itu tidak ditemukan, atau perjanjian dikaitkan dengan profil yang tidak aktif. Perbarui perjanjian dalam server Transfer Family untuk menyertakan profil aktif.
`CONNECTOR_NOT_FOUND`	Konektor atau konfigurasi terkait tidak ditemukan.	Entah konektor tidak ditemukan, atau konektor dikaitkan dengan profil yang tidak aktif. Perbarui konektor untuk menyertakan profil aktif.

Kode	Kesalahan	Deskripsi dan resolusi
`CREDENTIALS_RETRIEVAL_FAILED`	1. Rahasia tidak ditemukan di Secrets Manager. 2. Tidak dapat mengakses Secrets Manager. 3. Gagal mendekripsi rahasia di Secrets Manager. 4. Tidak bisa mendapatkan nilai rahasia karena pelambatan.	Untuk otentikasi AS2 Basic, rahasianya harus diformat dengan benar. Resolusi berikut sesuai dengan kesalahan yang tercantum di kolom sebelumnya. 1. Pastikan bahwa ID rahasia sudah benar. 2. Pastikan bahwa peran akses memiliki izin yang sesuai untuk membaca rahasia. Peran akses harus menyediakan akses baca dan tulis ke direktori induk dari lokasi file yang digunakan dalam `StartFileTransfer` permintaan. Selain itu, pastikan bahwa peran tersebut menyediakan akses baca dan tulis ke direktori induk dari file yang ingin Anda kirim`StartFileTransfer`. 3. Jika kunci yang dikelola pelanggan digunakan untuk rahasia, pastikan bahwa peran akses memiliki izin untuk kunci AWS Key Management Service (AWS KMS).

Kode	Kesalahan	Deskripsi dan resolusi
		4. Untuk kuota yang berlaku, lihatKuota untuk menangani rahasia.
DECOMPRESSION_FAILED	Gagal mendekompresi pesan.	Entah file yang dikirim rusak, atau algoritma kompresi tidak valid. Kirim ulang pesan dan verifikasi bahwa kompresi ZLIB digunakan, atau kirim ulang pesan tanpa kompresi diaktifkan.
DECRYPT_FAILED	*Gagal mendekripsi pesan Message-ID.* Pastikan bahwa mitra memiliki kunci enkripsi publik yang benar.	Dekripsi gagal. Konfirmasikan bahwa mitra mengirim muatan dengan menggunakan sertifikat yang valid dan enkripsi itu dilakukan dengan menggunakan algoritma enkripsi yang valid.
DECRYPT_FAILED_INVALID_SMIME_FORMAT	Tidak dapat mengurai mimePart yang diselimuti.	Payload MIME rusak atau dalam format SMIME yang tidak didukung. Pengirim harus memastikan bahwa format yang mereka gunakan didukung, dan kemudian mengirim ulang payload.

Kode	Kesalahan	Deskripsi dan resolusi
DECRYPT_FAILED_NO_DECRYPTION_KEY_FOUND	Tidak ditemukan kunci dekripsi yang cocok.	Profil mitra tidak memiliki sertifikat yang ditetapkan yang cocok dengan pesan, atau sertifikat yang cocok dengan pesan sekarang kedaluwarsa atau tidak lagi valid. Anda harus memperbarui profil mitra dan memastikan bahwa itu berisi sertifikat yang valid.
DECRYPT_FAILED_UNSUPPORTED_ENCRYPTION_ALG	*Dekripsi Muatan SMIME diminta menggunakan algoritma yang tidak didukung dengan ID: Encryption-ID.*	Pengirim jarak jauh telah mengirim muatan AS2 dengan algoritma enkripsi yang tidak didukung. Pengirim harus memilih algoritma enkripsi yang didukung olehAWS Transfer Family.
DUPLICATE_MESSAGE	Langkah duplikat atau proses ganda.	Muatan memiliki langkah pemrosesan duplikat. Misalnya, ada dua langkah enkripsi. Kirim ulang pesan dengan satu langkah untuk penandatanganan, kompresi, dan enkripsi.

Kode	Kesalahan	Deskripsi dan resolusi
`ENCRYPT_FAILED_NO_ENCRYPTION_KEY_FOUND`	Tidak ada sertifikat enkripsi publik yang valid yang ditemukan di profil: *`Local-profile-ID`*	Transfer Family mencoba mengenkripsi pesan keluar, tetapi tidak ada sertifikat enkripsi yang ditemukan untuk profil lokal. Opsi resolusi: • Pastikan bahwa profil lokal memiliki sertifikat dan kunci pribadi untuk enkripsi terlampir. • Pastikan sertifikat enkripsi saat ini aktif.
`ENCRYPTION_FAILED`	Gagal mengenkripsi *`nama`* file.	File yang akan dikirim tidak tersedia untuk enkripsi. Verifikasi bahwa file tersebut berada di lokasi AS2 yang diharapkan dan yang AWS Transfer Family memiliki izin untuk membaca file.
`FILE_SIZE_TOO_LARGE`	Ukuran file terlalu besar.	Ini terjadi saat mengirim atau menerima file yang melebihi batas ukuran file.

Kode	Kesalahan	Deskripsi dan resolusi
`HTTP_ERROR_RESPONSE_FROM_PARTNER`	*`Partner-URL mengembalikan status 400 untuk pesan dengan ID=Message-ID.`*	Berkomunikasi dengan server AS2 mitra mengembalikan kode respons HTTP yang tidak terduga. Mitra mungkin dapat memberikan lebih banyak diagnostik dari log server AS2 mereka.
`INSUFFICENT_MESSAGE_SECURITY_UNENCRYPTED`	Enkripsi diperlukan.	Mitra mengirim pesan yang tidak terenkripsi ke Transfer Family, yang tidak didukung. Pengirim harus menggunakan muatan terenkripsi.
`INVALID_ENDPOINT_PROTOCOL`	Hanya HTTP dan HTTPS yang didukung.	Anda harus menentukan HTTP atau HTTPS sebagai protokol dalam konfigurasi konektor AS2 Anda.

Kode	Kesalahan	Deskripsi dan resolusi
`INVALID_REQUEST`	1. Ada masalah dengan header pesan. 2. Tidak dapat mengurai JSON rahasia. Secret JSON tidak cocok dengan format yang diharapkan. 3. Rahasia harus berupa string JSON. 4. Nama pengguna tidak boleh mengandung titik dua. Nama pengguna tidak boleh mengandung karakter kontrol. Nama pengguna harus hanya berisi karakter ASCII. Kata sandi tidak boleh mengandung karakter kontrol. Kata sandi harus hanya berisi karakter ASCII.	Kesalahan ini memiliki beberapa penyebab. Resolusi berikut sesuai dengan kesalahan yang tercantum di kolom sebelumnya. 1. Periksa `as2-from` dan `as2-to` bidang. Pastikan ID pesan asli akurat untuk format MDN. Pastikan juga bahwa format ID pesan tidak hilang header AS2. 2. Pastikan bahwa nilai rahasia cocok dengan format yang didokumen tasikan, seperti yang dijelaskan dalamAktifkan otentikasi dasar untuk konektor AS2. 3. Pastikan bahwa rahasia disediakan sebagai string, dan bukan sebagai biner. 4. Lakukan koreksi yang diperlukan untuk nama pengguna atau kata sandi.

Kode	Kesalahan	Deskripsi dan resolusi
`INVALID_URL_FORMAT`	*`Format URL tidak valid: URL`*	Ini terjadi ketika Anda mengirim pesan keluar menggunakan konektor yang dikonfigurasi dengan URL yang salah bentuk. Pastikan konektor dikonfigu rasi dengan URL HTTP atau HTTPS yang valid.
`MDN_RESPONSE_INDIC ATES_AUTHENTICATIO N_FAILED`	Tidak berlaku	Penerima tidak dapat mengautentikasi pengirim. Mitra dagang mengembal ikan MDN ke Transfer Family dengan pengubah disposisi Error: authentication-failed.
`MDN_RESPONSE_INDIC ATES_DECOMPRESSION _FAILED`	Tidak berlaku	Ini terjadi ketika penerima tidak dapat mendekomp resi isi pesan. Mitra dagang mengembalikan MDN ke Transfer Family dengan pengubah disposisi Error: decompression-failed.
`MDN_RESPONSE_INDIC ATES_DECRYPTION_FA ILED`	Tidak berlaku	Penerima tidak dapat mendekripsi isi pesan. Mitra dagang mengembalikan MDN ke Transfer Family dengan pengubah disposisi Error: authentication-failed.

Kode	Kesalahan	Deskripsi dan resolusi
`MDN_RESPONSE_INDICATES_INSUFFICIENT_MESSAGE_SECURITY`	Tidak berlaku	Penerima mengharapkan pesan ditandatangani atau dienkripsi, tetapi sebenarnya tidak. Mitra dagang mengembalikan MDN ke Transfer Family dengan pengubah disposisi Kesalahan :. insufficient-message-security Aktifkan penandatanganan dan/atau enkripsi pada konektor agar sesuai dengan harapan mitra dagang.
`MDN_RESPONSE_INDICATES_INTEGRITY_CHECK_FAILED`	Tidak berlaku	Penerima tidak dapat memverifikasi integritas konten. Mitra dagang mengembalikan MDN ke Transfer Family dengan pengubah disposisi Kesalahan :. integrity-check-failed
`PATH_NOT_FOUND`	Tidak dapat membuat jalur *`berkas direktori.`* Jalur induk tidak dapat ditemukan.	Transfer Family mencoba membuat direktori di bucket Amazon S3 pelanggan, tetapi bucket tidak ditemukan. Pastikan bahwa setiap jalur yang disebutkan dalam `StartFileTransfer` perintah berisi nama bucket yang ada.

Kode	Kesalahan	Deskripsi dan resolusi
SEND_FILE_NOT_FOUND	Jalur file jalur *file tidak ditemukan* .	Transfer Family tidak dapat menemukan file dalam operasi kirim file. Periksa apakah direktori dan jalur home yang dikonfigu rasi valid dan Transfer Family telah membaca izin untuk file tersebut.
SERVER_NOT_FOUND	Server yang terkait dengan pesan tidak dapat ditemukan.	Transfer Family tidak dapat menemukan server saat menerima pesan. Hal ini dapat terjadi jika server dihapus selama pemrosesan pesan masuk.
SERVER_NOT_ONLINE	Server *Server-ID tidak online*.	Server Transfer Family sedang offline. Mulai server sehingga dapat menerima dan memproses pesan.
SIGNING_FAILED	Gagal menandatangani berkas.	File yang akan dikirim tidak tersedia untuk ditandatangani, atau penandatanganan tidak dapat dilakukan. Verifikasi bahwa file tersebut berada di lokasi AS2 yang diharapkan dan yang AWS Transfer Family memiliki izin untuk membaca file.

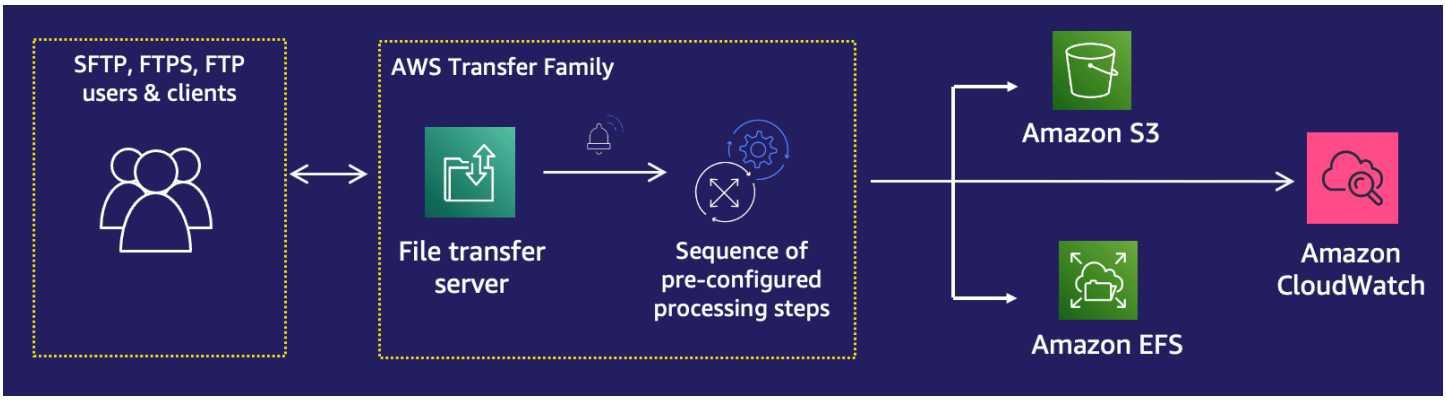
Kode	Kesalahan	Deskripsi dan resolusi
`SIGNING_FAILED_NO_SIGNING_KEY_FOUND`	Tidak ada sertifikat yang ditemukan untuk profil: *`Local-profile-ID`* .	Mencoba menandatangani pesan keluar, tetapi tidak ada sertifikat penandatanganan yang ditemukan untuk profil lokal. Opsi resolusi: • Pastikan bahwa profil lokal memiliki sertifikat dan kunci pribadi untuk penandata nganan terlampir. • Pastikan sertifikat penandatanganan saat ini aktif.
`UNABLE_RESOLVE_HOST_TO_IP_ADDRESS`	Tidak dapat menyelesaikan nama host ke alamat IP.	Transfer Family tidak dapat melakukan resolusi alamat DNS ke IP pada server DNS publik yang dikonfigurasi di konektor AS2. Perbarui konektor untuk menunjuk ke URL mitra yang valid.
`UNABLE_TO_CONNECT_TO_REMOTE_HOST_OR_IP`	Koneksi ke titik akhir habis waktu.	Transfer Family tidak dapat membuat koneksi soket ke server AS2 mitra yang dikonfigurasi. Periksa apakah server AS2 mitra tersedia di alamat IP yang dikonfigurasi.

Kode	Kesalahan	Deskripsi dan resolusi
`UNABLE_TO_RESOLVE_HOSTNAME`	Tidak dapat menyelesaikan nama host nama *`host`*.	Server Transfer Family tidak dapat menyelesaikan nama host mitra dengan menggunakan server DNS publik. Periksa apakah host yang dikonfigurasi terdaftar dan catatan DNS memiliki waktu untuk mempublikasikan.
`VERIFICATION_FAILED`	Verifikasi tanda tangan gagal untuk *`ID pesan`* AS2 atau kode MIC tidak cocok.	Periksa apakah sertifikat penandatanganan pengirim cocok dengan sertifikat penandatanganan untuk profil jarak jauh. Periksa juga apakah algoritma MIC kompatibel denganAWS Transfer Family.

Kode	Kesalahan	Deskripsi dan resolusi
`VERIFICATION_FAILED_NO_MATCHING_KEY_FOUND`	• Tidak ada tanda tangan pesan pencocokan sertifikat publik yang dapat ditemukan di profil: *`Partner-profile-ID`* . • *`Tidak bisa mendapatkan sertifikat untuk profil yang tidak ada: Partner-profile-ID.`* • Tidak ada sertifikat yang valid ditemukan di profil: *`Partner-profile-ID`* .	AWS Transfer Familysed ang mencoba memverifikasi tanda tangan untuk pesan yang diterima, tetapi tidak ada sertifikat penandatanganan yang cocok ditemukan untuk profil mitra. Opsi resolusi: • Pastikan bahwa profil mitra memiliki sertifikat penandatanganan terlampir. • Pastikan sertifikat saat ini aktif. • Pastikan bahwa sertifikat adalah sertifikat penandatanganan yang benar untuk mitra.

AWS Transfer Family alur kerja terkelola

AWS Transfer Family mendukung alur kerja terkelola untuk pemrosesan file. Dengan alur kerja terkelola, Anda dapat memulai alur kerja setelah file ditransfer melalui SFTP, FTPS, atau FTP. Dengan menggunakan fitur ini, Anda dapat dengan aman dan hemat biaya memenuhi persyaratan kepatuhan Anda untuk pertukaran file business-to-business (B2B) dengan mengoordinasikan semua langkah yang diperlukan untuk pemrosesan file. Selain itu, Anda mendapat manfaat dari end-to-end audit dan visibilitas.



Dengan mengatur tugas pemrosesan file, alur kerja terkelola membantu Anda memproses data sebelum dikonsumsi oleh aplikasi hilir. Tugas pemrosesan file tersebut mungkin termasuk:

- Memindahkan file ke folder khusus pengguna.
- Mendekripsi file sebagai bagian dari alur kerja.
- Menandai file.
- Melakukan pemrosesan kustom dengan membuat dan melampirkan AWS Lambda fungsi ke alur kerja.
- Mengirim pemberitahuan ketika file telah berhasil ditransfer. (Untuk posting blog yang merinci kasus penggunaan ini, lihat Menyesuaikan pemberitahuan pengiriman file menggunakan alur kerja AWS Transfer Family terkelola.)

Untuk dengan cepat mereplikasi dan menstandarisasi tugas pemrosesan file pasca-unggah umum yang mencakup beberapa unit bisnis di organisasi Anda, Anda dapat menerapkan alur kerja dengan menggunakan infrastruktur sebagai kode (IAc). Anda dapat menentukan alur kerja terkelola yang akan dimulai pada file yang diunggah secara penuh. Anda juga dapat menentukan alur kerja terkelola yang berbeda untuk dimulai pada file yang hanya diunggah sebagian karena pemutusan sesi prematur. Penanganan pengecualian bawaan membantu Anda bereaksi dengan cepat terhadap hasil

pemrosesan file, sambil menawarkan Anda kontrol atas cara menangani kegagalan. Selain itu, setiap langkah alur kerja menghasilkan log terperinci, yang dapat Anda audit untuk melacak garis keturunan data.

Untuk memulai, lakukan tugas-tugas berikut:

1. Siapkan alur kerja Anda agar berisi tindakan pra-pemrosesan, seperti menyalin, menandai, dan langkah-langkah lain berdasarkan kebutuhan Anda. Lihat Buat alur kerja untuk detail.
2. Konfigurasikan peran eksekusi, yang digunakan Transfer Family untuk menjalankan alur kerja. Lihat Kebijakan IAM untuk alur kerja untuk detail.
3. Petakan alur kerja ke server, sehingga pada saat kedatangan file, tindakan yang ditentukan dalam alur kerja ini dievaluasi dan dimulai secara real time. Lihat Konfigurasikan dan jalankan alur kerja untuk detail.

Informasi terkait

- Untuk memantau eksekusi alur kerja Anda, lihat. Menggunakan CloudWatch metrik untuk Transfer Family
- Untuk detail log eksekusi dan informasi pemecahan masalah, lihat. Memecahkan masalah kesalahan terkait alur kerja menggunakan Amazon CloudWatch
- Kami memiliki lokakarya yang dapat Anda hadiri, di mana Anda dapat membangun solusi transfer file. Solusi ini memanfaatkan AWS Transfer Family endpoint SFTP/FTPS terkelola serta Amazon Cognito dan DynamoDB untuk manajemen pengguna. Anda dapat melihat detail untuk lokakarya ini di sini.
- Lihat Alur Kerja AWS Transfer Family Terkelola untuk pengenalan singkat tentang alur kerja Transfer Family.

Topik

- Buat alur kerja
- Gunakan langkah-langkah yang telah ditentukan
- Gunakan langkah-langkah pemrosesan file khusus
- Kebijakan IAM untuk alur kerja
- Penanganan pengecualian untuk alur kerja
- Pantau eksekusi alur kerja
- Buat alur kerja dari template

- Menghapus alur kerja dari server Transfer Family
- Pembatasan dan batasan alur kerja yang dikelola

Untuk bantuan selengkapnya untuk memulai alur kerja terkelola, lihat sumber daya berikut:

- AWS Transfer Family video demo alur kerja terkelola
- Membangun platform transfer file cloud-native menggunakan AWS Transfer Family alur kerja posting blog

Buat alur kerja

Anda dapat membuat alur kerja terkelola menggunakan AWS Management Console, seperti yang dijelaskan dalam topik ini. Untuk membuat proses pembuatan alur kerja semudah mungkin, panel bantuan kontekstual tersedia untuk sebagian besar bagian di konsol.

Alur kerja memiliki dua jenis langkah:

- Langkah nominal — Langkah nominal adalah langkah pemrosesan file yang ingin Anda terapkan ke file yang masuk. Jika Anda memilih lebih dari satu langkah nominal, setiap langkah diproses dalam urutan linier.
- Langkah penanganan pengecualian - Penangan pengecualian adalah langkah pemrosesan file yang AWS Transfer Family dijalankan jika ada langkah nominal yang gagal atau mengakibatkan kesalahan validasi.

Buat alur kerja

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi kiri, pilih Alur kerja.
3. Pada halaman Alur kerja, pilih Buat alur kerja.
4. Pada halaman Buat alur kerja, masukkan deskripsi. Deskripsi ini muncul di halaman Alur Kerja.
5. Di bagian Langkah nominal, pilih Tambah langkah. Tambahkan satu atau lebih langkah.
    a. Pilih jenis langkah dari opsi yang tersedia. Untuk informasi selengkapnya tentang berbagai jenis langkah, lihatthe section called “Gunakan langkah-langkah yang telah ditentukan”.
    b. Pilih Berikutnya, lalu konfigurasikan parameter untuk langkah tersebut.

c. Pilih Berikutnya, lalu tinjau detail untuk langkahnya.

d. Pilih Buat langkah untuk menambahkan langkah dan melanjutkan.

e. Lanjutkan menambahkan langkah-langkah sesuai kebutuhan. Jumlah maksimum langkah dalam alur kerja adalah 8.

f. Setelah Anda menambahkan semua langkah nominal yang diperlukan, gulir ke bawah ke penangan Exception - bagian opsional, dan pilih Tambah langkah.

> **Note**
>
> Agar Anda diberitahu tentang kegagalan secara real time, kami sarankan Anda menyiapkan penangan pengecualian dan langkah-langkah untuk mengeksekusi ketika alur kerja Anda gagal.

6. Untuk mengonfigurasi penangan pengecualian, tambahkan langkah dengan cara yang sama seperti yang dijelaskan sebelumnya. Jika file menyebabkan langkah apa pun untuk melempar pengecualian, penangan pengecualian Anda dipanggil satu per satu.
7. (Opsional) Gulir ke bawah ke bagian Tag, dan tambahkan tag untuk alur kerja Anda.
8. Tinjau konfigurasi, dan pilih Buat alur kerja.

> **Important**
>
> Setelah membuat alur kerja, Anda tidak dapat mengeditnya, jadi pastikan untuk meninjau konfigurasi dengan cermat.

Konfigurasikan dan jalankan alur kerja

Sebelum Anda dapat menjalankan alur kerja, Anda harus mengaitkannya dengan server Transfer Family.

Untuk mengonfigurasi Transfer Family untuk menjalankan alur kerja pada file yang diunggah

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi kiri, pilih Server.

   - Untuk menambahkan alur kerja ke server yang ada, pilih server yang ingin Anda gunakan untuk alur kerja Anda.

- Atau, buat server baru dan tambahkan alur kerja ke dalamnya. Untuk informasi selengkapnya, lihat Mengkonfigurasi titik akhir server SFTP, FTPS, atau FTP.

3. Pada halaman detail untuk server, gulir ke bawah ke bagian Detail tambahan, lalu pilih Edit.

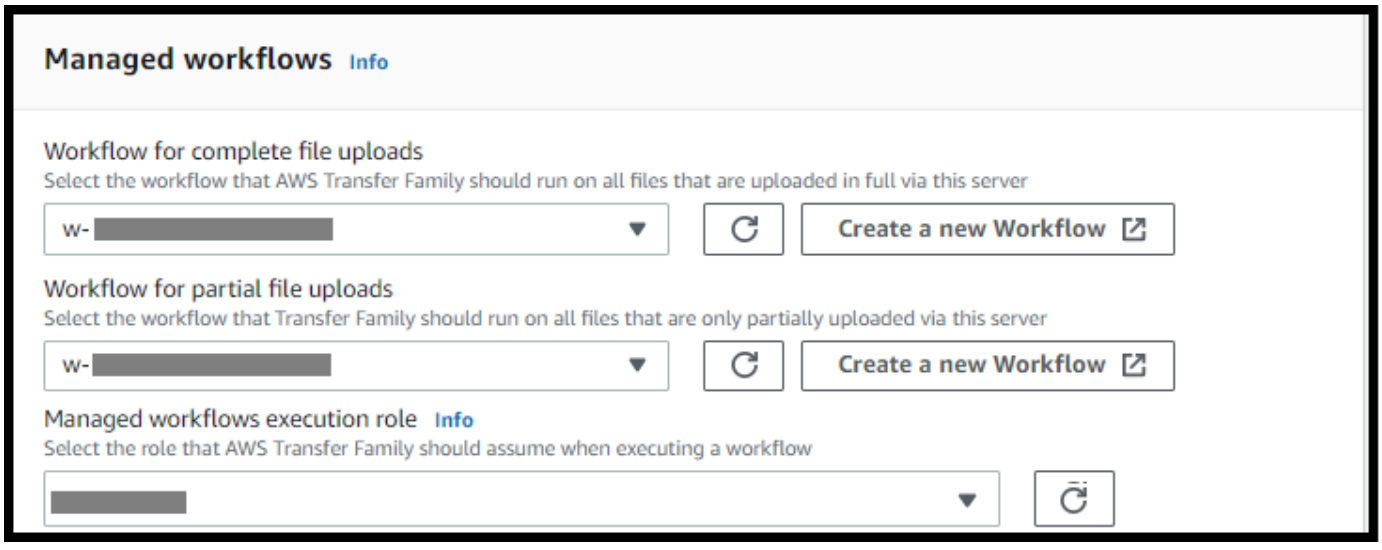
> **Note**
>
> Secara default, server tidak memiliki alur kerja terkait. Anda menggunakan bagian Detail tambahan untuk mengaitkan alur kerja dengan server yang dipilih.

4. Pada halaman Edit detail tambahan, di bagian Alur kerja terkelola, pilih alur kerja yang akan dijalankan di semua unggahan.

> **Note**
>
> Jika Anda belum memiliki alur kerja, pilih Buat Alur Kerja baru untuk membuatnya.

a. Pilih ID alur kerja yang akan digunakan.

b. Pilih peran eksekusi. Ini adalah peran yang diasumsikan Transfer Family saat menjalankan langkah-langkah alur kerja. Untuk informasi selengkapnya, lihat Kebijakan IAM untuk alur kerja. Pilih Simpan.



> **Note**
>
> Jika Anda tidak lagi ingin alur kerja dikaitkan dengan server, Anda dapat menghapus asosiasi. Untuk detailnya, lihat Menghapus alur kerja dari server Transfer Family.

Untuk menjalankan alur kerja

Untuk menjalankan alur kerja, Anda mengunggah file ke server Transfer Family yang dikonfigurasi dengan alur kerja terkait.

> **Note**
>
> Setiap kali Anda menghapus alur kerja dari server dan menggantinya dengan yang baru, atau memperbarui konfigurasi server (yang memengaruhi peran eksekusi alur kerja), Anda harus menunggu sekitar 10 menit sebelum menjalankan alur kerja baru. Server Transfer Family menyimpan cache detail alur kerja, dan dibutuhkan waktu 10 menit bagi server untuk menyegarkan cache-nya.
> Selain itu, Anda harus keluar dari sesi SFTP aktif apa pun, dan kemudian masuk kembali setelah masa tunggu 10 menit untuk melihat perubahannya.

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
>
```

Setelah file Anda diunggah, tindakan yang ditentukan dilakukan pada file Anda. Misalnya, jika alur kerja Anda berisi langkah penyalinan, file tersebut disalin ke lokasi yang Anda tentukan di langkah itu. Anda dapat menggunakan Amazon CloudWatch Logs untuk melacak langkah-langkah yang dijalankan dan status eksekusi mereka.

Lihat detail alur kerja

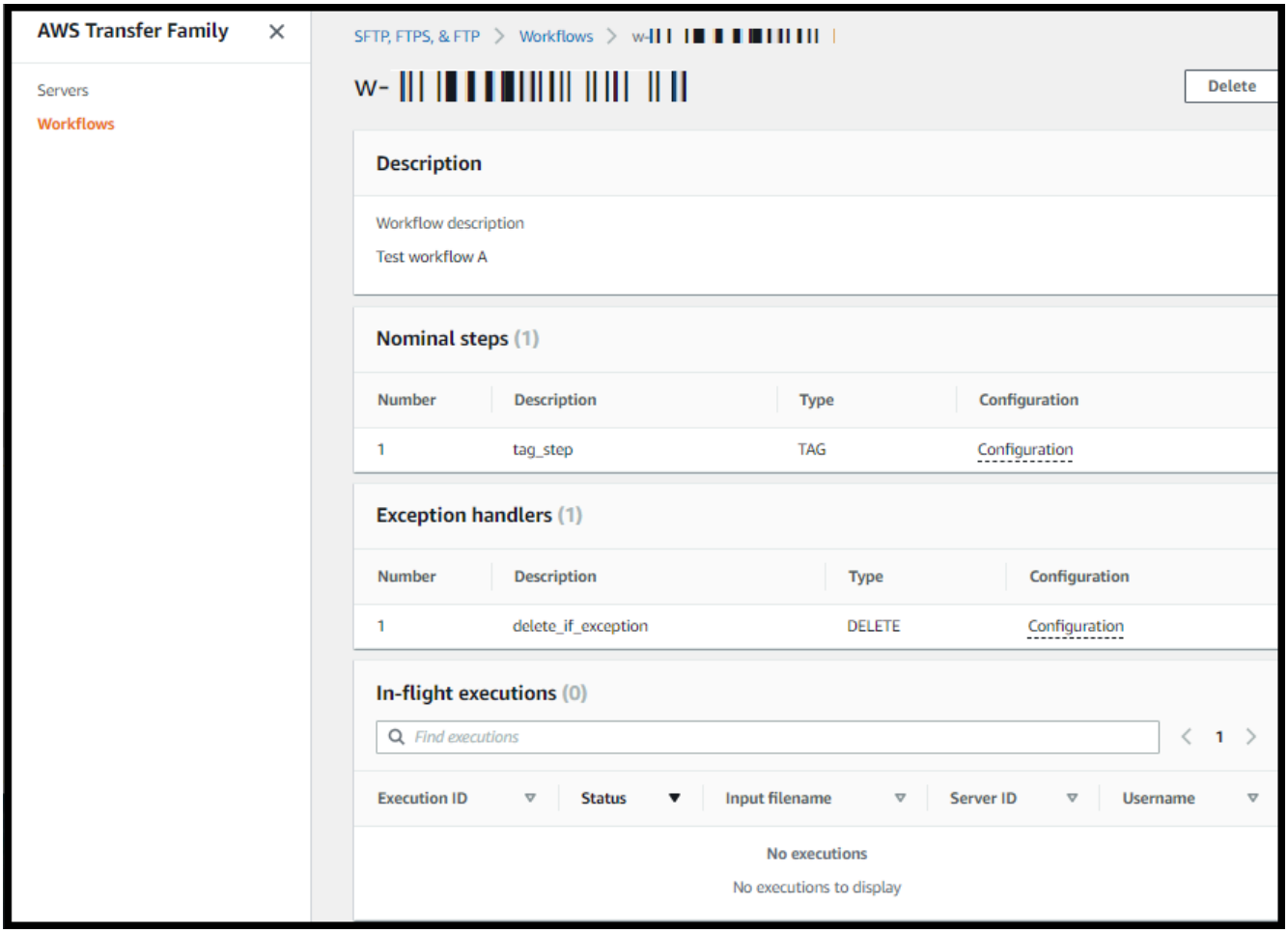
Anda dapat melihat detail tentang alur kerja yang dibuat sebelumnya atau eksekusi alur kerja. Untuk melihat detail ini, Anda dapat menggunakan konsol atau AWS Command Line Interface (AWS CLI).

Console

Lihat detail alur kerja

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi kiri, pilih Alur kerja.
3. Pada halaman Alur Kerja, pilih alur kerja.

   Halaman detail alur kerja terbuka.

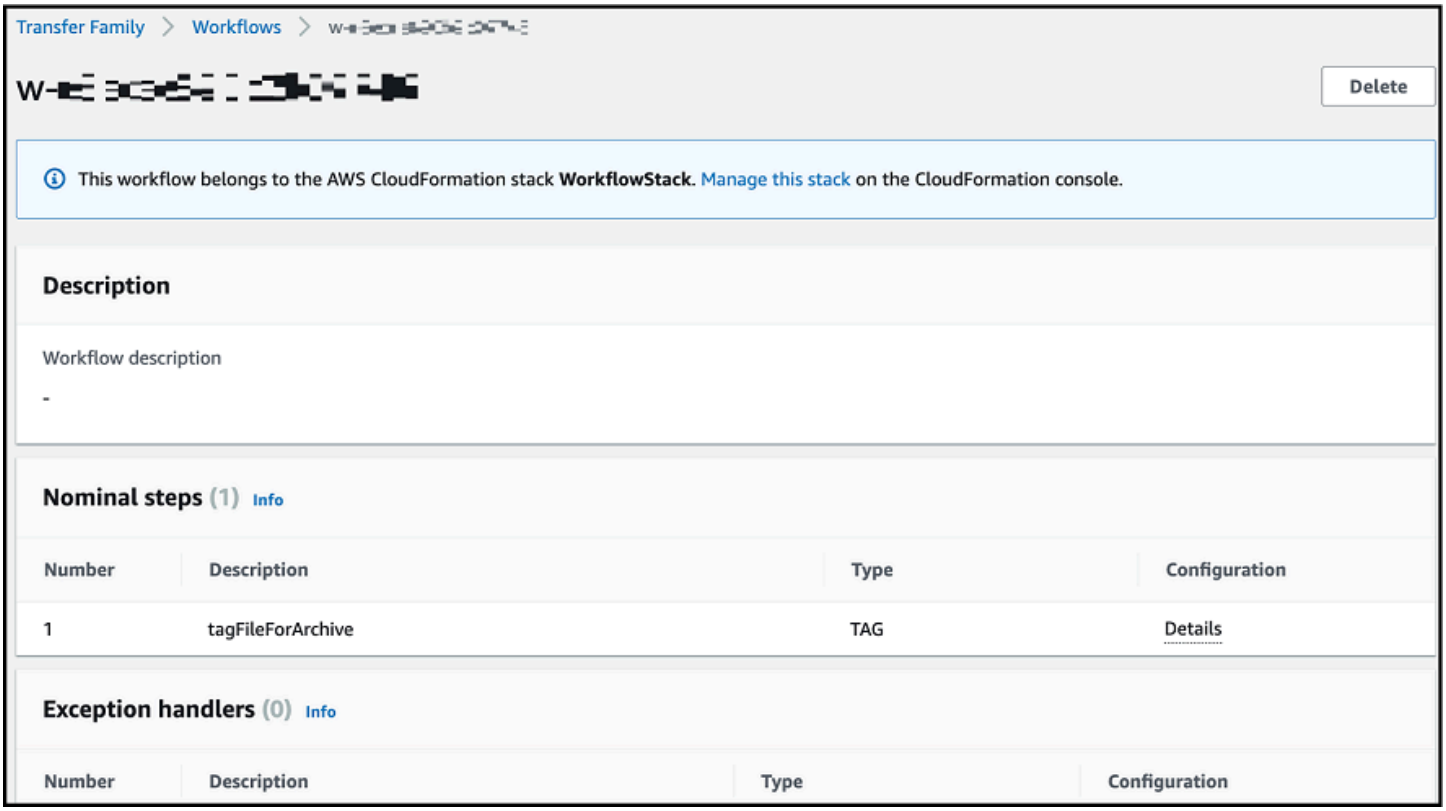


CLI

Untuk melihat detail alur kerja, gunakan perintah `describe-workflow` CLI, seperti yang ditunjukkan pada contoh berikut. Ganti ID alur kerja *w-1234567890abcdef0* dengan nilai Anda sendiri. Untuk informasi selengkapnya, lihat menjelaskan alur kerja di Referensi Perintah.AWS CLI

```
# 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": {}
    }
}
```

Jika alur kerja Anda dibuat sebagai bagian dari AWS CloudFormation tumpukan, Anda dapat mengelola alur kerja menggunakan AWS CloudFormation konsol (https://console.aws.amazon.com/cloudformation).



Gunakan langkah-langkah yang telah ditentukan

Saat membuat alur kerja, Anda dapat memilih untuk menambahkan salah satu langkah yang telah ditentukan berikut yang dibahas dalam topik ini. Anda juga dapat memilih untuk menambahkan langkah-langkah pemrosesan file kustom Anda sendiri. Untuk informasi selengkapnya, lihat the section called “Gunakan langkah-langkah pemrosesan file khusus”.

Topik

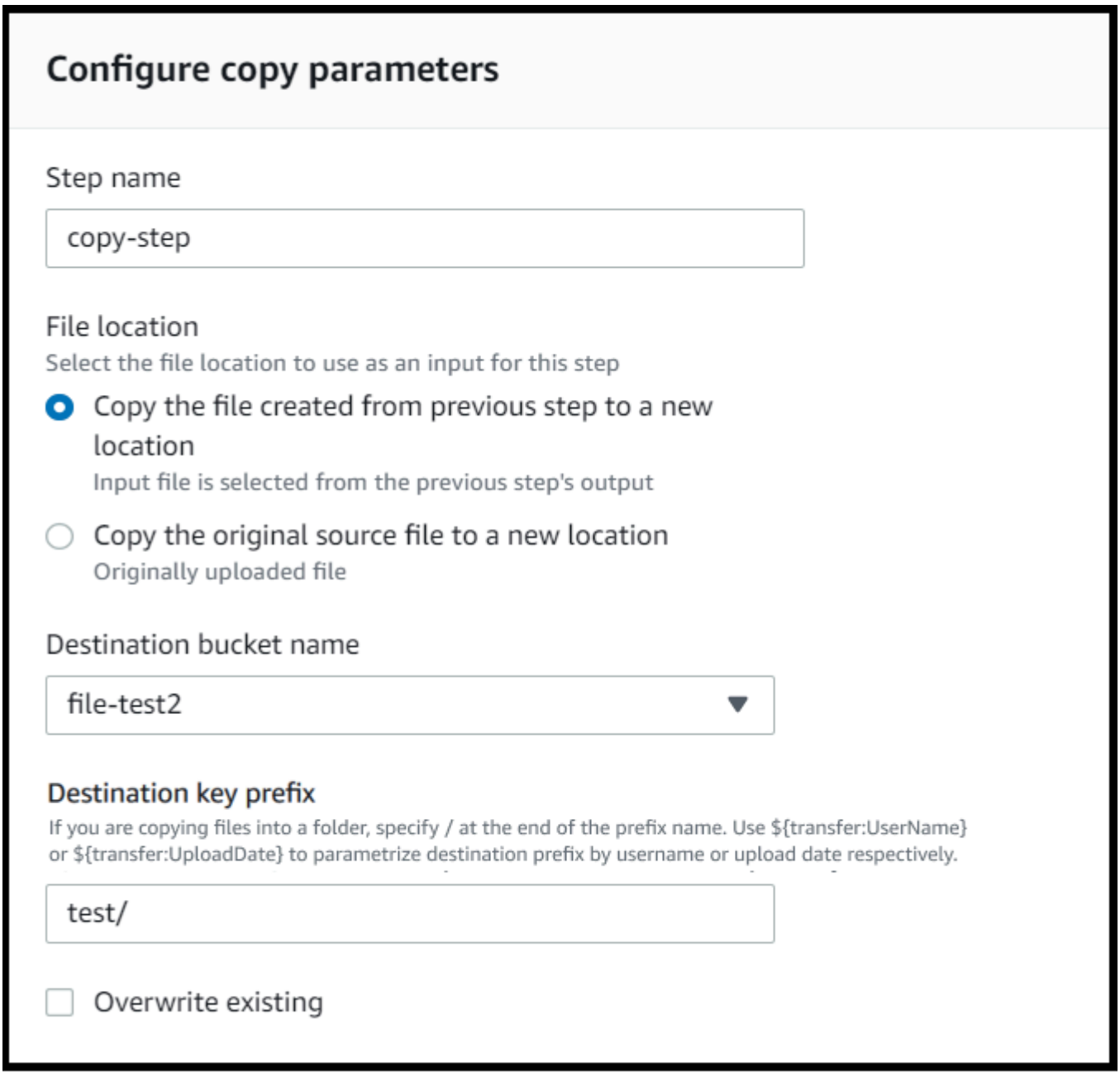
- Salin berkas
- Dekripsi file
- Berkas tag
- Hapus berkas
- Variabel bernama untuk alur kerja
- Contoh tag dan hapus alur kerja

Salin berkas

Langkah salin file membuat salinan file yang diunggah di lokasi Amazon S3 baru. Saat ini, Anda dapat menggunakan langkah salin file hanya dengan Amazon S3.

Langkah copy file berikut menyalin file ke `test` folder di bucket `file-test` tujuan.

Jika langkah salin file bukan langkah pertama alur kerja Anda, Anda dapat menentukan lokasi File. Dengan menentukan lokasi file, Anda dapat menyalin file yang digunakan pada langkah sebelumnya atau file asli yang diunggah. Anda dapat menggunakan fitur ini untuk membuat beberapa salinan dari file asli sambil menjaga file sumber tetap utuh untuk arsip file dan penyimpanan catatan. Sebagai contoh, lihat Contoh tag dan hapus alur kerja.



Berikan ember dan detail kunci

Anda harus memberikan nama bucket dan kunci untuk tujuan langkah salin file. Kuncinya bisa berupa nama jalur atau nama file. Apakah kunci diperlakukan sebagai nama jalur atau nama file ditentukan oleh apakah Anda mengakhiri kunci dengan karakter garis miring (/) maju.

Jika karakter terakhir adalah/, file Anda disalin ke folder, dan namanya tidak berubah. Jika karakter terakhir adalah alfanumerik, file yang Anda unggah diubah namanya menjadi nilai kunci. Dalam hal

ini, jika file dengan nama itu sudah ada, perilaku tergantung pada pengaturan untuk bidang Timpa yang ada.

- Jika Timpa yang ada dipilih, file yang ada diganti dengan file yang sedang diproses.
- Jika Timpa yang ada tidak dipilih, tidak ada yang terjadi, dan pemrosesan alur kerja berhenti.

> **Tip**
>
> Jika penulisan bersamaan dijalankan pada jalur file yang sama, hal itu dapat mengakibatkan perilaku yang tidak terduga saat menimpa file.

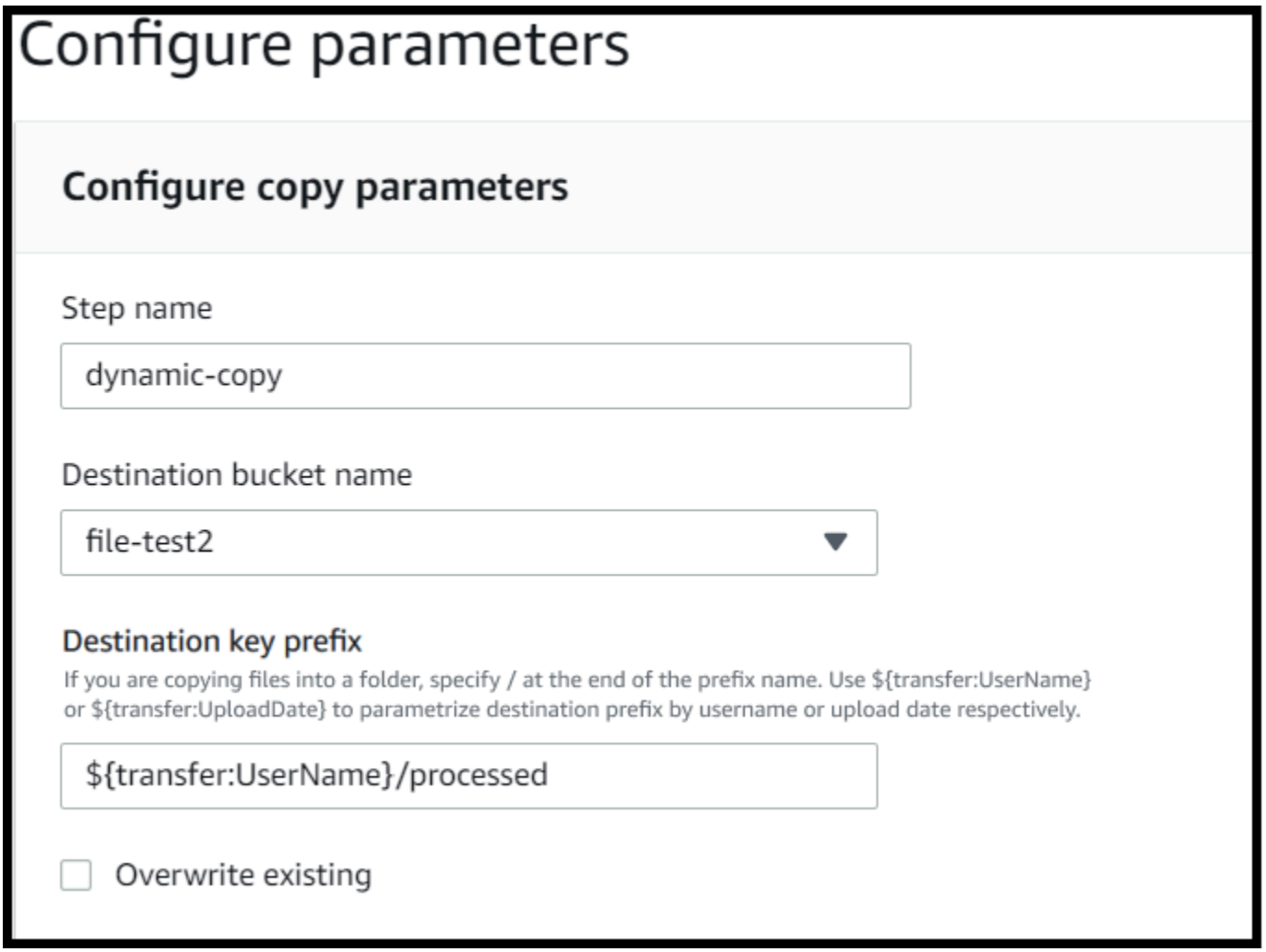
Misalnya, jika nilai kunci Anda adalah`test/`, file yang Anda unggah akan disalin ke folder. `test` Jika nilai kunci Anda`test/today`, (dan Timpa yang ada dipilih) setiap file yang Anda unggah disalin ke file bernama `today` di `test` folder, dan setiap file berikutnya menimpa yang sebelumnya.

> **Note**
>
> Amazon S3 mendukung bucket dan objek, dan tidak memiliki hierarki. Namun, Anda dapat menggunakan awalan dan pembatas dalam nama kunci objek untuk menyiratkan hierarki dan mengatur data Anda dengan cara yang mirip dengan folder.

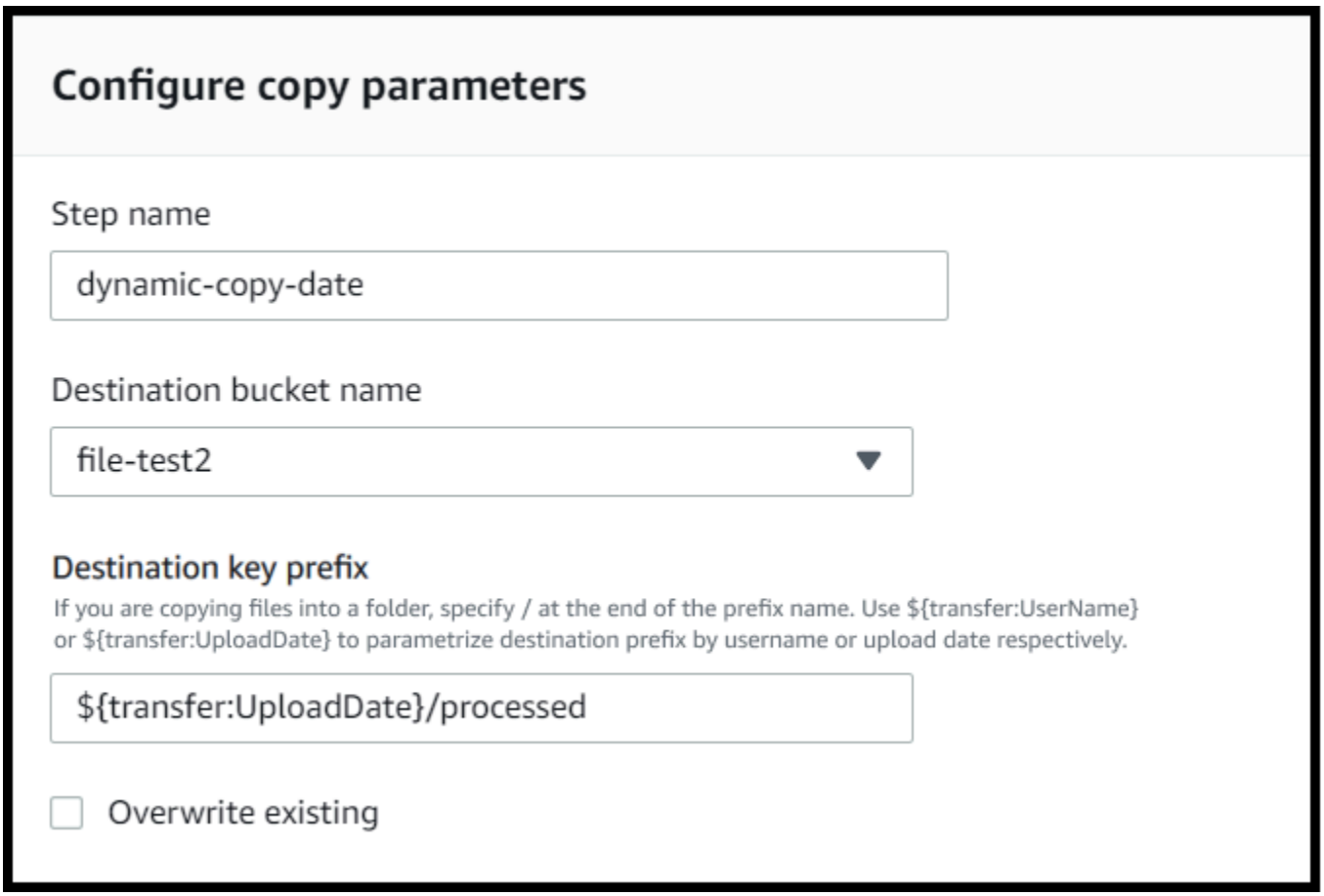
Gunakan variabel bernama dalam langkah copy file

Dalam langkah salin file, Anda dapat menggunakan variabel untuk menyalin file Anda secara dinamis ke folder khusus pengguna. Saat ini, Anda dapat menggunakan `${transfer:UserName}` atau `${transfer:UploadDate}` sebagai variabel untuk menyalin file ke lokasi tujuan untuk pengguna tertentu yang mengunggah file, atau berdasarkan tanggal saat ini.

Dalam contoh berikut, jika pengguna `richard-roe` mengunggah file, itu akan disalin ke folder`file-test2/richard-roe/processed/`. Jika pengguna `mary-major` mengunggah file, itu akan disalin ke folder. `file-test2/mary-major/processed/`



Demikian pula, Anda dapat menggunakan `${transfer:UploadDate}` sebagai variabel untuk menyalin file ke lokasi tujuan yang dinamai untuk tanggal saat ini. Dalam contoh berikut, jika Anda menetapkan tujuan `${transfer:UploadDate}/processed` pada 1 Februari 2022, file yang diunggah akan disalin ke folder`file-test2/2022-02-01/processed/`.



Anda juga dapat menggunakan kedua variabel ini bersama-sama, menggabungkan fungsionalitasnya. Sebagai contoh:

- Anda dapat mengatur key prefix Destination ke**`folder/${transfer:UserName}/${transfer:UploadDate}/`**, yang akan membuat folder bersarang, misalnya. `folder/marymajor/2023-01-05/`
- Anda dapat mengatur key prefix Destination ke**`folder/${transfer:UserName}-${transfer:UploadDate}/`**, untuk menggabungkan dua variabel, misalnya. `folder/marymajor-2023-01-05/`

Izin IAM untuk langkah penyalinan

Agar langkah penyalinan berhasil, pastikan peran eksekusi untuk alur kerja Anda berisi izin berikut.

```
{
            "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**
>
> `s3:ListBucket`Izin hanya diperlukan jika Anda tidak memilih Timpa yang sudah ada. Izin ini memeriksa bucket Anda untuk melihat apakah file dengan nama yang sama sudah ada. Jika Anda telah memilih Timpa yang ada, alur kerja tidak perlu memeriksa file, dan hanya bisa menulisnya.
> Jika file Amazon S3 Anda memiliki tag, Anda perlu menambahkan satu atau dua izin ke kebijakan IAM Anda.
>
> - Tambahkan `s3:GetObjectTagging` file Amazon S3 yang tidak berversi.
> - Tambahkan `s3:GetObjectVersionTagging` untuk file Amazon S3 yang berversi.

Dekripsi file

Blog AWS penyimpanan memiliki posting yang menjelaskan cara mengenkripsi dan mendekripsi file, mengenkripsi dan mendekripsi file dengan PGP dan. AWS Transfer Family

Gunakan dekripsi PGP dalam alur kerja Anda

Transfer Family memiliki dukungan bawaan untuk dekripsi Pretty Good Privacy (PGP). Anda dapat menggunakan dekripsi PGP pada file yang diunggah melalui SFTP, FTPS, atau FTP ke Amazon Simple Storage Service (Amazon S3) atau Amazon Elastic File System (Amazon EFS).

Untuk menggunakan dekripsi PGP, Anda harus membuat dan menyimpan kunci pribadi PGP yang akan digunakan untuk dekripsi file Anda. Pengguna Anda kemudian dapat mengenkripsi file dengan menggunakan kunci enkripsi PGP yang sesuai sebelum mengunggah file ke server Transfer Family Anda. Setelah Anda menerima file terenkripsi, Anda dapat mendekripsi file-file tersebut dalam alur kerja Anda. Untuk tutorial detail, lihat Menyiapkan alur kerja terkelola untuk mendekripsi file.

Untuk menggunakan dekripsi PGP dalam alur kerja Anda

1. Identifikasi server Transfer Family untuk meng-host alur kerja Anda, atau buat yang baru. Anda harus memiliki ID server sebelum Anda dapat menyimpan kunci PGP Anda AWS Secrets Manager dengan nama rahasia yang benar.
2. Simpan kunci PGP Anda di AWS Secrets Manager bawah nama rahasia yang diperlukan. Untuk detailnya, lihat Kelola kunci PGP. Alur kerja dapat secara otomatis menemukan kunci PGP yang benar untuk digunakan untuk dekripsi berdasarkan nama rahasia di Secrets Manager.

   > **Note**
   >
   > Ketika Anda menyimpan rahasia di Secrets Manager, Anda Akun AWS dikenakan biaya. Untuk informasi tentang harga, lihat AWS Secrets Manager Harga.

3. Enkripsi file dengan menggunakan key pair PGP Anda. (Untuk daftar klien yang didukung, lihatKlien PGP yang didukung.) Jika Anda menggunakan baris perintah, jalankan perintah berikut. Untuk menggunakan perintah ini, ganti *username@example.com* dengan alamat email yang Anda gunakan untuk membuat key pair PGP. Ganti *testfile.txt* dengan nama file yang ingin Anda enkripsi.

   ```
   gpg -e -r username@example.com testfile.txt
   ```

4. Unggah file terenkripsi ke server Transfer Family Anda.
5. Konfigurasikan langkah dekripsi dalam alur kerja Anda. Untuk informasi selengkapnya, lihat Tambahkan langkah dekripsi.

Tambahkan langkah dekripsi

Langkah dekripsi mendekripsi file terenkripsi yang diunggah ke Amazon S3 atau Amazon EFS sebagai bagian dari alur kerja Anda. Untuk detail tentang mengonfigurasi dekripsi, lihat. Gunakan dekripsi PGP dalam alur kerja Anda

Saat Anda membuat langkah dekripsi untuk alur kerja, Anda harus menentukan tujuan untuk file yang didekripsi. Anda juga harus memilih apakah akan menimpa file yang ada jika file sudah ada di lokasi tujuan. Anda dapat memantau hasil alur kerja dekripsi dan mendapatkan log audit untuk setiap file secara real time menggunakan Amazon Logs. CloudWatch

Setelah Anda memilih jenis file Dekripsi untuk langkah Anda, halaman Konfigurasi parameter akan muncul. Isi nilai untuk bagian Konfigurasi parameter dekripsi PGP.

Opsi yang tersedia adalah sebagai berikut:

- Nama langkah - Masukkan nama deskriptif untuk langkah tersebut.
- Lokasi file — Dengan menentukan lokasi file, Anda dapat mendekripsi file yang digunakan pada langkah sebelumnya atau file asli yang diunggah.

> **Note**
>
> Parameter ini tidak tersedia jika langkah ini adalah langkah pertama dari alur kerja.

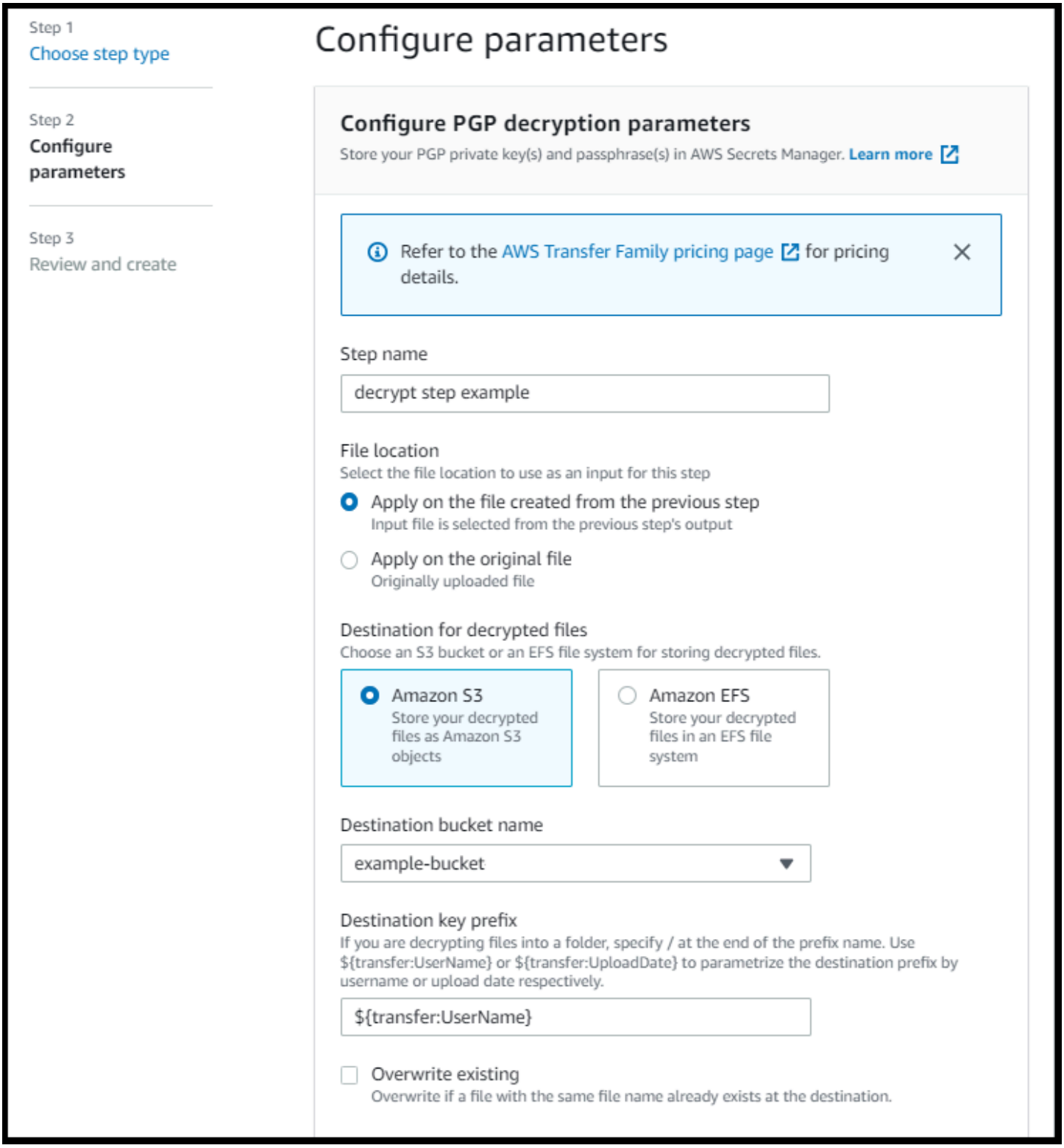
- Tujuan untuk file yang didekripsi — Pilih bucket Amazon S3 atau sistem file Amazon EFS sebagai tujuan untuk file yang didekripsi.
  - Jika memilih Amazon S3, Anda harus memberikan nama bucket tujuan dan awalan key tujuan. Untuk memparameterisasi awalan key tujuan dengan nama pengguna, **`${transfer:UserName}`** masukkan untuk Destination key prefix. Demikian pula, untuk parameterisasi awalan key tujuan dengan tanggal upload, **`${Transfer:UploadDate}`** masukkan untuk Destination key prefix.
  - Jika Anda memilih Amazon EFS, Anda harus menyediakan sistem dan jalur file tujuan.

> **Note**
>
> Opsi penyimpanan yang Anda pilih di sini harus sesuai dengan sistem penyimpanan yang digunakan oleh server Transfer Family yang terkait dengan alur kerja ini. Jika tidak, Anda akan menerima kesalahan saat mencoba menjalankan alur kerja ini.

- Timpa yang ada — Jika Anda mengunggah file, dan file dengan nama file yang sama sudah ada di tujuan, perilaku tergantung pada pengaturan untuk parameter ini:
  - Jika Timpa yang ada dipilih, file yang ada diganti dengan file yang sedang diproses.
  - Jika Timpa yang ada tidak dipilih, tidak ada yang terjadi, dan pemrosesan alur kerja berhenti.

> **Tip**
>
> Jika penulisan bersamaan dijalankan pada jalur file yang sama, hal itu dapat mengakibatkan perilaku yang tidak terduga saat menimpa file.

Tangkapan layar berikut menunjukkan contoh opsi yang mungkin Anda pilih untuk langkah dekripsi file Anda.



Izin IAM untuk langkah dekripsi

Agar langkah dekripsi berhasil, pastikan peran eksekusi untuk alur kerja Anda berisi izin berikut.

```
{
            "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

`s3:ListBucket`Izin hanya diperlukan jika Anda tidak memilih Timpa yang sudah ada. Izin ini memeriksa bucket Anda untuk melihat apakah file dengan nama yang sama sudah ada. Jika Anda telah memilih Timpa yang ada, alur kerja tidak perlu memeriksa file, dan hanya bisa menulisnya.
Jika file Amazon S3 Anda memiliki tag, Anda perlu menambahkan satu atau dua izin ke kebijakan IAM Anda.

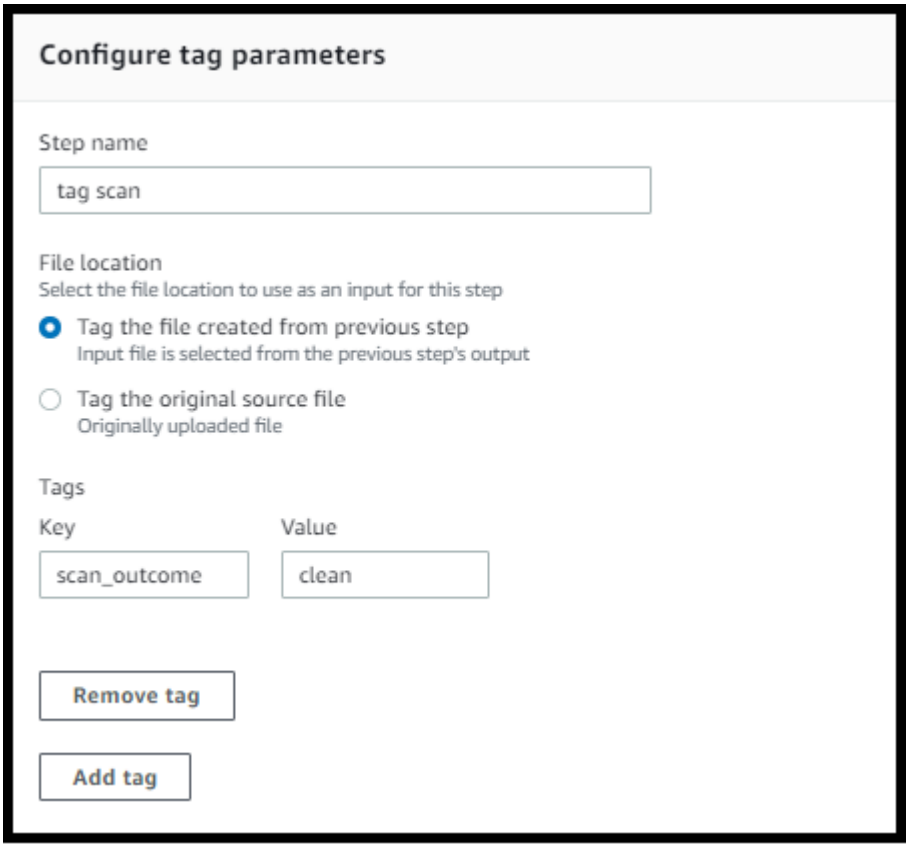
- Tambahkan `s3:GetObjectTagging` file Amazon S3 yang tidak berversi.

- Tambahkan `s3:GetObjectVersionTagging` untuk file Amazon S3 yang berversi.

Berkas tag

Untuk menandai file yang masuk untuk pemrosesan hilir lebih lanjut, gunakan langkah tag. Masukkan nilai tag yang ingin Anda tetapkan ke file yang masuk. Saat ini, operasi tag hanya didukung jika Anda menggunakan Amazon S3 untuk penyimpanan server Transfer Family Anda.

Berikut contoh langkah tag menetapkan `scan_outcome` dan `clean` sebagai kunci tag dan nilai, masing-masing.



Agar langkah tag berhasil, pastikan peran eksekusi untuk alur kerja Anda berisi izin berikut.

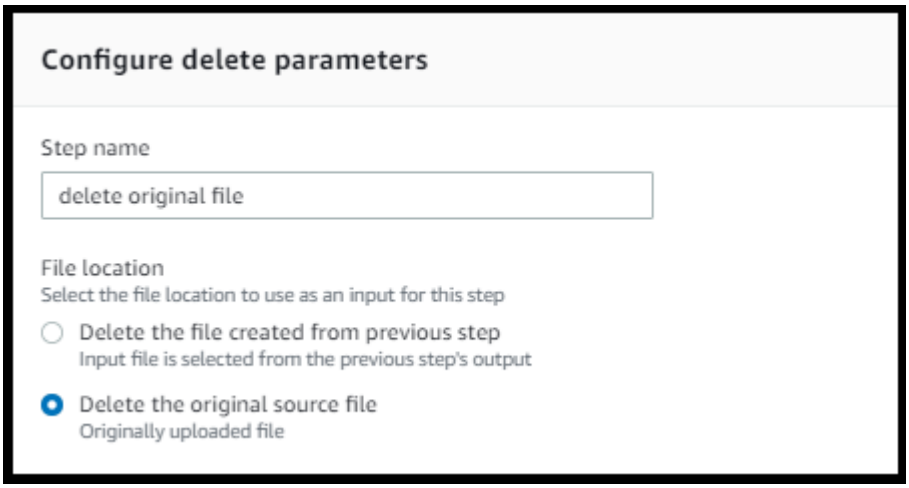
```
{
            "Sid": "Tag",
            "Effect": "Allow",
            "Action": [
                "s3:PutObjectTagging",
                "s3:PutObjectVersionTagging"
```

```
            ],
            "Resource": [
                "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
            ]
}
```

> **Note**
>
> Jika alur kerja Anda berisi langkah tag yang berjalan sebelum langkah salin atau dekripsi, Anda perlu menambahkan satu atau dua izin ke kebijakan IAM Anda.
>
> - Tambahkan `s3:GetObjectTagging` file Amazon S3 yang tidak berversi.
> - Tambahkan `s3:GetObjectVersionTagging` untuk file Amazon S3 yang berversi.

Hapus berkas

Untuk menghapus file yang diproses dari langkah alur kerja sebelumnya atau untuk menghapus file yang diunggah semula, gunakan langkah menghapus file.



Agar langkah penghapusan berhasil, pastikan peran eksekusi untuk alur kerja Anda berisi izin berikut.

```
{
            "Sid": "Delete",
            "Effect": "Allow",
            "Action": [
                "s3:DeleteObjectVersion",
```

```
                "s3:DeleteObject"
            ],
            "Resource": "arn:aws:secretsmanager:region:account-ID:secret:aws/transfer/
*"
        }
```

Variabel bernama untuk alur kerja

Untuk langkah menyalin dan mendekripsi, Anda dapat menggunakan variabel untuk melakukan tindakan secara dinamis. Saat ini, AWS Transfer Family mendukung variabel bernama berikut.

- Gunakan `${transfer:UserName}` untuk menyalin atau mendekripsi file ke tujuan berdasarkan pengguna yang mengunggah file.
- Gunakan `${transfer:UploadDate}` untuk menyalin atau mendekripsi file ke lokasi tujuan berdasarkan tanggal saat ini.

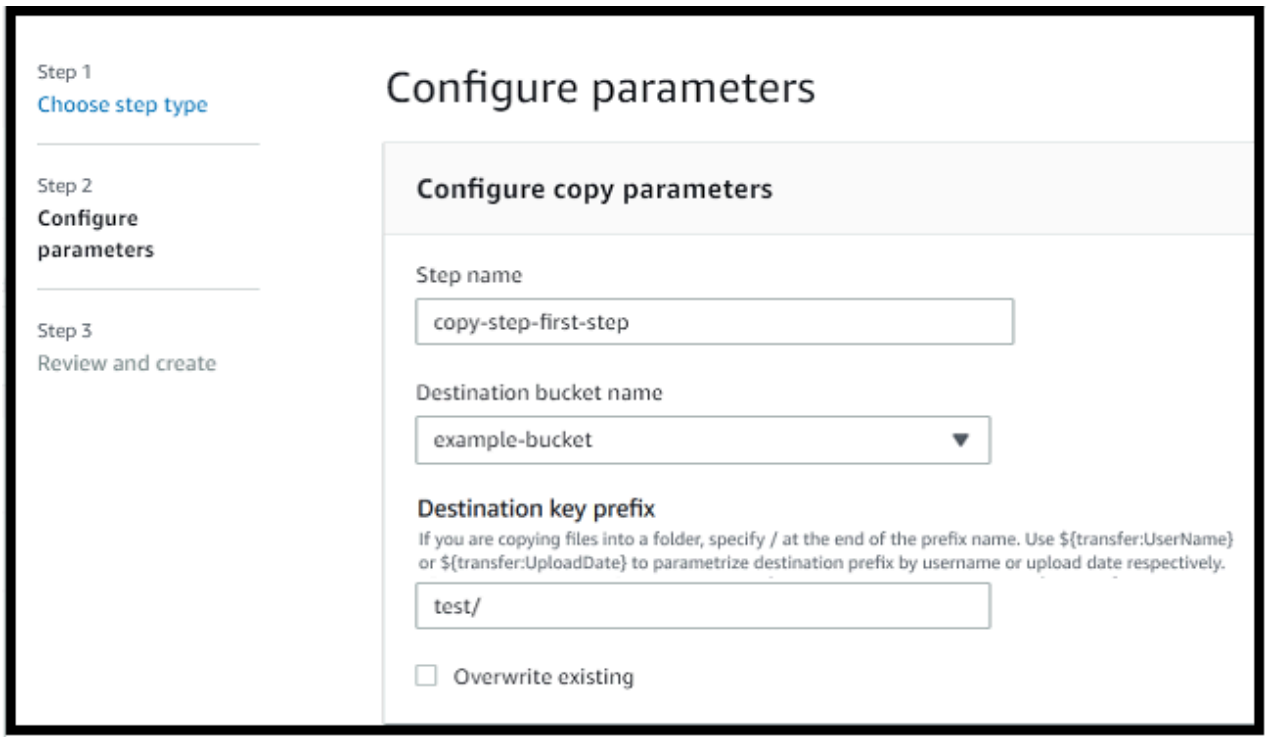
Contoh tag dan hapus alur kerja

Contoh berikut menggambarkan alur kerja yang menandai file masuk yang perlu diproses oleh aplikasi hilir, seperti platform analisis data. Setelah menandai file yang masuk, alur kerja kemudian menghapus file yang awalnya diunggah untuk menghemat biaya penyimpanan.

Console

Contoh tag dan pindahkan alur kerja

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi kiri, pilih Alur kerja.
3. Pada halaman Alur kerja, pilih Buat alur kerja.
4. Pada halaman Buat alur kerja, masukkan deskripsi. Deskripsi ini muncul di halaman Alur Kerja.
5. Tambahkan langkah pertama (salin).
    a. Di bagian Langkah nominal, pilih Tambah langkah.
    b. Pilih Salin file, lalu pilih Berikutnya.
    c. Masukkan nama langkah, lalu pilih bucket tujuan dan key prefix.



d. Pilih Berikutnya, lalu tinjau detail untuk langkahnya.

e. Pilih Buat langkah untuk menambahkan langkah dan melanjutkan.

6. Tambahkan langkah kedua (tag).

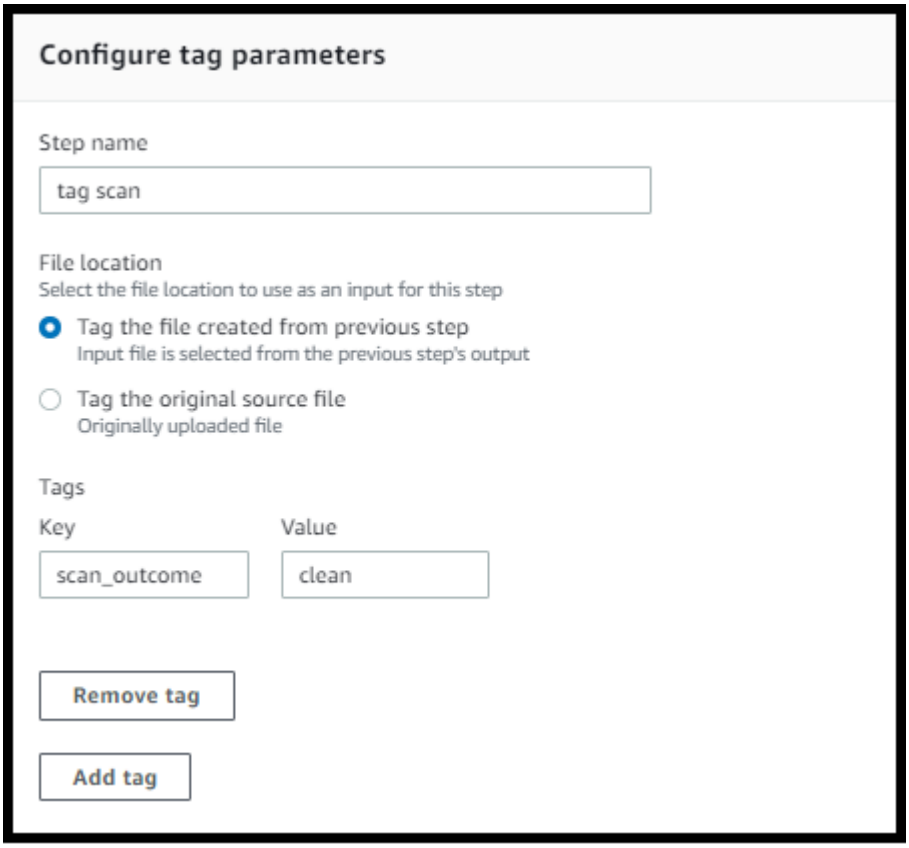
a. Di bagian Langkah nominal, pilih Tambah langkah.

b. Pilih File Tag, lalu pilih Berikutnya.

c. Masukkan nama langkah.

d. Untuk lokasi File, pilih Tag file yang dibuat dari langkah sebelumnya.

e. Masukkan Kunci dan Nilai.



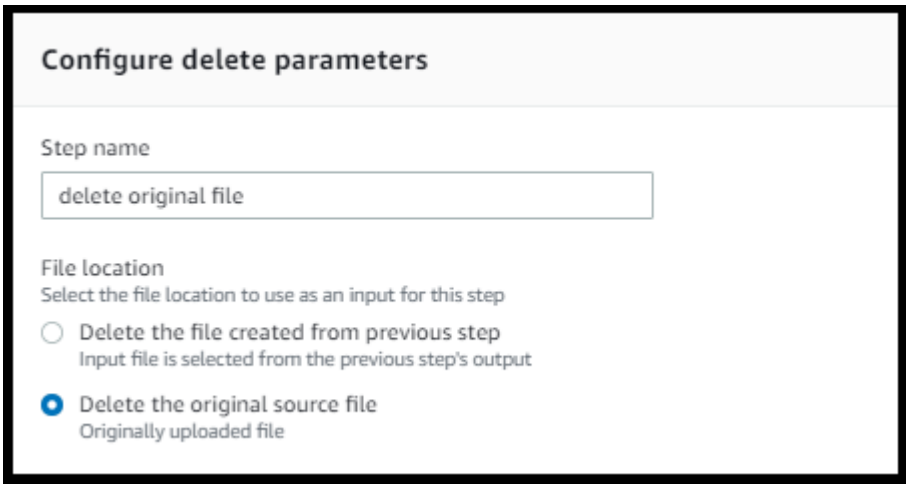
f. Pilih Berikutnya, lalu tinjau detail untuk langkahnya.

g. Pilih Buat langkah untuk menambahkan langkah dan melanjutkan.

7. Tambahkan langkah ketiga (hapus).

a. Di bagian Langkah nominal, pilih Tambah langkah.

b. Pilih Hapus file, lalu pilih Berikutnya.



c. Masukkan nama langkah.

d. Untuk lokasi File, pilih Hapus file sumber asli.

e. Pilih Berikutnya, lalu tinjau detail untuk langkahnya.

f. Pilih Buat langkah untuk menambahkan langkah dan melanjutkan.

8. Tinjau konfigurasi alur kerja, lalu pilih Buat alur kerja.

CLI

Contoh tag dan pindahkan alur kerja

1. Simpan kode berikut ke dalam file; misalnya,`tagAndMoveWorkflow.json`. Ganti masing-masing *`user input placeholder`* dengan informasi Anda sendiri.

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

Langkah pertama menyalin file yang diunggah ke lokasi Amazon S3 baru. Langkah kedua menambahkan tag (pasangan nilai kunci) ke file (`previous.file`) yang disalin ke lokasi baru. Dan, akhirnya, langkah ketiga menghapus file asli (`original.file`).

2. Buat alur kerja dari file yang disimpan. Ganti masing-masing *`user input placeholder`* dengan informasi Anda sendiri.

```
aws transfer create-workflow --description "short-description" --steps
 file://path-to-file --region region-ID
```

Sebagai contoh:

```
aws transfer create-workflow --description "copy-tag-delete workflow" --steps
 file://tagAndMoveWorkflow.json --region us-east-1
```

> **Note**
>
> Untuk detail selengkapnya tentang menggunakan file untuk memuat parameter, lihat Cara memuat parameter dari file.

3. Perbarui server yang ada.

> **Note**
>
> Langkah ini mengasumsikan Anda sudah memiliki server Transfer Family dan Anda ingin mengaitkan alur kerja dengannya. Jika belum, lihat Mengkonfigurasi titik akhir server SFTP, FTPS, atau FTP. Ganti masing-masing *`user input placeholder`* dengan informasi Anda sendiri.

```
aws transfer update-server --server-id server-ID --region region-ID
```

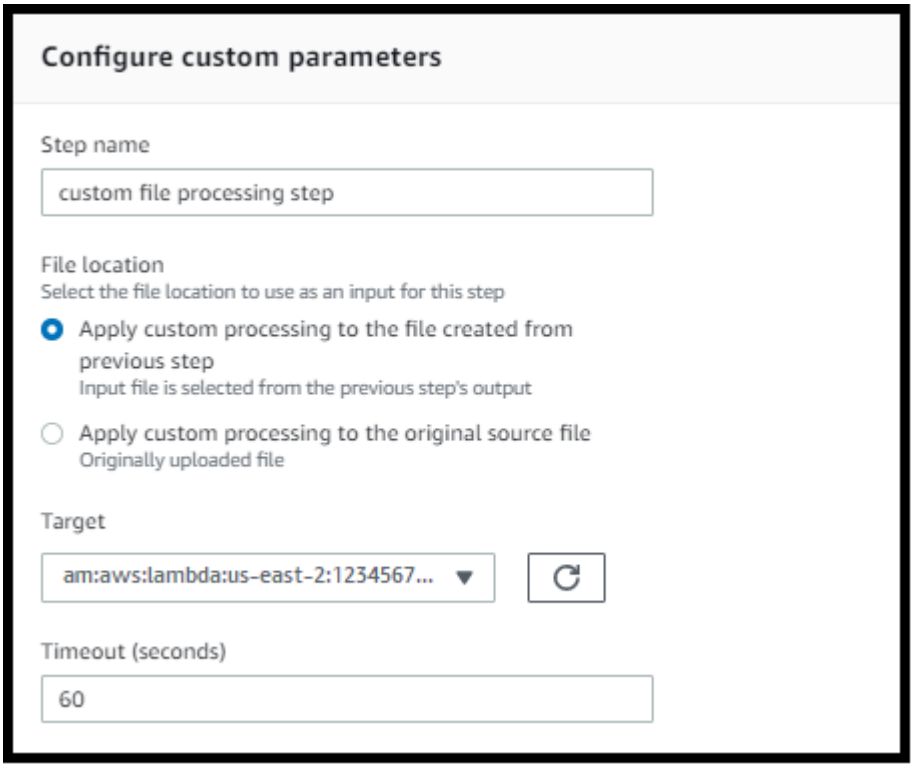
```
  --workflow-details '{"OnUpload":[{ "WorkflowId": "workflow-
ID","ExecutionRole": "execution-role-ARN"}]}'
```

Sebagai contoh:

```
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"}]}'
```

Gunakan langkah-langkah pemrosesan file khusus

Dengan menggunakan langkah pemrosesan file kustom, Anda dapat menggunakan logika pemrosesan file Anda Sendiri. AWS Lambda Setelah kedatangan file, server Transfer Family memanggil fungsi Lambda yang berisi logika pemrosesan file khusus, seperti mengenkripsi file, memindai malware, atau memeriksa jenis file yang salah. Dalam contoh berikut, AWS Lambda fungsi target digunakan untuk memproses file output dari langkah sebelumnya.



> **Note**
>
> Untuk contoh fungsi Lambda, lihat. Contoh fungsi Lambda untuk langkah alur kerja khusus Misalnya peristiwa (termasuk lokasi untuk file yang diteruskan ke Lambda), lihat. Contoh peristiwa dikirim ke AWS Lambda saat file upload

Dengan langkah alur kerja khusus, Anda harus mengonfigurasi fungsi Lambda untuk memanggil operasi API SendWorkflowStepState. `SendWorkflowStepState`memberitahukan eksekusi alur kerja bahwa langkah telah selesai dengan status sukses atau kegagalan. Status operasi `SendWorkflowStepState` API memanggil langkah penangan pengecualian atau langkah nominal dalam urutan linier, berdasarkan hasil fungsi Lambda.

Jika fungsi Lambda gagal atau habis waktu, langkahnya gagal, dan Anda lihat `StepErrored` di log Anda CloudWatch . Jika fungsi Lambda adalah bagian dari langkah nominal dan fungsi merespons `SendWorkflowStepState` dengan `Status="FAILURE"` atau waktu habis, aliran berlanjut dengan langkah-langkah penangan pengecualian. Dalam hal ini, alur kerja tidak terus mengeksekusi langkah-langkah nominal yang tersisa (jika ada). Untuk detail selengkapnya, lihat Penanganan pengecualian untuk alur kerja.

Ketika Anda memanggil operasi `SendWorkflowStepState` API, Anda harus mengirim parameter berikut:

```
{
    "ExecutionId": "string",
    "Status": "string",
    "Token": "string",
    "WorkflowId": "string"
}
```

Anda dapat mengekstrak`ExecutionId`,`Token`, dan `WorkflowId` dari peristiwa masukan yang diteruskan ketika fungsi Lambda dijalankan (contoh ditampilkan di bagian berikut). `Status`Nilainya bisa berupa `SUCCESS` atau`FAILURE`.

Untuk dapat memanggil operasi `SendWorkflowStepState` API dari fungsi Lambda, Anda harus menggunakan versi AWS SDK yang diterbitkan setelah Alur Kerja Terkelola diperkenalkan.

Menggunakan beberapa fungsi Lambda secara berurutan

Bila Anda menggunakan beberapa langkah kustom satu demi satu, opsi lokasi File bekerja secara berbeda daripada jika Anda hanya menggunakan satu langkah kustom. Transfer Family tidak mendukung meneruskan file yang diproses Lambda kembali untuk digunakan sebagai input langkah berikutnya. Jadi, jika Anda memiliki beberapa langkah khusus yang semuanya dikonfigurasi untuk menggunakan `previous.file` opsi, semuanya menggunakan lokasi file yang sama (lokasi file input untuk langkah kustom pertama).

> **Note**
>
> `previous.file`Pengaturan juga bekerja secara berbeda jika Anda memiliki langkah yang telah ditentukan (tag, salin, dekripsi, atau hapus) setelah langkah khusus. Jika langkah yang telah ditentukan dikonfigurasi untuk menggunakan `previous.file` pengaturan, langkah yang telah ditentukan menggunakan file input yang sama yang digunakan oleh langkah kustom. File yang diproses dari langkah kustom tidak diteruskan ke langkah yang telah ditentukan.

Mengakses file setelah pemrosesan kustom

Jika Anda menggunakan Amazon S3 sebagai penyimpanan, dan jika alur kerja menyertakan langkah khusus yang melakukan tindakan pada file yang diunggah semula, langkah selanjutnya tidak dapat mengakses file yang diproses tersebut. Artinya, langkah apa pun setelah langkah khusus tidak dapat mereferensikan file yang diperbarui dari output langkah khusus.

Misalnya, Anda memiliki tiga langkah berikut dalam alur kerja Anda.

- Langkah 1 - Unggah file bernama`example-file.txt`.
- Langkah 2 — Memanggil fungsi Lambda yang `example-file.txt` berubah dalam beberapa cara.
- Langkah 3 — Mencoba untuk melakukan pemrosesan lebih lanjut pada versi terbaru dari`example-file.txt`.

Jika Anda mengonfigurasi `sourceFileLocation` untuk Langkah 3`${original.file}`, Langkah 3 menggunakan lokasi file asli dari saat server mengunggah file ke penyimpanan di Langkah 1. Jika

Anda menggunakan `${previous.file}` untuk Langkah 3, Langkah 3 menggunakan kembali lokasi file yang Langkah 2 digunakan sebagai input.

Oleh karena itu, Langkah 3 menyebabkan kesalahan. Misalnya, jika langkah 3 mencoba menyalin yang diperbarui`example-file.txt`, Anda menerima kesalahan berikut:

```
{
    "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"
    },
```

Kesalahan ini terjadi karena langkah kustom memodifikasi tag entitas (ETag) `example-file.txt` agar tidak cocok dengan file aslinya.

> **Note**
>
> Perilaku ini tidak terjadi jika Anda menggunakan Amazon EFS karena Amazon EFS tidak menggunakan tag entitas untuk mengidentifikasi file.

Contoh peristiwa dikirim ke AWS Lambda saat file upload

Contoh berikut menunjukkan peristiwa yang dikirim ke AWS Lambda ketika file upload selesai. Salah satu contoh menggunakan server Transfer Family tempat domain dikonfigurasi dengan Amazon S3. Contoh lainnya menggunakan server Transfer Family di mana domain menggunakan Amazon EFS.

Custom step that uses an Amazon S3 domain

```
{
    "token": "MzI0Nzc4ZDktMGRmMi00MjFhLTgxMjUtYWZmZmRmODNkYjc0",
    "serviceMetadata": {
        "executionDetails": {
            "workflowId": "w-1234567890example",
            "executionId": "abcd1234-aa11-bb22-cc33-abcdef123456"
        },
        "transferDetails": {
```

```
            "sessionId": "36688ff5d2deda8c",
            "userName": "myuser",
            "serverId": "s-example1234567890"
        }
    },
    "fileLocation": {
        "domain": "S3",
        "bucket": "DOC-EXAMPLE-BUCKET",
        "key": "path/to/mykey",
        "eTag": "d8e8fca2dc0f896fd7cb4cb0031ba249",
        "versionId": null
    }
}
```

Custom step that uses an Amazon EFS domain

```
{
    "token": "MTg0N2Y3N2UtNWI5Ny00ZmZlLTk5YTgtZTU3YzViYjllNmZm",
    "serviceMetadata": {
        "executionDetails": {
            "workflowId": "w-1234567890example",
            "executionId": "abcd1234-aa11-bb22-cc33-abcdef123456"
        },
        "transferDetails": {
            "sessionId": "36688ff5d2deda8c",
            "userName": "myuser",
            "serverId": "s-example1234567890"
        }
    },
    "fileLocation": {
        "domain": "EFS",
        "fileSystemId": "fs-1234567",
        "path": "/path/to/myfile"
    }
}
```

Contoh fungsi Lambda untuk langkah alur kerja khusus

Fungsi Lambda berikut mengekstrak informasi mengenai status eksekusi, dan kemudian memanggil operasi SendWorkflowStepStateAPI untuk mengembalikan status ke alur kerja untuk langkah tersebut—baik atau. `SUCCESS` `FAILURE` Sebelum fungsi Anda memanggil operasi

`SendWorkflowStepState` API, Anda dapat mengonfigurasi Lambda untuk mengambil tindakan berdasarkan logika alur kerja Anda.

```
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)
    }
```

Izin IAM untuk langkah khusus

Agar langkah yang memanggil Lambda berhasil, pastikan peran eksekusi untuk alur kerja Anda berisi izin berikut.

```
{
            "Sid": "Custom",
            "Effect": "Allow",
            "Action": [
                "lambda:InvokeFunction"
            ],
            "Resource": [
                "arn:aws:lambda:region:account-id:function:function-name"
            ]
        }
```

Kebijakan IAM untuk alur kerja

Ketika Anda menambahkan alur kerja ke server, Anda harus memilih peran eksekusi. Server menggunakan peran ini ketika menjalankan alur kerja. Jika peran tidak memiliki izin yang tepat, AWS Transfer Family tidak dapat menjalankan alur kerja.

Bagian ini menjelaskan satu kemungkinan set izin AWS Identity and Access Management (IAM) yang dapat Anda gunakan untuk menjalankan alur kerja. Contoh lain dijelaskan nanti dalam topik ini.

> **Note**
>
> Jika file Amazon S3 Anda memiliki tag, Anda perlu menambahkan satu atau dua izin ke kebijakan IAM Anda.
>
> - Tambahkan `s3:GetObjectTagging` file Amazon S3 yang tidak berversi.
> - Tambahkan `s3:GetObjectVersionTagging` untuk file Amazon S3 yang berversi.

Untuk membuat peran eksekusi untuk alur kerja Anda

1. Buat peran IAM baru, dan tambahkan kebijakan AWS terkelola `AWSTransferFullAccess` ke peran tersebut. Untuk informasi selengkapnya tentang membuat peran IAM baru, lihatthe section called “Buat peran dan kebijakan IAM”.
2. Buat kebijakan lain dengan izin berikut, dan lampirkan ke peran Anda. Ganti masing-masing *`user input placeholder`* dengan informasi Anda sendiri.

```
{
    "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. Simpan peran ini dan tentukan sebagai peran eksekusi saat Anda menambahkan alur kerja ke server.

> **Note**
>
> Ketika Anda membangun peran IAM, AWS merekomendasikan agar Anda membatasi akses ke sumber daya Anda sebanyak mungkin untuk alur kerja Anda.

Hubungan kepercayaan alur kerja

Peran eksekusi alur kerja juga membutuhkan hubungan kepercayaan dengan`transfer.amazonaws.com`. Untuk membangun hubungan kepercayaan AWS Transfer Family, lihatUntuk membangun hubungan kepercayaan.

Saat Anda membangun hubungan kepercayaan Anda, Anda juga dapat mengambil langkah-langkah untuk menghindari masalah wakil yang membingungkan. Untuk deskripsi masalah ini, serta contoh cara menghindarinya, lihatthe section called “Pencegahan confused deputy lintas layanan”.

Contoh peran eksekusi: Dekripsi, salin, dan tag

Jika Anda memiliki alur kerja yang menyertakan langkah penandaan, penyalinan, dan dekripsi, Anda dapat menggunakan kebijakan IAM berikut. Ganti masing-masing *`user input placeholder`* dengan informasi Anda sendiri.

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

Contoh peran eksekusi: Jalankan fungsi dan hapus

Dalam contoh ini, Anda memiliki alur kerja yang memanggil fungsi. AWS Lambda Jika alur kerja menghapus file yang diunggah dan memiliki langkah penangan pengecualian untuk menindaklanjuti eksekusi alur kerja yang gagal pada langkah sebelumnya, gunakan kebijakan IAM berikut. Ganti masing-masing *user input placeholder* dengan informasi Anda sendiri.

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

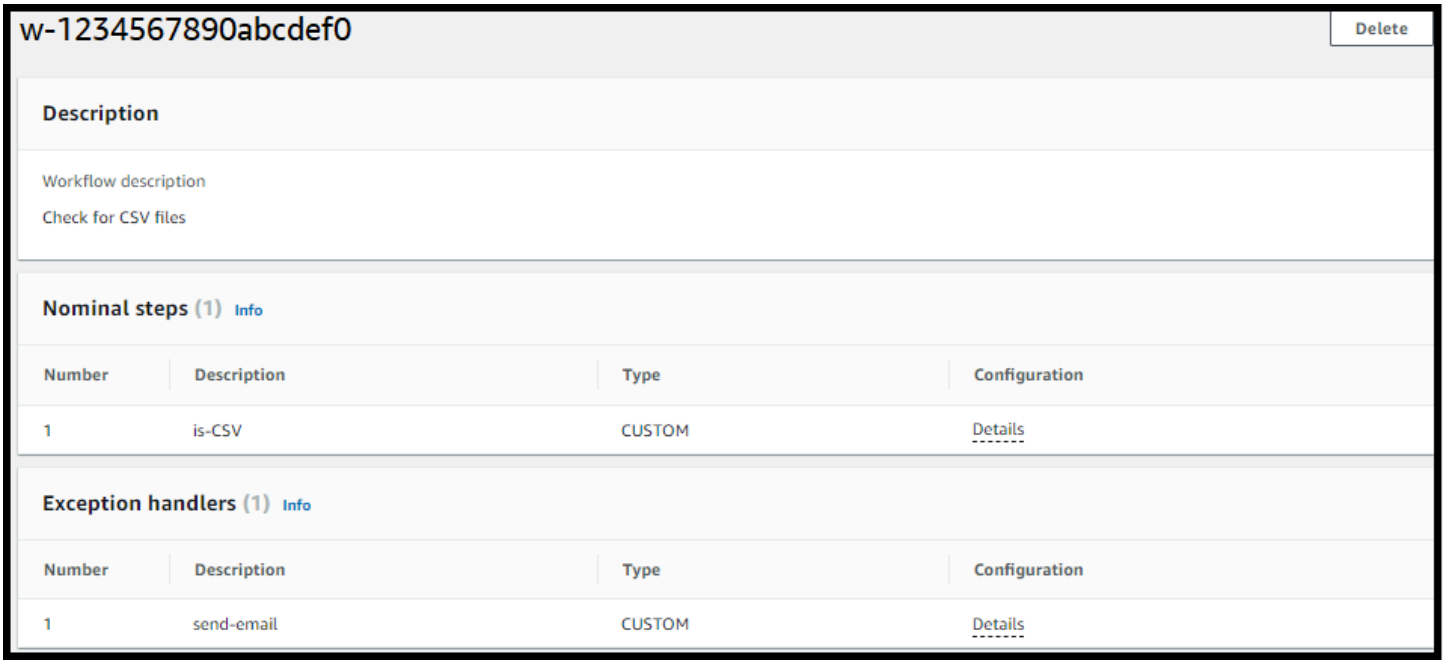
Penanganan pengecualian untuk alur kerja

Jika terjadi kesalahan selama eksekusi alur kerja, langkah-langkah penanganan pengecualian yang Anda tentukan akan dijalankan. Anda menentukan langkah penanganan kesalahan untuk alur kerja dengan cara yang sama seperti Anda menentukan langkah nominal untuk alur kerja. Misalnya, misalkan Anda telah mengonfigurasi pemrosesan kustom dalam langkah-langkah nominal untuk memvalidasi file yang masuk. Jika validasi file gagal, langkah penanganan pengecualian dapat mengirim email ke administrator.

Contoh alur kerja berikut berisi dua langkah:

- Satu langkah nominal yang memeriksa apakah file yang diunggah dalam format CSV
- Langkah penanganan pengecualian yang mengirim email jika file yang diunggah tidak dalam format CSV, dan langkah nominal gagal

Untuk memulai langkah penanganan pengecualian, AWS Lambda fungsi dalam langkah nominal harus merespons dengan. `Status="FAILURE"` Untuk informasi selengkapnya tentang penanganan kesalahan dalam alur kerja, lihatthe section called “Gunakan langkah-langkah pemrosesan file khusus”.



Pantau eksekusi alur kerja

Amazon CloudWatch memantau AWS sumber daya Anda dan aplikasi yang Anda jalankan AWS Cloud secara real time. Anda dapat menggunakan Amazon CloudWatch untuk mengumpulkan dan melacak metrik, yang merupakan variabel yang dapat Anda ukur untuk alur kerja Anda. Anda dapat melihat metrik alur kerja dan log konsolidasi menggunakan Amazon. CloudWatch

CloudWatch logging untuk alur kerja

CloudWatch menyediakan audit dan pencatatan terkonsolidasi untuk kemajuan dan hasil alur kerja.

Lihat CloudWatch log Amazon untuk alur kerja

1. Buka CloudWatch konsol Amazon di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi kiri, pilih Log, lalu pilih Grup log.
3. Pada halaman Grup log, pada bilah navigasi, pilih Wilayah yang benar untuk AWS Transfer Family server Anda.
4. Pilih grup log yang sesuai dengan server Anda.

   Misalnya, jika ID server Anda`s-1234567890abcdef0`, grup log Anda adalah`/aws/transfer/s-1234567890abcdef0`.

5. Pada halaman detail grup log untuk server Anda, aliran log terbaru ditampilkan. Ada dua aliran log untuk pengguna yang Anda jelajahi:

   - Satu untuk setiap sesi Secure Shell (SSH) File Transfer Protocol (SFTP).
   - Satu untuk alur kerja yang sedang dijalankan untuk server Anda. Format untuk aliran log untuk alur kerja adalah*username*.*workflowID*.*uniqueStreamSuffix*.

   Misalnya, jika pengguna Anda`mary-major`, Anda memiliki aliran log berikut:

   ```
   mary-major-east.1234567890abcdef0
   mary.w-abcdef01234567890.021345abcdef6789
   ```

   > **Note**
   >
   > Pengidentifikasi alfanumerik 16 digit yang tercantum dalam contoh ini adalah fiktif. Nilai yang Anda lihat di Amazon CloudWatch berbeda.

Halaman peristiwa Log untuk `mary-major-usa-east.1234567890abcdef0` menampilkan detail untuk setiap sesi pengguna, dan aliran `mary.w-abcdef01234567890.021345abcdef6789` log berisi detail untuk alur kerja.

Berikut ini adalah contoh aliran log untuk`mary.w-abcdef01234567890.021345abcdef6789`, berdasarkan alur kerja (`w-abcdef01234567890`) yang berisi langkah salin.

```
{
    "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 metrik untuk alur kerja

AWS Transfer Family menyediakan beberapa metrik untuk alur kerja. Anda dapat melihat metrik berapa banyak eksekusi alur kerja yang dimulai, diselesaikan dengan sukses, dan gagal pada menit sebelumnya. Semua CloudWatch metrik untuk Transfer Family dijelaskan dalamMenggunakan CloudWatch metrik untuk Transfer Family.

Buat alur kerja dari template

Anda dapat menerapkan AWS CloudFormation tumpukan yang membuat alur kerja dan server dari template. Prosedur ini berisi contoh yang dapat Anda gunakan untuk menyebarkan alur kerja dengan cepat.

Untuk membuat AWS CloudFormation tumpukan yang membuat AWS Transfer Family alur kerja dan server

1. Buka AWS CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
2. Simpan kode berikut ke file.

   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. Ganti item berikut dengan nilai aktual Anda.

   - Ganti *workflow-execution-role-arn*dengan ARN untuk peran eksekusi alur kerja yang sebenarnya. Misalnya, `arn:aws:transfer:us-east-2:111122223333:workflow/w-1234567890abcdef0`
   - Ganti `arn:aws:lambda:`*region*`:`*account-id*`:function:`*function-name* dengan ARN untuk fungsi Lambda Anda. Misalnya, `arn:aws:lambda:us-east-2:123456789012:function:example-lambda-idp`.

4. Ikuti petunjuk untuk menerapkan AWS CloudFormation tumpukan dari template yang ada di Memilih template tumpukan di Panduan AWS CloudFormation Pengguna.

Setelah tumpukan digunakan, Anda dapat melihat detailnya di tab Output di CloudFormation konsol. Template membuat server AWS Transfer Family SFTP baru yang dikelola pengguna layanan pengguna, dan alur kerja baru, dan mengaitkan alur kerja dengan server baru.

Menghapus alur kerja dari server Transfer Family

Jika Anda telah mengaitkan alur kerja dengan server Transfer Family, dan sekarang Anda ingin menghapus asosiasi tersebut, Anda dapat melakukannya dengan menggunakan konsol atau secara terprogram.

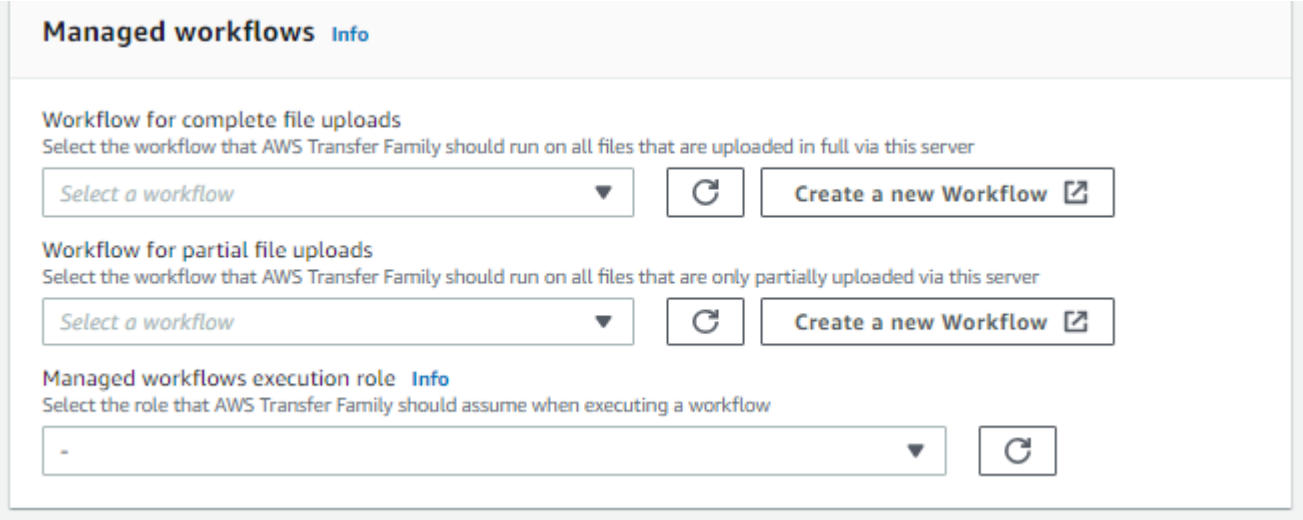
Console

Untuk menghapus alur kerja dari server Transfer Family

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi kiri, pilih Server.
3. Pilih pengenal untuk server di kolom ID Server.
4. Pada halaman detail untuk server, gulir ke bawah ke bagian Detail tambahan, lalu pilih Edit.
5. Pada halaman Edit detail tambahan, di bagian Alur kerja terkelola, kosongkan informasi untuk semua pengaturan:
   - Pilih tanda hubung (-) dari daftar alur kerja untuk Alur Kerja untuk upload file lengkap.
   - Jika belum dihapus, pilih tanda hubung (-) dari daftar alur kerja untuk Alur Kerja untuk unggahan file sebagian.
   - Pilih tanda hubung (-) dari daftar peran untuk peran eksekusi alur kerja terkelola.

   Jika Anda tidak melihat tanda hubung, gulir ke atas hingga Anda melihatnya, karena ini adalah nilai pertama di setiap menu.

   Layar akan terlihat seperti berikut ini.



6. Gulir ke bawah dan pilih Simpan untuk menyimpan perubahan Anda.

CLI

Anda menggunakan panggilan `update-server` (atau `UpdateServer` untuk API), dan memberikan argumen kosong untuk `OnUpload` dan `OnPartialUpload` parameter.

Dari AWS CLI, jalankan perintah berikut:

```
aws transfer update-server --server-id your-server-id --workflow-details
 '{"OnPartialUpload":[],"OnUpload":[]}'
```

Ganti *`your-server-id`* dengan ID untuk server Anda. Misalnya, jika ID server Andas`-01234567890abcdef`, perintahnya adalah sebagai berikut:

```
aws transfer update-server --server-id s-01234567890abcdef --workflow-details
 '{"OnPartialUpload":[],"OnUpload":[]}'
```

Pembatasan dan batasan alur kerja yang dikelola

Pembatasan

Pembatasan berikut saat ini berlaku untuk alur kerja pemrosesan pasca-unggah untuk. AWS Transfer Family

- AWS Lambda Fungsi lintas akun dan lintas wilayah tidak didukung. Namun, Anda dapat menyalin seluruh akun, asalkan kebijakan AWS Identity and Access Management (IAM) Anda dikonfigurasi dengan benar.
- Untuk semua langkah alur kerja, setiap bucket Amazon S3 yang diakses oleh alur kerja harus berada di wilayah yang sama dengan alur kerja itu sendiri.
- Untuk langkah dekripsi, tujuan dekripsi harus cocok dengan sumber untuk Wilayah dan penyimpanan cadangan (misalnya, jika file yang akan didekripsi disimpan di Amazon S3, maka tujuan yang ditentukan juga harus di Amazon S3).
- Hanya langkah kustom asinkron yang didukung.
- Batas waktu langkah khusus adalah perkiraan. Artinya, mungkin perlu waktu sedikit lebih lama dari yang ditentukan. Selain itu, alur kerja tergantung pada fungsi Lambda. Oleh karena itu, jika fungsi tertunda selama eksekusi, alur kerja tidak menyadari penundaan.
- Jika Anda melebihi batas pembatasan, Transfer Family tidak menambahkan operasi alur kerja ke antrian.
- Alur kerja tidak dimulai untuk file yang memiliki ukuran 0. File dengan ukuran lebih besar dari 0 melakukan memulai alur kerja terkait.

Batasan

Selain itu, batasan fungsional berikut berlaku untuk alur kerja untuk Transfer Family:

- Jumlah alur kerja per Wilayah, per akun, dibatasi hingga 10.
- Batas waktu maksimum untuk langkah-langkah khusus adalah 30 menit.
- Jumlah maksimum langkah dalam alur kerja adalah 8.
- Jumlah maksimum tag per alur kerja adalah 50.
- Jumlah maksimum eksekusi bersamaan yang berisi langkah dekripsi adalah 250 per alur kerja.
- Anda dapat menyimpan maksimal 3 kunci pribadi PGP, per server Transfer Family, per pengguna.
- Ukuran maksimum untuk file yang didekripsi adalah 10 GB.
- Kami membatasi tingkat eksekusi baru menggunakan sistem token bucket dengan kapasitas burst 100 dan tingkat isi ulang 1.
- Setiap kali Anda menghapus alur kerja dari server dan menggantinya dengan yang baru, atau memperbarui konfigurasi server (yang memengaruhi peran eksekusi alur kerja), Anda harus menunggu sekitar 10 menit sebelum menjalankan alur kerja baru. Server Transfer Family

menyimpan cache detail alur kerja, dan dibutuhkan waktu 10 menit bagi server untuk menyegarkan cache-nya.

Selain itu, Anda harus keluar dari sesi SFTP aktif apa pun, dan kemudian masuk kembali setelah masa tunggu 10 menit untuk melihat perubahannya.

Mengelola server

Di bagian ini, Anda dapat menemukan informasi tentang cara melihat daftar server Anda, cara melihat detail server Anda, cara mengedit detail server Anda, dan cara mengubah kunci host untuk server berkemampuan SFTP Anda.

Topik

- Lihat daftar server
- Hapus server
- Lihat detail server SFTP, FTPS, dan FTP
- Lihat detail server AS2
- Edit detail server
- Kelola kunci host untuk server berkemampuan SFTP
- Memantau penggunaan di konsol

Lihat daftar server

Di AWS Transfer Family konsol, Anda dapat menemukan daftar semua server Anda yang terletak di AWS Wilayah yang Anda pilih.

Untuk menemukan daftar server Anda yang ada di suatu AWS Wilayah

- Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.

  Jika Anda memiliki satu atau beberapa server di AWS Wilayah saat ini, konsol terbuka untuk menampilkan daftar server Anda. Jika Anda tidak melihat daftar server, pastikan Anda berada di Wilayah yang benar. Anda juga dapat memilih Server dari panel navigasi.

  Untuk informasi selengkapnya tentang melihat detail server Anda, lihatLihat detail server SFTP, FTPS, dan FTP.

Hapus server

Prosedur ini menjelaskan cara menghapus server Transfer Family dengan menggunakan AWS Transfer Family konsol atau AWS CLI.

> **Important**
>
> Anda ditagih, untuk setiap protokol yang diaktifkan untuk mengakses titik akhir Anda, hingga Anda menghapus server.

> **Warning**
>
> Menghapus server mengakibatkan semua penggunanya dihapus. Data dalam bucket yang diakses dengan menggunakan server tidak dihapus, dan tetap dapat diakses oleh AWS pengguna yang memiliki hak istimewa untuk bucket Amazon S3 tersebut.

Console

Untuk menghapus server dengan menggunakan konsol

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi kiri, pilih Server.
3. Pilih kotak centang server yang ingin Anda hapus.
4. Untuk Tindakan, pilih Hapus.
5. Di kotak dialog konfirmasi yang muncul, masukkan kata**`delete`**, lalu pilih Hapus untuk mengonfirmasi bahwa Anda ingin menghapus server.

Server dihapus dari halaman Server dan Anda tidak lagi ditagih untuk itu.

AWS CLI

Untuk menghapus server dengan menggunakan CLI

1. (Opsional) Jalankan perintah berikut untuk melihat detail server yang ingin Anda hapus secara permanen.

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

   `describe-server`Perintah ini mengembalikan semua detail untuk server Anda.
2. Jalankan perintah berikut untuk menghapus server.

```
aws transfer delete-server --server-id your-server-id
```

Jika berhasil, perintah menghapus server dan tidak mengembalikan informasi apa pun.

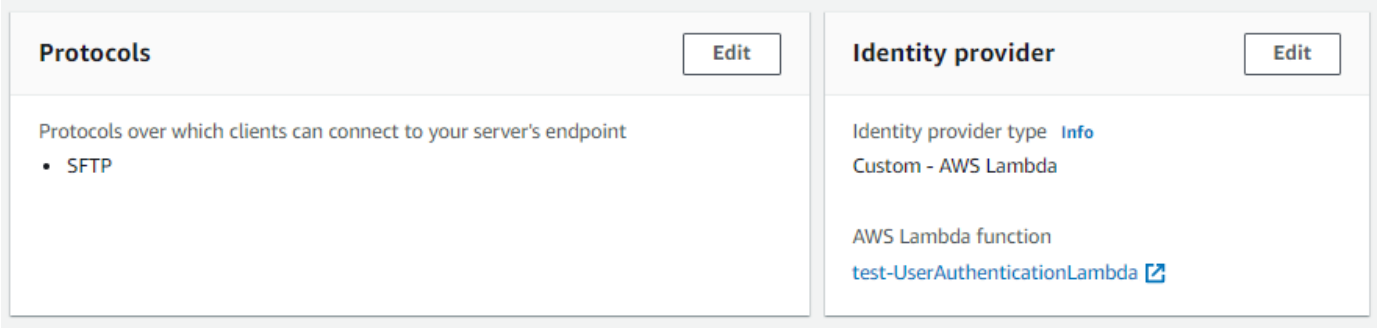
Lihat detail server SFTP, FTPS, dan FTP

Anda dapat menemukan daftar detail dan properti untuk AWS Transfer Family server individual. Properti server termasuk protokol, penyedia identitas, status, jenis titik akhir, nama host khusus, titik akhir, pengguna, peran logging, kunci host server, dan tag.

Untuk melihat detail server

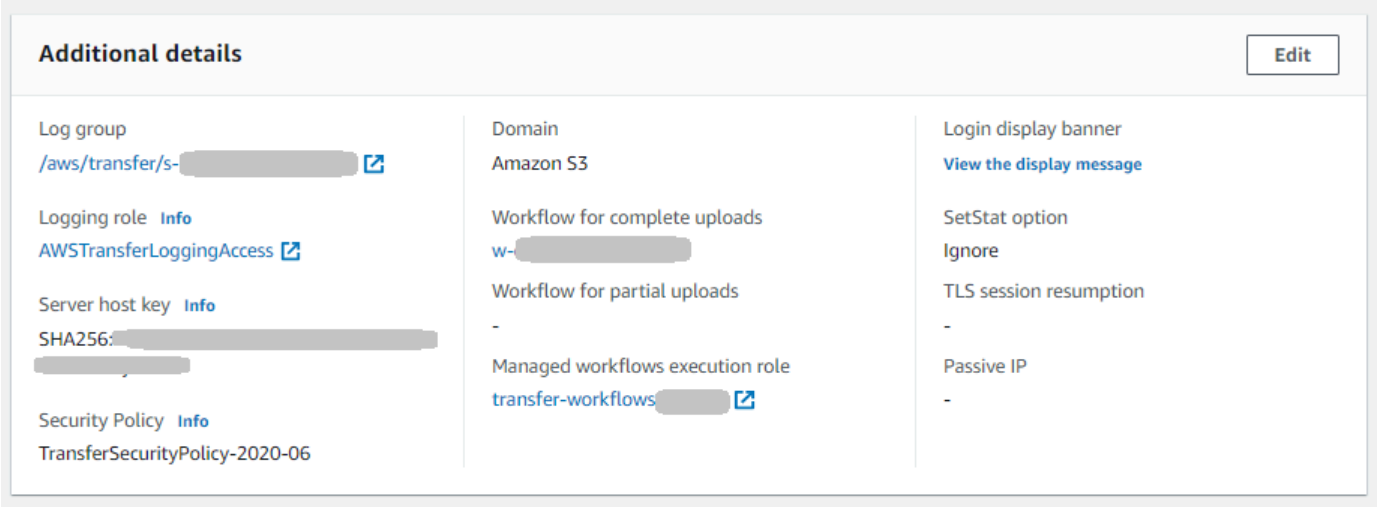
1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi, pilih Server.
3. Pilih pengenal di kolom ID Server untuk melihat halaman detail Server, ditampilkan berikut.

   Anda dapat mengubah properti server di halaman ini dengan memilih Edit. Untuk informasi selengkapnya tentang mengedit detail server, lihatEdit detail server. Halaman detail untuk server AS2 sedikit berbeda. Untuk server AS2, lihatLihat detail server AS2.



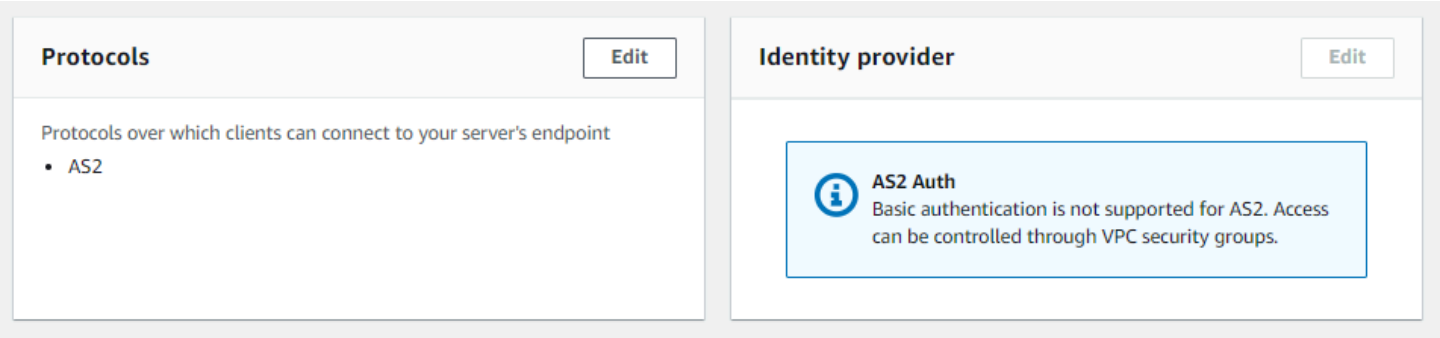
> **Note**
>
> Kunci host server Deskripsi dan Nilai impor Tanggal baru per September 2022. Nilai-nilai ini diperkenalkan untuk mendukung fitur beberapa kunci host. Fitur ini memerlukan migrasi dari setiap kunci host tunggal yang digunakan sebelum pengenalan beberapa kunci host.
> Nilai tanggal yang diimpor untuk kunci host server yang dimigrasi diatur ke tanggal modifikasi terakhir untuk server. Artinya, tanggal yang Anda lihat untuk kunci host yang

dimigrasi sesuai dengan tanggal terakhir Anda memodifikasi server dengan cara apa pun, sebelum migrasi kunci host server.
Satu-satunya kunci yang dimigrasikan adalah kunci host server tertua atau satu-satunya. Setiap kunci tambahan memiliki tanggal aktualnya sejak Anda mengimpornya. Selain itu, kunci yang dimigrasi memiliki deskripsi yang membuatnya mudah untuk mengidentifikasinya sebagai telah dimigrasi.
Migrasi terjadi antara 2 September dan 13 September. Tanggal migrasi aktual dalam rentang ini tergantung pada Wilayah server Anda.

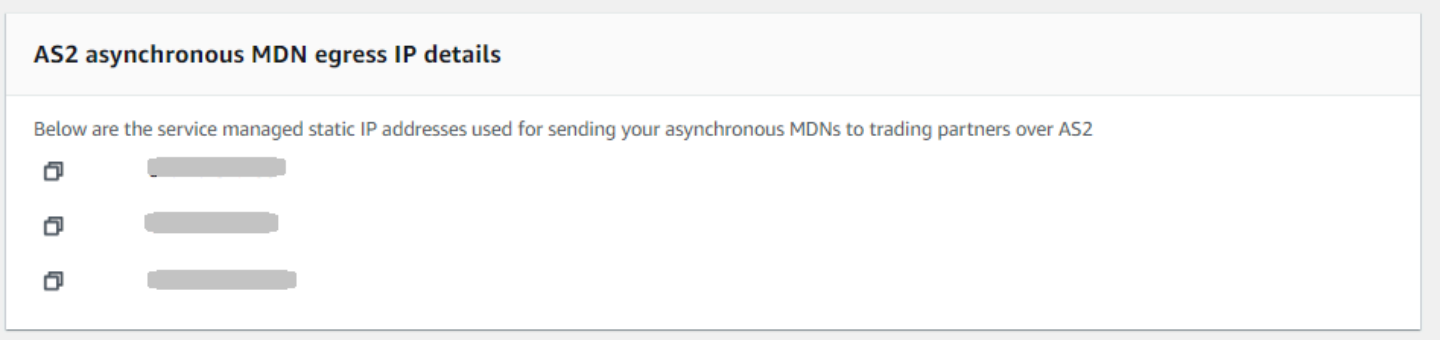


Lihat detail server AS2

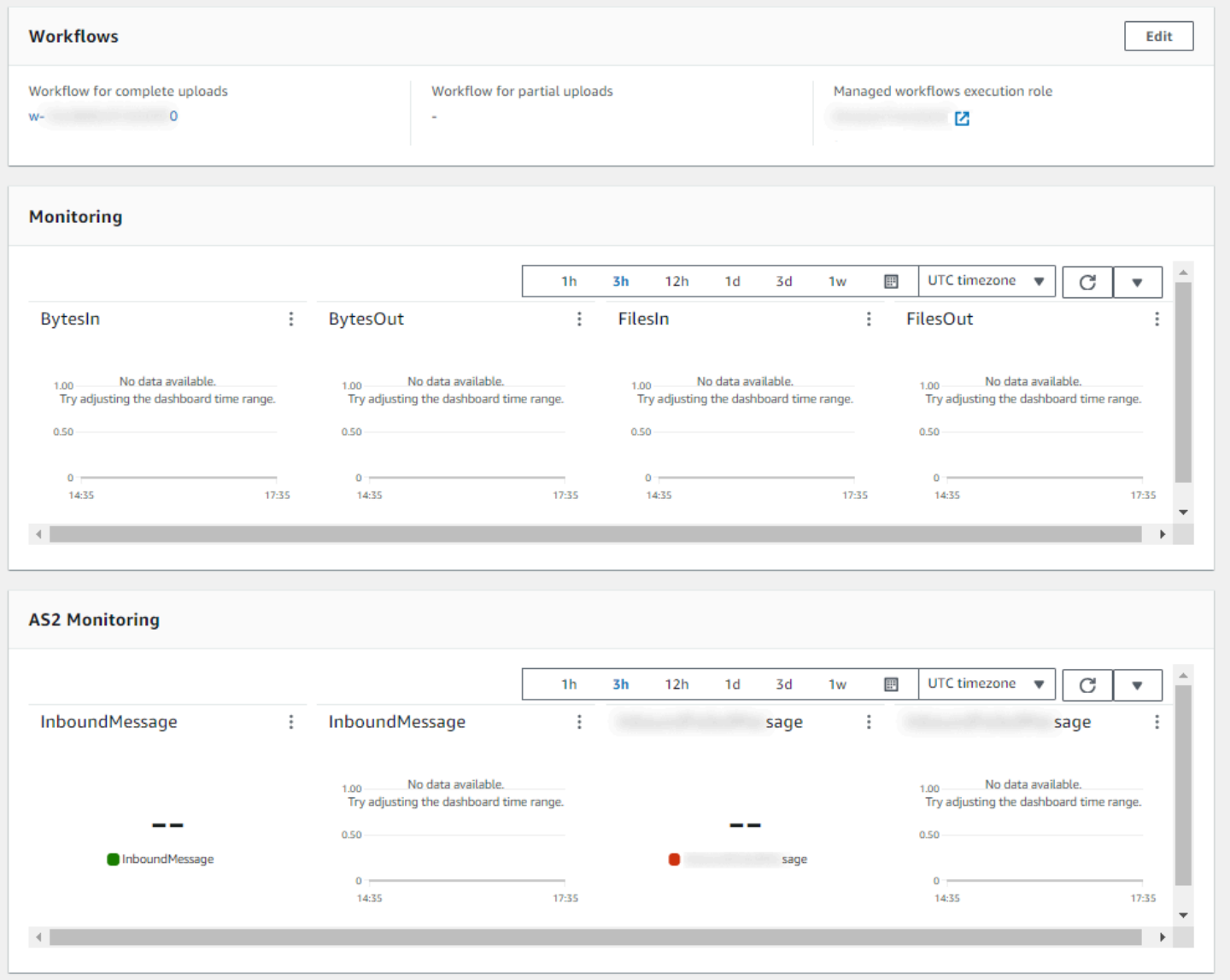
Anda dapat menemukan daftar detail dan properti untuk AWS Transfer Family server individual. Properti server termasuk protokol, status, dan banyak lagi. Untuk server AS2, Anda juga dapat melihat alamat IP keluar MDN asinkron AS2.



Setiap server AS2 diberi tiga alamat IP statis. Gunakan alamat IP ini untuk mengirim mDNS asinkron ke mitra dagang Anda melalui AS2.



Bagian bawah halaman detail server AS2 berisi detail untuk alur kerja terlampir dan informasi pemantauan dan penandaan.



Edit detail server

Setelah Anda membuat AWS Transfer Family server, Anda dapat mengedit konfigurasi server.

Topik

- Edit protokol transfer file
- Edit parameter penyedia identitas khusus
- Edit titik akhir server
- Edit konfigurasi logging Anda
- Edit kebijakan keamanan
- Mengubah alur kerja terkelola untuk server Anda
- Ubah spanduk tampilan untuk server Anda
- Menempatkan server Anda secara online atau offline

Untuk mengedit konfigurasi server

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi kiri, pilih Server.
3. Pilih pengenal di kolom ID Server untuk melihat halaman detail Server, ditampilkan berikut.

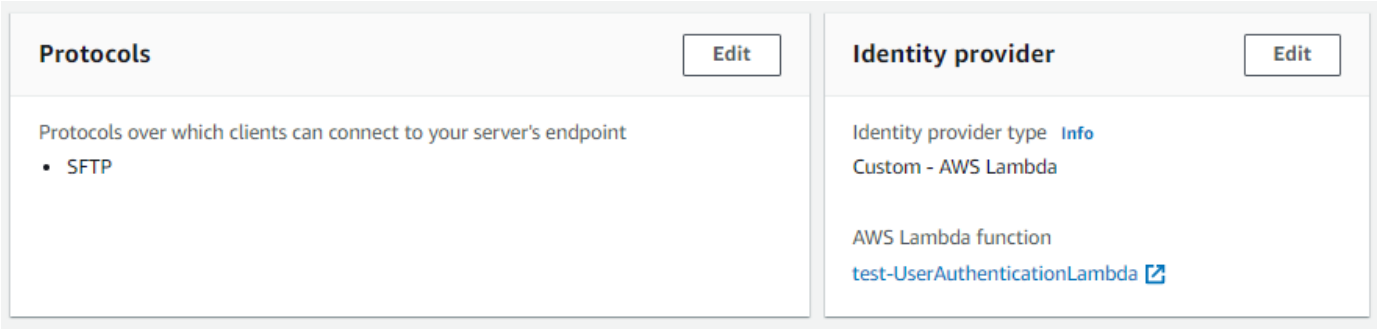
   Anda dapat mengubah properti server di halaman ini dengan memilih Edit:

   - Untuk mengubah protokol, lihat. Edit protokol transfer file
   - Untuk penyedia identitas, perhatikan bahwa Anda tidak dapat mengubah jenis penyedia identitas server setelah membuat server. Untuk mengubah penyedia identitas, hapus server dan buat yang baru dengan penyedia identitas yang Anda inginkan.

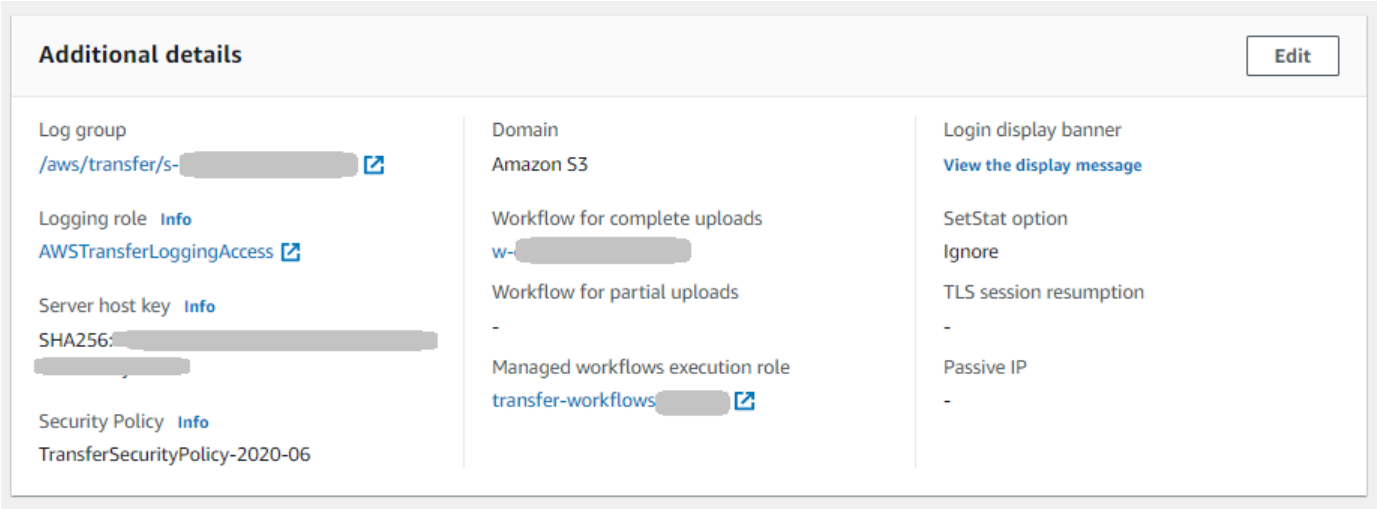
     > **Note**
     >
     > Jika server Anda menggunakan penyedia identitas khusus, Anda dapat mengedit beberapa properti. Untuk detailnya, lihat Edit parameter penyedia identitas khusus.

   - Untuk mengubah jenis titik akhir atau nama host kustom, lihat. Edit titik akhir server
   - Untuk menambahkan perjanjian, Anda harus terlebih dahulu menambahkan AS2 sebagai protokol ke server Anda. Untuk detailnya, lihat Edit protokol transfer file.

- Untuk mengelola kunci host untuk server Anda, lihatKelola kunci host untuk server berkemampuan SFTP.
- Di bawah Detail tambahan, Anda dapat mengedit informasi berikut:
  - Untuk mengubah peran logging, lihatEdit konfigurasi logging Anda.
  - Untuk mengubah kebijakan keamanan, lihatEdit kebijakan keamanan.
  - Untuk mengubah kunci host server, lihatKelola kunci host untuk server berkemampuan SFTP.
  - Untuk mengubah alur kerja terkelola untuk server Anda, lihatMengubah alur kerja terkelola untuk server Anda.
  - Untuk mengedit spanduk tampilan untuk server Anda, lihatUbah spanduk tampilan untuk server Anda.
- Di bawah Konfigurasi tambahan, Anda dapat mengedit informasi berikut:
  - SetStat opsi: aktifkan opsi ini untuk mengabaikan kesalahan yang dihasilkan saat klien mencoba menggunakan `SETSTAT` pada file yang Anda unggah ke bucket Amazon S3. Untuk detail tambahan, lihat `SetStatOption` dokumentasi dalam ProtocolDetailstopik.
  - Dimulainya kembali sesi TLS: menyediakan mekanisme untuk melanjutkan atau berbagi kunci rahasia yang dinegosiasikan antara kontrol dan koneksi data untuk sesi FTPS. Untuk detail tambahan, lihat `TlsSessionResumptionMode` dokumentasi dalam ProtocolDetailstopik.
  - IP pasif: menunjukkan mode pasif, untuk protokol FTP dan FTPS. Masukkan satu alamat IPv4, seperti alamat IP publik firewall, router, atau penyeimbang beban. Untuk detail tambahan, lihat `PassiveIp` dokumentasi dalam ProtocolDetailstopik.
- Untuk memulai atau menghentikan server Anda, lihatMenempatkan server Anda secara online atau offline.
- Untuk menghapus server, lihatHapus server.
- Untuk mengedit properti pengguna, lihatMengelola kontrol akses.



> **Note**
>
> Kunci host server Deskripsi dan Nilai impor Tanggal baru per September 2022. Nilai-nilai ini diperkenalkan untuk mendukung fitur beberapa kunci host. Fitur ini memerlukan migrasi dari setiap kunci host tunggal yang digunakan sebelum pengenalan beberapa kunci host.
>
> Nilai tanggal yang diimpor untuk kunci host server yang dimigrasi diatur ke tanggal modifikasi terakhir untuk server. Artinya, tanggal yang Anda lihat untuk kunci host yang dimigrasi sesuai dengan tanggal terakhir Anda memodifikasi server dengan cara apa pun, sebelum migrasi kunci host server.
>
> Satu-satunya kunci yang dimigrasikan adalah kunci host server tertua atau satu-satunya. Setiap kunci tambahan memiliki tanggal aktualnya sejak Anda mengimpornya. Selain itu, kunci yang dimigrasi memiliki deskripsi yang membuatnya mudah untuk mengidentifikasinya sebagai telah dimigrasi.
>
> Migrasi terjadi antara 2 September dan 13 September. Tanggal migrasi aktual dalam rentang ini tergantung pada Wilayah server Anda.



Edit protokol transfer file

Di AWS Transfer Family konsol, Anda dapat mengedit protokol transfer file. Protokol transfer file menghubungkan klien ke titik akhir server Anda.

Untuk mengedit protokol

1. Pada halaman Detail Server, pilih Edit di samping Protokol.
2. Pada halaman Edit protokol, pilih atau kosongkan kotak centang protokol atau kotak centang untuk menambah atau menghapus protokol transfer file berikut:

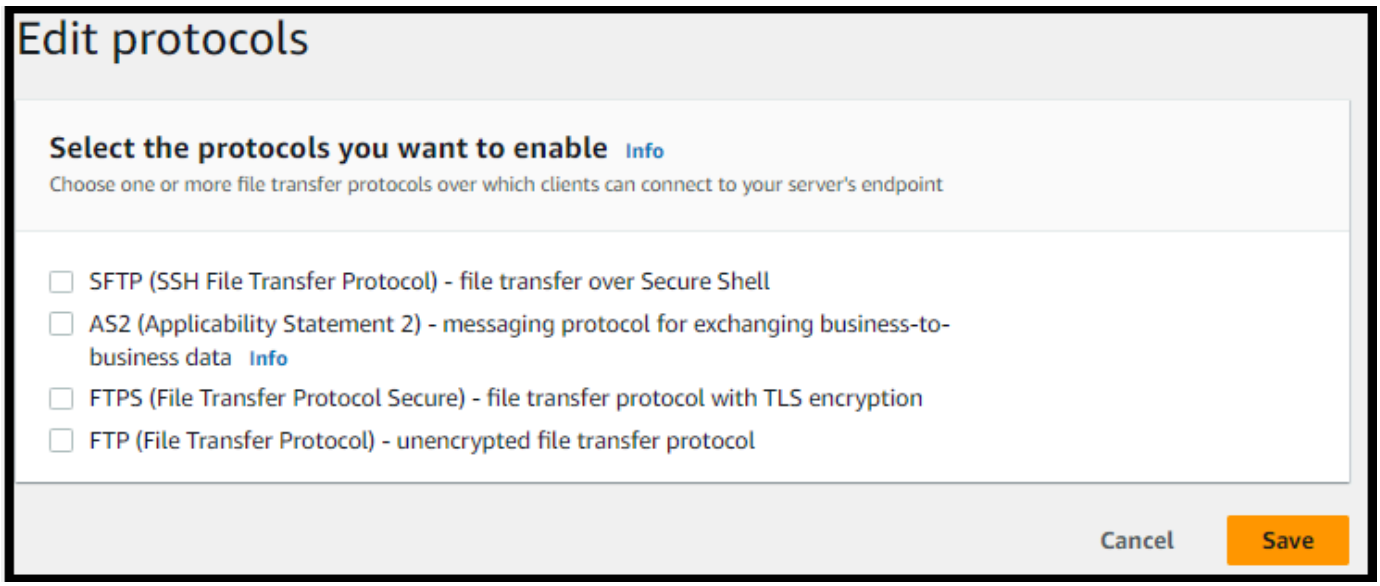
   - Secure Shell (SSH) File Transfer Protocol (SFTP) — transfer file melalui SSH

     Untuk informasi lebih lanjut tentang SFTP, lihat. Buat server berkemampuan SFTP
   - File Transfer Protocol Secure (FTPS) — transfer file dengan enkripsi TLS

     Untuk informasi lebih lanjut tentang FTP, lihatBuat server berkemampuan FTPS.
   - File Transfer Protocol (FTP) — transfer file tidak terenkripsi

     Untuk informasi lebih lanjut tentang FTPS, lihatBuat server berkemampuan FTP.

> **Note**
>
> Jika Anda memiliki server yang ada diaktifkan hanya untuk SFTP, dan Anda ingin menambahkan FTPS dan FTP, Anda harus memastikan bahwa Anda memiliki penyedia identitas yang tepat dan pengaturan tipe titik akhir yang kompatibel dengan FTPS dan FTP.



Jika Anda memilih FTPS, Anda harus memilih sertifikat yang disimpan di AWS Certificate Manager (ACM) yang akan digunakan untuk mengidentifikasi server Anda ketika klien terhubung ke sana melalui FTPS.

Untuk meminta sertifikat publik baru, lihat Meminta sertifikat publik di Panduan AWS Certificate Manager Pengguna.

Untuk mengimpor sertifikat yang ada ke ACM, lihat Mengimpor sertifikat ke ACM di Panduan Pengguna.AWS Certificate Manager

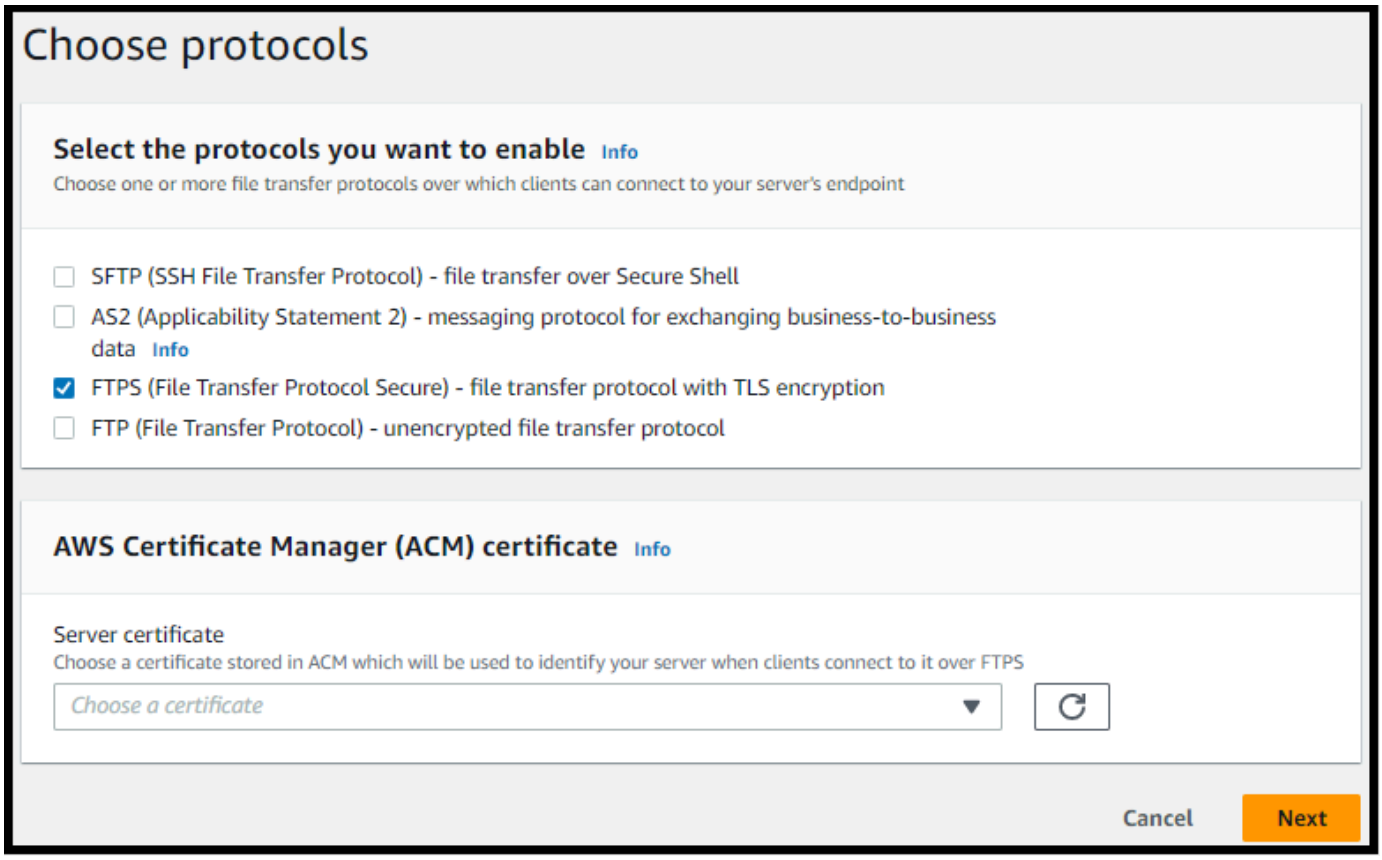
Untuk meminta sertifikat pribadi untuk menggunakan FTPS melalui alamat IP pribadi, lihat Meminta sertifikat pribadi di AWS Certificate Manager Panduan Pengguna.

Sertifikat dengan algoritme kriptografi dan ukuran kunci berikut didukung:

- 2048-bit RSA (RSA_2048)

- 4096-bit RSA (RSA_4096)
- Elliptic Prime Curve 256 bit (EC_prime256v1)
- Elliptic Prime Curve 384 bit (EC_secp384r1)
- Elliptic Prime Curve 521 bit (EC_secp521r1)

> **Note**
>
> Sertifikat harus berupa sertifikat SSL/TLS X.509 versi 3 yang valid dengan FQDN atau alamat IP yang ditentukan dan informasi tentang penerbitnya.



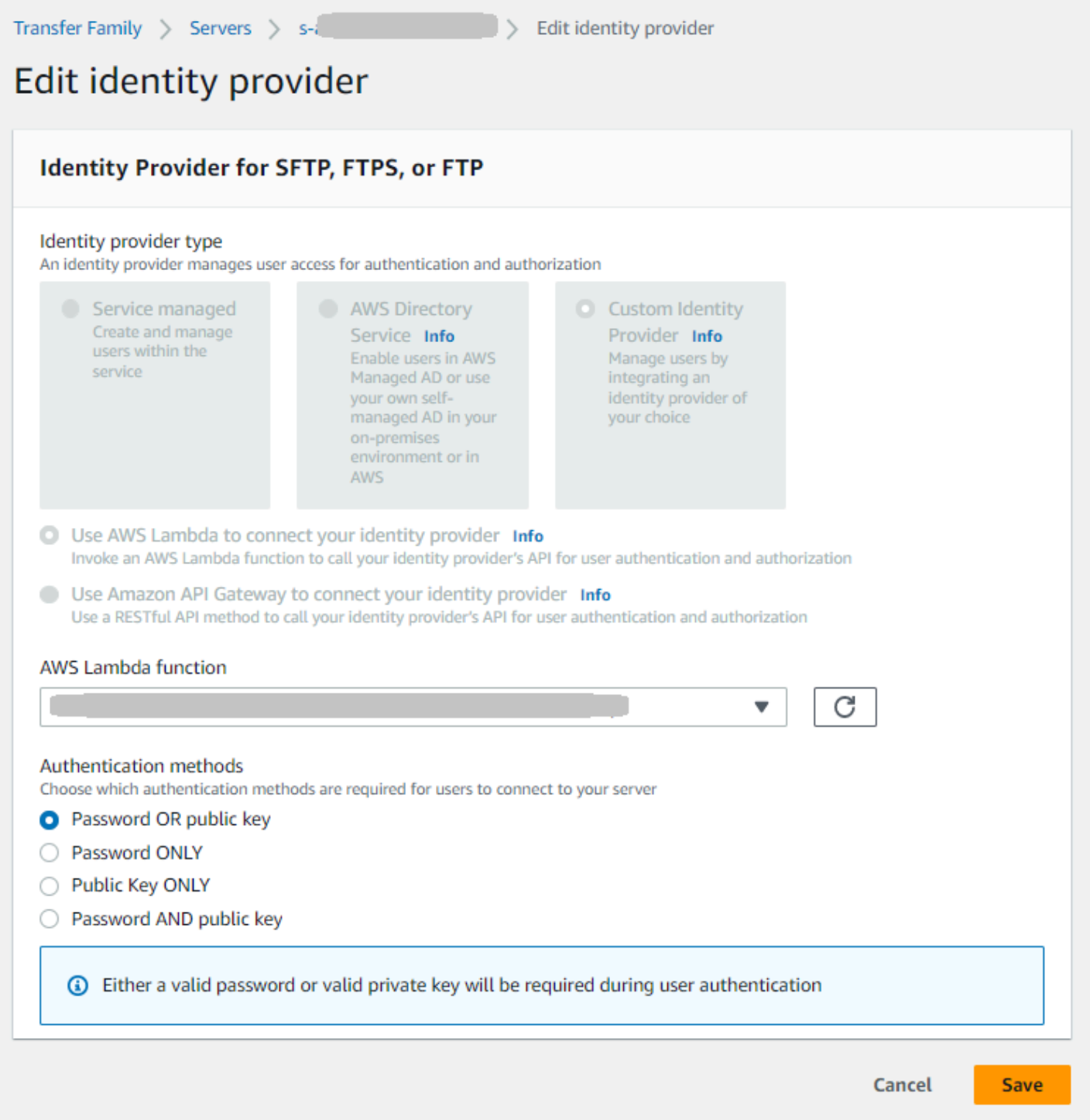
3. Pilih Simpan. Anda dikembalikan ke halaman detail Server.

Edit parameter penyedia identitas khusus

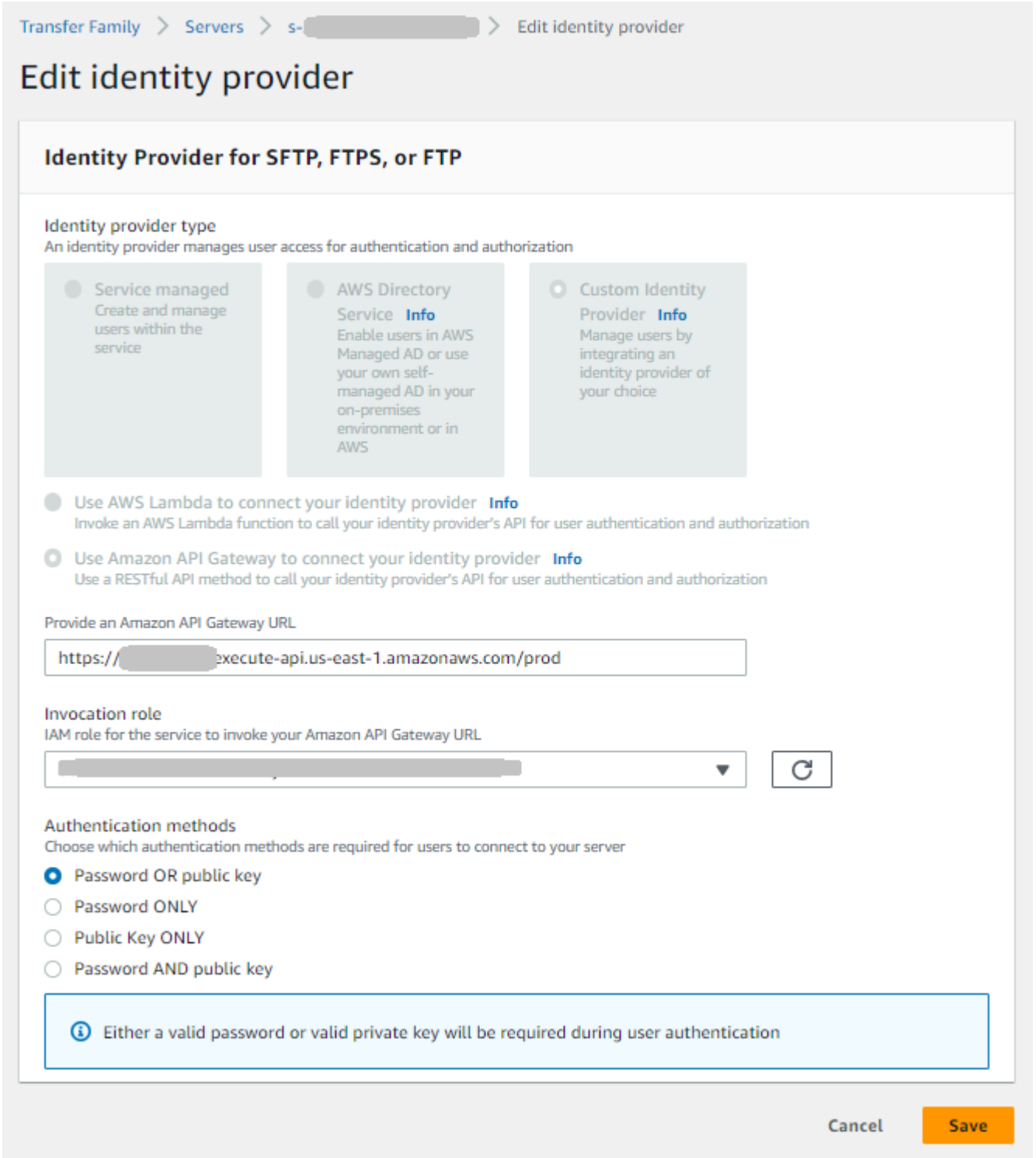
Di AWS Transfer Family konsol, untuk penyedia identitas kustom, Anda dapat mengubah beberapa pengaturan, tergantung pada apakah Anda menggunakan fungsi Lambda atau API Gateway. Dalam

kedua kasus, jika server Anda menggunakan protokol SFTP, Anda dapat mengedit metode otentikasi Anda.

- Jika Anda menggunakan Lambda sebagai penyedia identitas, Anda dapat mengubah fungsi Lambda yang mendasarinya.



- Jika Anda menggunakan API Gateway sebagai penyedia identitas, Anda dapat memperbarui URL Gateway atau peran pemanggilan, atau keduanya.



Edit titik akhir server

Di AWS Transfer Family konsol, Anda dapat memodifikasi jenis titik akhir server dan nama host khusus.

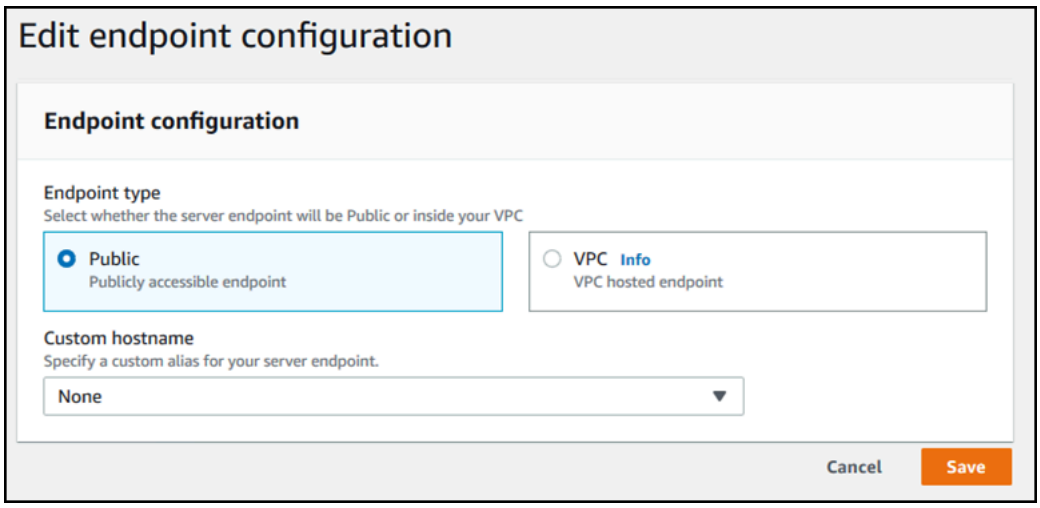
Untuk mengedit detail titik akhir server

1. Pada halaman Detail Server, pilih Edit di samping Detail titik akhir.
2. Pada halaman konfigurasi Edit titik akhir, untuk tipe Endpoint, pilih salah satu dari berikut ini:
   - Publik — Opsi ini membuat server Anda dapat diakses melalui internet.
   - VPC — Opsi ini membuat server Anda dapat diakses di cloud pribadi virtual (VPC) Anda. Untuk informasi tentang VPC, lihat. Buat server di cloud pribadi virtual
3. Untuk nama host Kustom, pilih salah satu dari berikut ini:
   - Tidak ada - Jika Anda tidak ingin menggunakan domain khusus, pilih Tidak Ada.

     Anda mendapatkan nama host server yang disediakan oleh AWS Transfer Family. Nama host server mengambil formulir`serverId`.`server.transfer.`.`regionId`.`amazonaws.com`.
   - Amazon Route 53 Alias DNS — Untuk menggunakan alias DNS yang dibuat secara otomatis untuk Anda di Route 53, pilih opsi ini.
   - DNS Lainnya — Untuk menggunakan nama host yang sudah Anda miliki di layanan DNS eksternal pilih DNS Lainnya.

   Memilih Amazon Route 53 Alias DNS atau DNS Lainnya menentukan metode resolusi nama untuk dikaitkan dengan titik akhir server Anda.

   Misalnya, domain kustom Anda mungkin`sftp.inbox.example.com`. Nama host khusus menggunakan nama DNS yang Anda berikan dan dapat diselesaikan oleh layanan DNS. Anda dapat menggunakan Route 53 sebagai penyelesai DNS Anda, atau menggunakan penyedia layanan DNS Anda sendiri. Untuk mempelajari cara AWS Transfer Family menggunakan Route 53 untuk merutekan lalu lintas dari domain kustom Anda ke titik akhir server, lihatBekerja dengan nama host khusus.



4. Pilih Simpan. Anda dikembalikan ke halaman detail Server.

Edit konfigurasi logging Anda

Di AWS Transfer Family konsol, Anda dapat mengubah konfigurasi logging Anda.

> **Note**
>
> Jika Transfer Family membuat peran IAM CloudWatch logging untuk Anda saat membuat server, peran IAM akan dipanggil. `AWSTransferLoggingAccess` Anda dapat menggunakannya untuk semua server Transfer Family Anda.

Untuk mengedit konfigurasi logging

1. Pada halaman Detail Server, pilih Edit di samping Detail tambahan.
2. Berdasarkan konfigurasi Anda, pilih antara peran logging, logging JSON terstruktur, atau keduanya. Untuk informasi selengkapnya, lihat Memperbarui logging untuk server.

Edit kebijakan keamanan

Prosedur ini menjelaskan cara mengubah kebijakan keamanan server Transfer Family dengan menggunakan AWS Transfer Family konsol atau AWS CLI.

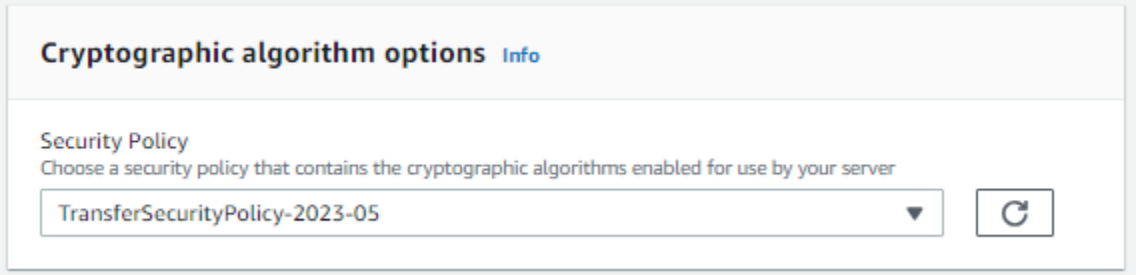
> **Note**
>
> Jika titik akhir Anda diaktifkan FIPS, Anda tidak dapat mengubah kebijakan keamanan FIPS menjadi kebijakan keamanan non-FIPS.

Console

Untuk mengedit kebijakan keamanan dengan menggunakan konsol

1. Pada halaman Detail Server, pilih Edit di samping Detail tambahan.
2. Di bagian Opsi Algoritma Kriptografi, pilih kebijakan keamanan yang berisi algoritma kriptografi yang diaktifkan untuk digunakan oleh server Anda.

   Untuk informasi selengkapnya tentang kebijakan keamanan, lihatKebijakan keamanan untuk AWS Transfer Family server.

3. Pilih Simpan.

   Anda dikembalikan ke halaman Detail Server di mana Anda dapat melihat kebijakan keamanan yang diperbarui.

AWS CLI

Untuk mengedit kebijakan keamanan dengan menggunakan CLI

1. Jalankan perintah berikut untuk melihat kebijakan keamanan saat ini yang dilampirkan ke server Anda.

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

   `describe-server`Perintah ini mengembalikan semua detail untuk server Anda, termasuk baris berikut:

```
"SecurityPolicyName": "TransferSecurityPolicy-2018-11"
```

Dalam hal ini, kebijakan keamanan untuk server adalah`TransferSecurityPolicy-2018-11`.

2. Pastikan untuk memberikan nama yang tepat dari kebijakan keamanan ke perintah. Misalnya, jalankan perintah berikut untuk memperbarui server ke`TransferSecurityPolicy-2023-05`.

```
aws transfer update-server --server-id your-server-id --security-policy-name
 "TransferSecurityPolicy-2023-05"
```

> **Note**
>
> Nama-nama kebijakan keamanan yang tersedia tercantum di[Kebijakan keamanan untuk AWS Transfer Family server](#).

Jika berhasil, perintah mengembalikan kode berikut, dan memperbarui kebijakan keamanan server Anda.

```
{
    "ServerId": "your-server-id"
}
```

Mengubah alur kerja terkelola untuk server Anda

Di AWS Transfer Family konsol, Anda dapat mengubah alur kerja terkelola yang terkait dengan server.

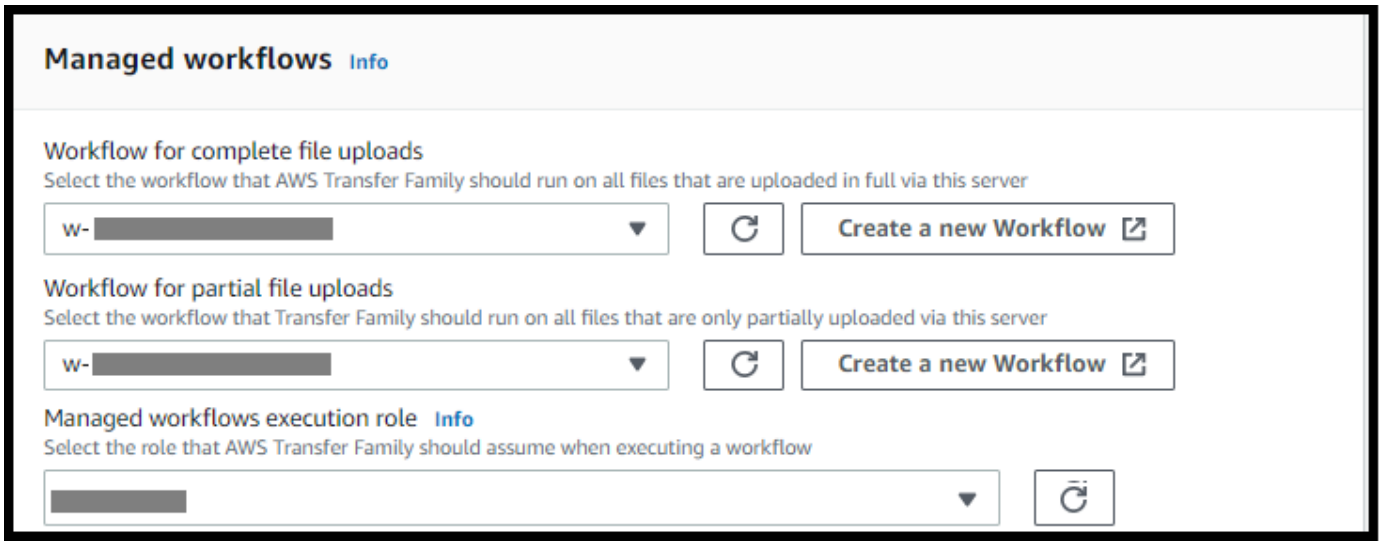
Untuk mengubah alur kerja terkelola

1. Pada halaman Detail Server, pilih Edit di samping Detail tambahan.
2. Pada halaman Edit detail tambahan, di bagian Alur kerja terkelola, pilih alur kerja yang akan dijalankan di semua unggahan.

> **Note**
>
> Jika Anda belum memiliki alur kerja, pilih Buat alur kerja baru untuk membuatnya.

a. Pilih ID alur kerja yang akan digunakan.

b. Pilih peran eksekusi. Ini adalah peran yang diasumsikan Transfer Family saat menjalankan langkah-langkah alur kerja. Untuk informasi selengkapnya, lihat Kebijakan IAM untuk alur kerja. Pilih Simpan.



3. Pilih Simpan. Anda dikembalikan ke halaman detail Server.

Ubah spanduk tampilan untuk server Anda

Di AWS Transfer Family konsol, Anda dapat mengubah spanduk tampilan yang terkait dengan server.

Untuk mengubah spanduk tampilan

1. Pada halaman Detail Server, pilih Edit di samping Detail tambahan.
2. Pada halaman Edit detail tambahan, di bagian Tampilan spanduk, masukkan teks untuk spanduk tampilan yang tersedia.
3. Pilih Simpan. Anda dikembalikan ke halaman detail Server.

Menempatkan server Anda secara online atau offline

Di AWS Transfer Family konsol, Anda dapat membawa server Anda online atau membawanya offline.

Untuk membawa server Anda secara online

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi, pilih Server.
3. Pilih kotak centang server yang sedang offline.
4. Untuk Tindakan, pilih Mulai.

Diperlukan beberapa menit bagi server untuk beralih dari offline ke online.

> **Note**
>
> Ketika Anda menghentikan server untuk membuatnya offline, saat ini Anda masih dikenakan biaya layanan untuk server itu. Untuk menghilangkan biaya berbasis server tambahan, hapus server itu.

Untuk membuat server Anda offline

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi, pilih Server.
3. Pilih kotak centang server yang sedang online.
4. Untuk Tindakan, pilih Berhenti.

Saat server memulai atau mematikan, server tidak tersedia untuk operasi file. Konsol tidak menampilkan status mulai dan berhenti.

Jika Anda menemukan kondisi kesalahan `START_FAILED` atau`STOP_FAILED`, hubungi AWS Support untuk membantu menyelesaikan masalah Anda.

Kelola kunci host untuk server berkemampuan SFTP

Important

Jika Anda tidak berencana untuk memigrasikan pengguna yang ada dari server berkemampuan SFTP yang sudah ada ke server baru yang mendukung SFTP, abaikan bagian ini.
Mengubah kunci host server secara tidak sengaja dapat mengganggu. Bergantung pada bagaimana klien SFTP Anda dikonfigurasi, klien SFTP dapat segera gagal, dengan pesan bahwa tidak ada kunci host tepercaya, atau menyajikan petunjuk yang mengancam. Jika ada skrip untuk mengotomatisasi koneksi, kemungkinan besar akan gagal juga.

Secara default, AWS Transfer Family berikan kunci host untuk server berkemampuan SFTP Anda. Anda dapat mengganti kunci host default dengan kunci host dari server lain. Lakukan hanya jika Anda berencana untuk memindahkan pengguna yang ada dari server berkemampuan SFTP yang ada ke server berkemampuan SFTP baru Anda.

Untuk mencegah pengguna diminta memverifikasi keaslian server berkemampuan SFTP lagi, impor kunci host untuk server lokal Anda ke server berkemampuan SFTP. Melakukan hal ini juga mencegah pengguna Anda mendapatkan peringatan tentang potensi man-in-the-middle serangan.

Anda juga dapat memutar kunci host secara berkala, sebagai langkah keamanan tambahan.

Note

Meskipun konsol Transfer Family memungkinkan Anda untuk menentukan dan menambahkan kunci host server untuk semua server, kunci ini hanya berguna untuk server yang menggunakan protokol SFTP.

Topik

- Tambahkan kunci host server tambahan
- Hapus kunci host server
- Putar kunci host server
- Informasi kunci host server tambahan

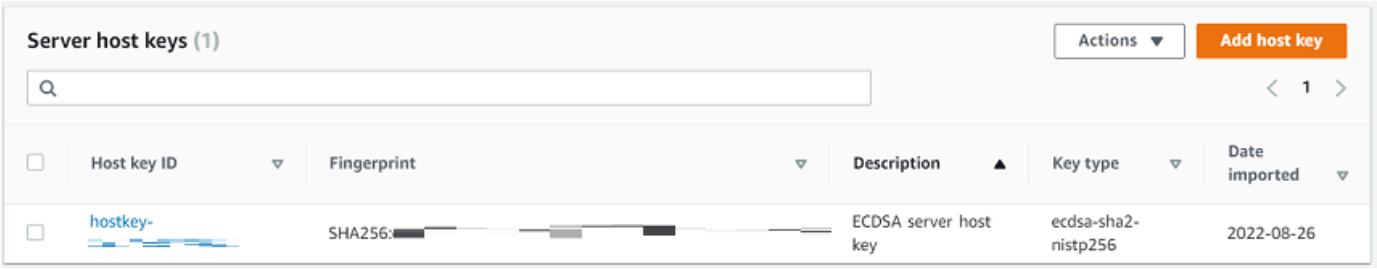
Tambahkan kunci host server tambahan

Di AWS Transfer Family konsol, Anda dapat menambahkan kunci host server tambahan. Menambahkan kunci host tambahan dari format yang berbeda dapat berguna untuk mengidentifikasi server ketika klien terhubung dengannya, serta meningkatkan profil keamanan Anda. Misalnya, jika kunci asli Anda adalah kunci RSA, Anda dapat menambahkan kunci ECDSA tambahan.

> **Note**
>
> Klien SFTP terhubung menggunakan kunci publik pertama yang dimilikinya yang dapat mencocokkan salah satu kunci server aktif.

Untuk menambahkan kunci host server tambahan

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi kiri, pilih Server, lalu pilih server yang menggunakan protokol SFTP.
3. Pada halaman detail server, gulir ke bawah ke bagian Kunci host Server.

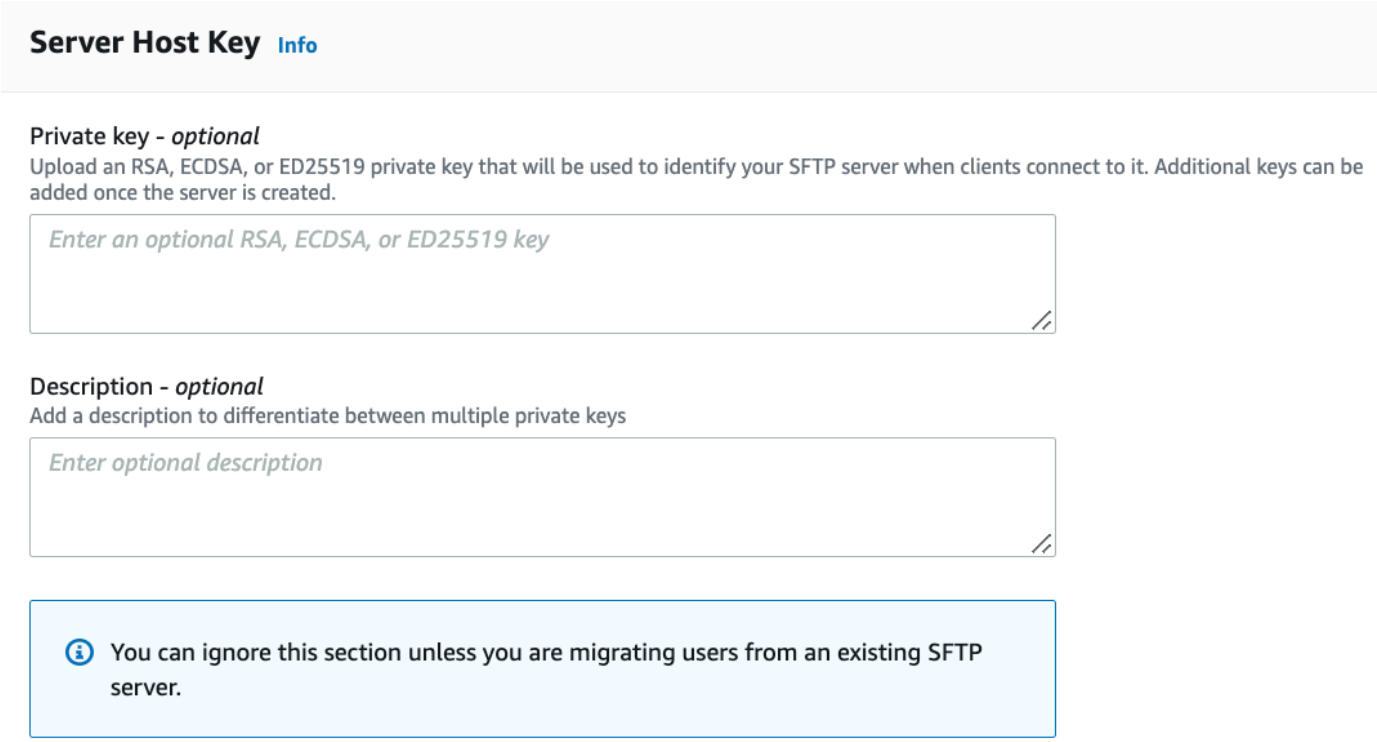


4. Pilih Tambahkan kunci host.

   Halaman kunci Add server host ditampilkan.

5. Di bagian Server Host Key, masukkan kunci pribadi RSA, ECDSA, atau ED25519 yang digunakan untuk mengidentifikasi server Anda ketika klien terhubung dengannya melalui server yang mendukung SFTP.

   > **Note**
   >
   > Saat Anda membuat kunci host server, pastikan untuk menentukan `-N ""` (tidak ada frasa sandi). Lihat Membuat kunci SSH di macOS, Linux, atau Unix untuk detail tentang cara menghasilkan pasangan kunci.



6. (Opsional) Tambahkan deskripsi untuk membedakan antara beberapa kunci host server. Anda juga dapat menambahkan tag untuk kunci Anda.
7. Pilih Tambah kunci. Anda dikembalikan ke halaman detail Server.

Untuk menambahkan kunci host dengan menggunakan AWS Command Line Interface (AWS CLI), gunakan operasi the section called “ImportHostKey” API dan berikan kunci host baru. Jika Anda membuat server baru yang mendukung SFTP, Anda memberikan kunci host sebagai parameter dalam operasi API. the section called “CreateServer” Anda juga dapat menggunakan AWS CLI untuk memperbarui deskripsi untuk kunci host yang ada.

Contoh `import-host-key` AWS CLI perintah berikut mengimpor kunci host untuk server berkemampuan SFTP tertentu.

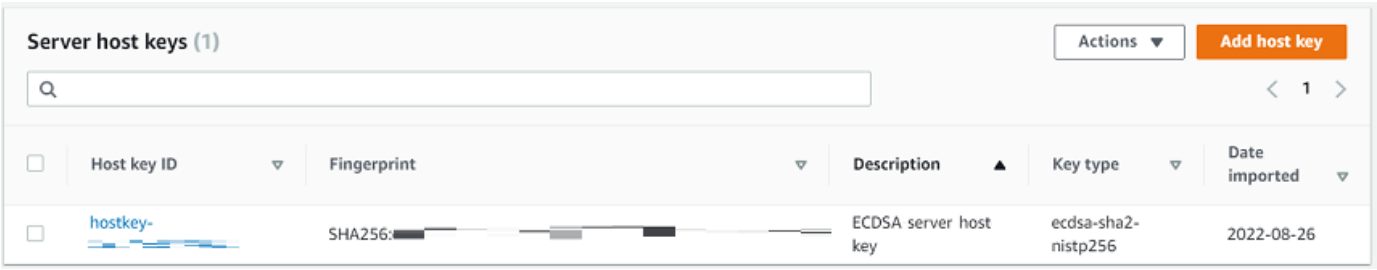
```
aws transfer import-host-key --description key-description --server-id your-server-id
 --host-key-body file://my-host-key
```

Hapus kunci host server

Di AWS Transfer Family konsol, Anda dapat menghapus kunci host server.

Untuk menghapus kunci host server

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Di panel navigasi kiri, pilih Server, lalu pilih server yang menggunakan protokol SFTP.
3. Pada halaman detail server, gulir ke bawah ke bagian Kunci host Server.



4. Di bagian Server Host Keys, pilih kunci, dan kemudian di bawah Tindakan, pilih Hapus.
5. Di kotak dialog konfirmasi yang muncul, masukkan kata**`delete`**, lalu pilih Hapus untuk mengonfirmasi bahwa Anda ingin menghapus kunci host.

Kunci host dihapus dari halaman Server.

Untuk menghapus kunci host dengan menggunakan AWS CLI, gunakan operasi the section called "DeleteHostKey" API dan berikan ID server dan ID kunci host.

Contoh `delete-host-key` AWS CLI perintah berikut menghapus kunci host untuk server berkemampuan SFTP tertentu.

```
aws transfer delete-host-key --server-id your-server-id --host-key-id your-host-key-id
```

Putar kunci host server

Secara berkala, Anda dapat memutar kunci host server Anda.

Bagaimana klien memilih kunci host server

Cara Transfer Family memilih kunci server mana yang akan diterapkan tergantung pada kondisi untuk klien SFTP, seperti yang dijelaskan di sini. Asumsinya adalah bahwa ada satu kunci yang lebih tua dan satu kunci yang lebih baru.

- Klien SFTP tidak memiliki kunci host publik sebelumnya untuk server. Pertama kali klien terhubung ke server, salah satu dari berikut ini terjadi:
  - Klien gagal koneksi, jika dikonfigurasi untuk melakukannya.

  - Atau, klien memilih kunci pertama yang cocok dengan algoritme yang mungkin tersedia dan bertanya kepada pengguna apakah kunci itu dapat dipercaya. Jika demikian, klien memperbarui file secara otomatis (atau `known_hosts` file konfigurasi lokal atau sumber daya apa pun yang digunakan klien untuk merekam keputusan kepercayaan) dan memasukkan kunci itu.
- Klien SFTP memiliki kunci yang lebih lama dalam file-nya`known_hosts`. Klien lebih suka menggunakan kunci ini, bahkan jika ada kunci yang lebih baru, baik untuk algoritma kunci ini atau algoritma lain. Ini karena klien memiliki tingkat kepercayaan yang lebih tinggi untuk kunci yang ada di `known_hosts` file-nya.
- Klien SFTP memiliki kunci baru (dalam salah satu algoritma yang tersedia) dalam file kuncinya`known_hosts`. Klien mengabaikan kunci lama karena tidak dipercaya dan menggunakan kunci baru.
- Klien SFTP memiliki kedua kunci dalam file-nya`known_hosts`. Klien memilih kunci pertama berdasarkan indeks yang cocok dengan daftar kunci yang tersedia yang ditawarkan oleh server.

Transfer Family lebih suka klien SFTP memiliki semua kunci dalam `known_hosts` file-nya, karena ini memungkinkan fleksibilitas paling besar saat menghubungkan ke server Transfer Family. Rotasi kunci didasarkan pada fakta bahwa beberapa entri dapat ada dalam `known_hosts` file untuk server Transfer Family yang sama.

Putar prosedur kunci host server

Sebagai contoh, asumsikan bahwa Anda telah menambahkan set kunci host server berikut ke server Transfer Family Anda.

Kunci host server

Jenis kunci host	Tanggal ditambahkan ke server
RSA	1 April 2020
ECDSA	Februari 1, 2020
ED25519	1 Desember, 2019
RSA	1 Oktober 2019
ECDSA	1 Juni 2019

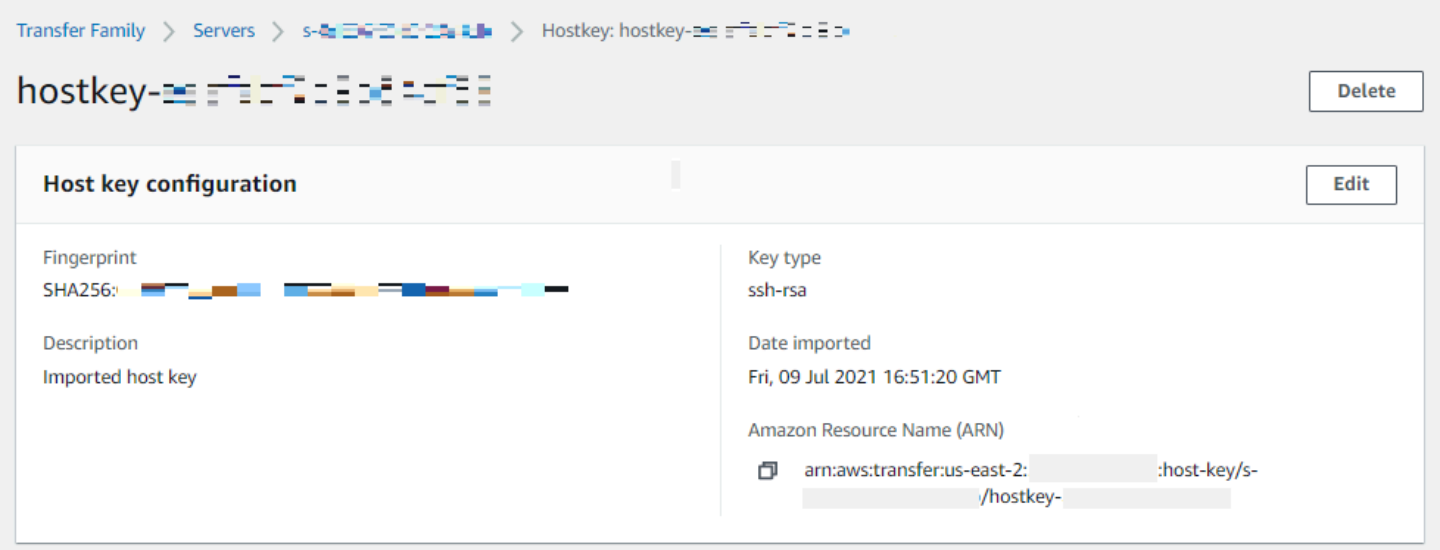
Jenis kunci host	Tanggal ditambahkan ke server
ED25519	Maret 1, 2019

Untuk memutar kunci host server

1. Tambahkan kunci host server baru. Prosedur ini dijelaskan dalamTambahkan kunci host server tambahan.
2. Hapus satu atau beberapa kunci host dari jenis yang sama yang telah Anda tambahkan sebelumnya. Prosedur ini dijelaskan dalamHapus kunci host server.
3. Semua kunci terlihat, dan dapat aktif, tunduk pada perilaku yang dijelaskan sebelumnya diBagaimana klien memilih kunci host server.

Informasi kunci host server tambahan

Anda dapat memilih kunci host untuk menampilkan detail kunci tersebut.



Anda dapat menghapus kunci host, atau mengedit deskripsinya dari menu Tindakan di layar Detail Server. Pilih tombol host, lalu pilih tindakan yang sesuai dari menu.

Memantau penggunaan di konsol

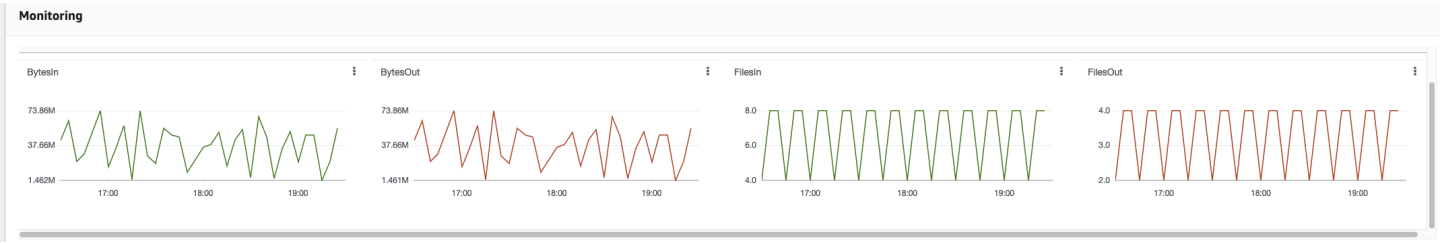
Anda bisa mendapatkan informasi tentang metrik server Anda di halaman detail Servernya. Ini memberi Anda satu tempat untuk memantau beban kerja transfer file Anda. Anda dapat melacak berapa banyak file yang telah Anda tukarkan dengan mitra Anda dan melacak penggunaannya dengan cermat menggunakan dasbor terpusat. Untuk detailnya, lihat Lihat detail server SFTP, FTPS, dan FTP. Tabel berikut menjelaskan metrik yang tersedia untuk Transfer Family.

Namespace	Metrik	Deskripsi
`AWS/Transfer`	`BytesIn`	Jumlah total byte yang ditransfer ke server. Unit: Hitungan Periode: 5 menit
	`BytesOut`	Jumlah total byte yang ditransfer keluar dari server. Unit: Jumlah Periode: 5 menit
	`FilesIn`	Jumlah total file yang ditransfer ke server. Untuk server yang menggunakan protokol AS2, metrik ini mewakili jumlah pesan yang diterima. Unit: Hitungan Periode: 5 menit
	`FilesOut`	Jumlah total file yang ditransfer keluar dari server.

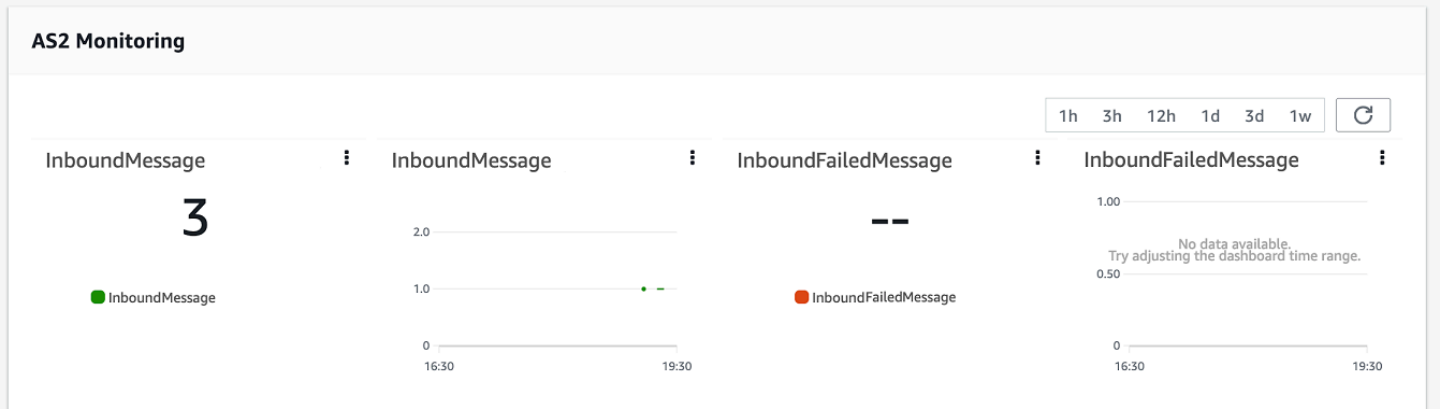
Namespace	Metrik	Deskripsi
		Unit: Hitungan Periode: 5 menit
	`InboundMessage`	Jumlah total pesan AS2 yang berhasil diterima dari mitra dagang. Unit: Hitungan Periode: 5 menit
	`InboundFailedMessage`	Jumlah total pesan AS2 yang tidak berhasil diterima dari mitra dagang. Artinya, mitra dagang mengirim pesan, tetapi server Transfer Family tidak berhasil memproses nya. Unit: Hitungan Periode: 5 menit
	`OnPartialUploadExecutionsStarted`	Jumlah total eksekusi on-partial-upload alur kerja dimulai di server. Unit: Hitungan Periode: 1 menit
	`OnPartialUploadExecutionsSuccess`	Jumlah total eksekusi on-partial-upload alur kerja yang berhasil di server. Unit: Hitungan Periode: 1 menit
	`OnPartialUploadExecutionsFailed`	Jumlah total eksekusi on-partial-upload alur kerja yang gagal di server. Unit: Hitungan Periode: 1 menit

Namespace	Metrik	Deskripsi
	`OnUploadExecutionsStarted`	Jumlah total eksekusi alur kerja dimulai di server. Unit: Hitungan Periode: 1 menit
	`OnUploadExecutionsSuccess`	Jumlah total eksekusi alur kerja yang berhasil di server. Unit: Hitungan Periode: 1 menit
	`OnUploadExecutionsFailed`	Jumlah total eksekusi alur kerja yang gagal di server. Unit: Hitungan Periode: 1 menit

Bagian Monitoring berisi empat grafik individual. Grafik ini menunjukkan byte masuk, byte keluar, file masuk, dan file keluar.



Untuk server yang memiliki protokol AS2 diaktifkan, ada bagian Pemantauan AS2 di bawah informasi Pemantauan. Bagian ini berisi detail untuk jumlah pesan masuk, baik yang berhasil maupun yang gagal.



Untuk membuka grafik yang dipilih di jendelanya sendiri, pilih ikon perluas (). Anda juga dapat mengklik ikon elipsis vertikal grafik () untuk membuka menu tarik-turun dengan item berikut:

- Memperbesar — Membuka grafik yang dipilih di jendelanya sendiri.
- Refresh — Muat ulang grafik dengan data terbaru.
- Lihat dalam metrik — Membuka detail metrik yang sesuai di Amazon. CloudWatch
- Lihat log - Membuka grup log yang sesuai di CloudWatch.

Mengelola kontrol akses

Anda dapat mengontrol akses pengguna ke AWS Transfer Family sumber daya dengan menggunakan kebijakan AWS Identity and Access Management (IAM). Kebijakan IAM adalah pernyataan, biasanya dalam format JSON, yang memungkinkan tingkat akses tertentu ke sumber daya. Anda menggunakan kebijakan IAM untuk menentukan operasi file apa yang Anda ingin mengizinkan pengguna Anda untuk melakukan dan tidak melakukan. Anda juga dapat menggunakan kebijakan IAM untuk menentukan bucket atau bucket Amazon S3 yang ingin Anda akses kepada pengguna. Untuk menentukan kebijakan ini bagi pengguna, Anda membuat peran IAM AWS Transfer Family yang memiliki kebijakan IAM dan hubungan kepercayaan yang terkait dengannya.

Setiap pengguna diberi peran IAM. Jenis peran IAM yang AWS Transfer Family digunakan disebut peran layanan. Saat pengguna masuk ke server Anda, AWS Transfer Family asumsikan peran IAM yang dipetakan ke pengguna. Untuk mempelajari cara membuat peran IAM yang menyediakan akses pengguna ke bucket Amazon S3, lihat Membuat peran untuk mendelegasikan izin ke AWS layanan di Panduan Pengguna IAM.

Anda dapat memberikan akses hanya tulis ke objek Amazon S3 dengan menggunakan izin tertentu dalam kebijakan IAM. Untuk detailnya, lihat Memberikan kemampuan untuk hanya menulis dan daftar file.

Blog AWS Penyimpanan berisi posting yang merinci cara mengatur akses hak istimewa paling sedikit. Untuk detailnya, lihat Menerapkan akses hak istimewa terkecil dalam AWS Transfer Family alur kerja.

> **Note**
>
> Jika bucket Amazon S3 dienkripsi menggunakan AWS Key Management Service (AWS KMS), Anda harus menentukan izin tambahan dalam kebijakan Anda. Untuk detailnya, lihat Enkripsi data di Amazon S3. Selain itu, Anda dapat melihat informasi selengkapnya tentang kebijakan sesi di Panduan Pengguna IAM.

Topik

- Mengizinkan akses baca dan tulis ke bucket Amazon S3
- Membuat kebijakan sesi untuk bucket Amazon S3

- Mencegah pengguna berjalan mkdir di bucket S3

Mengizinkan akses baca dan tulis ke bucket Amazon S3

Bagian ini menjelaskan cara membuat kebijakan IAM yang memungkinkan akses baca dan tulis ke bucket Amazon S3 tertentu. Menetapkan peran IAM yang memiliki kebijakan IAM ini kepada pengguna Anda memberi pengguna akses baca/tulis ke bucket Amazon S3 yang ditentukan.

Kebijakan berikut menyediakan akses baca, tulis, dan penandaan terprogram ke bucket Amazon S3. `PutObjectACL`Pernyataan `GetObjectACL` dan pernyataan hanya diperlukan jika Anda perlu mengaktifkan Akses Lintas Akun. Artinya, server Transfer Family Anda perlu mengakses bucket di akun yang berbeda.

```
{
  "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`Tindakan tersebut membutuhkan izin untuk ember itu sendiri. `DELETE`Tindakan`PUT`,`GET`, dan memerlukan izin objek. Karena ini adalah sumber daya yang berbeda, mereka ditentukan menggunakan Nama Sumber Daya Amazon (ARN) yang berbeda.

Untuk lebih membatasi akses pengguna hanya ke `home` awalan bucket Amazon S3 yang ditentukan, lihat. Membuat kebijakan sesi untuk bucket Amazon S3

Membuat kebijakan sesi untuk bucket Amazon S3

Kebijakan sesi adalah kebijakan AWS Identity and Access Management (IAM) yang membatasi pengguna pada bagian tertentu dari bucket Amazon S3. Ia melakukannya dengan mengevaluasi akses secara real time.

> **Note**
>
> Kebijakan sesi hanya digunakan dengan Amazon S3. Untuk Amazon EFS, Anda menggunakan izin file POSIX untuk membatasi akses.

Anda dapat menggunakan kebijakan sesi saat Anda perlu memberikan akses yang sama ke sekelompok pengguna ke bagian tertentu dari bucket Amazon S3 Anda. Misalnya, sekelompok pengguna mungkin hanya memerlukan akses ke `home` direktori. Kelompok pengguna itu berbagi peran IAM yang sama.

> **Note**
>
> Panjang maksimum kebijakan sesi adalah 2048 karakter. Untuk detail selengkapnya, lihat parameter Permintaan kebijakan untuk `CreateUser` tindakan dalam referensi API.

Untuk membuat kebijakan sesi, gunakan variabel kebijakan berikut dalam kebijakan IAM Anda:

- `${transfer:HomeBucket}`
- `${transfer:HomeDirectory}`
- `${transfer:HomeFolder}`
- `${transfer:UserName}`

Important

Anda tidak dapat menggunakan variabel sebelumnya dalam Kebijakan Terkelola. Anda juga tidak dapat menggunakannya sebagai variabel kebijakan dalam definisi peran IAM. Anda membuat variabel-variabel ini dalam kebijakan IAM dan menyediakannya secara langsung saat menyiapkan pengguna Anda. Selain itu, Anda tidak dapat menggunakan `${aws:Username}` variabel dalam kebijakan sesi ini. Variabel ini mengacu pada nama pengguna IAM dan bukan nama pengguna yang diperlukan oleh AWS Transfer Family.

Kode berikut menunjukkan contoh kebijakan sesi.

```
{
  "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**
>
> Contoh kebijakan sebelumnya mengasumsikan bahwa pengguna memiliki direktori home mereka diatur untuk menyertakan garis miring, untuk menandakan bahwa itu adalah direktori. Jika, di sisi lain, Anda menetapkan pengguna `HomeDirectory` tanpa garis miring, maka Anda harus memasukkannya sebagai bagian dari kebijakan Anda.

Dalam contoh kebijakan sebelumnya, perhatikan penggunaan parameter`transfer:HomeFolder`,`transfer:HomeBucket`, dan `transfer:HomeDirectory` kebijakan. Parameter ini diatur untuk `HomeDirectory` yang dikonfigurasi untuk pengguna, seperti yang dijelaskan dalam HomeDirectorydanMenerapkan metode API Gateway. Parameter ini memiliki definisi berikut:

- `transfer:HomeBucket`Parameter diganti dengan komponen pertama dari`HomeDirectory`.
- `transfer:HomeFolder`Parameter diganti dengan bagian `HomeDirectory` parameter yang tersisa.
- `transfer:HomeDirectory`Parameter memiliki garis miring depan (`/`) yang dihapus sehingga dapat digunakan sebagai bagian dari Nama Sumber Daya Amazon S3 (ARN) dalam sebuah pernyataan. `Resource`

> **Note**
>
> Jika Anda menggunakan direktori logik—yaitu, pengguna adalah `LOGICAL` —parameter kebijakan ini (`HomeBucket`,`HomeDirectory`, dan`HomeFolder`) tidak didukung. `homeDirectoryType`

Misalnya, asumsikan bahwa `HomeDirectory` parameter yang dikonfigurasi untuk pengguna Transfer Family adalah`/home/bob/amazon/stuff/`.

- `transfer:HomeBucket`diatur ke`/home`.
- `transfer:HomeFolder`diatur ke`/bob/amazon/stuff/`.
- `transfer:HomeDirectory`menjadi`home/bob/amazon/stuff/`.

Yang pertama `"Sid"` memungkinkan pengguna untuk membuat daftar semua direktori mulai dari`/home/bob/amazon/stuff/`.

Yang kedua `"Sid"` membatasi pengguna `put` dan `get` akses ke jalur yang sama,`/home/bob/amazon/stuff/`.

Dengan kebijakan sebelumnya, saat pengguna masuk, mereka hanya dapat mengakses objek di direktori home mereka. Pada waktu koneksi, AWS Transfer Family ganti variabel-variabel ini dengan nilai yang sesuai untuk pengguna. Melakukan hal ini memudahkan penerapan dokumen kebijakan yang sama ke beberapa pengguna. Pendekatan ini mengurangi overhead peran IAM dan manajemen kebijakan untuk mengelola akses pengguna ke bucket Amazon S3 Anda.

Anda juga dapat menggunakan kebijakan sesi untuk menyesuaikan akses untuk setiap pengguna berdasarkan kebutuhan bisnis Anda. Untuk informasi selengkapnya, lihat Izin untuk AssumeRole, AssumeRoleWith SALL, dan AssumeRoleWithWebIdentity di Panduan Pengguna IAM.

> **Note**
>
> AWS Transfer Family menyimpan kebijakan JSON, bukan Nama Sumber Daya Amazon (ARN) kebijakan. Jadi, ketika Anda mengubah kebijakan di konsol IAM, Anda harus kembali ke AWS Transfer Family konsol dan memperbarui pengguna Anda dengan konten kebijakan terbaru. Anda dapat memperbarui pengguna pada tab Info Kebijakan di bagian Konfigurasi pengguna.
> Jika Anda menggunakan AWS CLI, Anda dapat menggunakan perintah berikut untuk memperbarui kebijakan.
>
> ```
> aws transfer update-user --server-id server --user-name user --policy \
>    "$(aws iam get-policy-version --policy-arn policy --version-id version --
> output json)"
> ```

Mencegah pengguna berjalan **`mkdir`** di bucket S3

Anda dapat membatasi kemampuan pengguna untuk membuat direktori di bucket Amazon S3. Untuk melakukannya, Anda membuat kebijakan IAM yang memungkinkan `s3:PutObject` tindakan tetapi juga menyangkalnya ketika kunci diakhiri dengan “/” (garis miring ke depan). Kebijakan contoh berikut memungkinkan pengguna untuk mengunggah file ke bucket Amazon S3 tetapi menolak mkdir perintah di bucket Amazon S3.

```
{
   "Sid":"DenyMkdir",
   "Action":[
      "s3:PutObject"
   ],
   "Effect":"Deny",
   "Resource":[
      "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*/",
      "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*/*"
    ]
}
```

> **Note**
>
> Baris sumber daya kedua membuat tidak mungkin bagi pengguna untuk membuat sub-folder dengan menjalankan perintah sepertiput `my-file` *DOC-EXAMPLE-BUCKET*`/new-folder/my-file`.

Logging untuk AWS Transfer Family

AWS Transfer Familyterintegrasi dengan keduanya AWS CloudTrail dan Amazon CloudWatch. CloudTrail dan CloudWatch melayani tujuan yang berbeda tetapi saling melengkapi:

- CloudTrail adalah AWS layanan yang membuat catatan tindakan yang diambil dalam diri AndaAkun AWS. Ini terus memantau dan merekam panggilan API untuk aktivitas seperti login konsol, AWS Command Line Interface perintah, dan panggilan SDK/API. Ini memungkinkan Anda untuk menyimpan log siapa yang mengambil tindakan apa, kapan, dan dari mana. CloudTrail membantu audit, manajemen akses, dan kepatuhan terhadap peraturan dengan memberikan riwayat semua aktivitas di AWS lingkungan Anda. Untuk detailnya, lihat Panduan AWS CloudTrail Pengguna.
- CloudWatch adalah layanan pemantauan untuk AWS sumber daya dan aplikasi. Ini mengumpulkan metrik dan log untuk memberikan visibilitas ke pemanfaatan sumber daya, kinerja aplikasi, dan kesehatan sistem secara keseluruhan. CloudWatch membantu tugas operasional seperti pemecahan masalah, pengaturan alarm, dan penskalaan otomatis. Untuk detailnya, lihat Panduan CloudWatch Pengguna Amazon.

Topik

- AWS CloudTrailpenebangan untuk AWS Transfer Family
- CloudWatch Pencatatan Amazon untuk AWS Transfer Family

AWS CloudTrailpenebangan untuk AWS Transfer Family

AWS Transfer Familyterintegrasi denganAWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau AWS layanan diAWS Transfer Family. CloudTrail menangkap semua panggilan API untuk AWS Transfer Family sebagai peristiwa. Panggilan yang direkam mencakup panggilan dari AWS Transfer Family konsol dan panggilan kode ke operasi API AWS Transfer Family ini.

Trail adalah konfigurasi yang memungkinkan pengiriman peristiwa sebagai file log ke bucket Amazon S3 yang Anda tentukan. CloudTrail file log berisi satu atau lebih entri log. Peristiwa mewakili permintaan tunggal dari sumber manapun dan mencakup informasi tentang tindakan yang diminta, tanggal dan waktu tindakan, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah

jejak tumpukan yang diurutkan dari panggilan API publik, jadi file tersebut tidak muncul dalam urutan tertentu.

Untuk catatan berkelanjutan tentang peristiwa di akun AWS Anda, termasuk peristiwa untuk AWS Transfer Family, buat jejak. Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Secara default, saat Anda membuat jejak di konsol tersebut, jejak diterapkan ke semua Wilayah AWS. Jejak mencatat peristiwa dari semua Wilayah di partisi AWS dan mengirimkan file log ke bucket Amazon S3 yang Anda tentukan. Selain itu, Anda dapat mengonfigurasi AWS layanan lain untuk menganalisis lebih lanjut dan menindaklanjuti data peristiwa yang dikumpulkan dalam CloudTrail log. Untuk informasi selengkapnya, lihat berikut:

- Gambaran umum untuk membuat jejak
- CloudTrail layanan dan integrasi yang didukung
- Mengonfigurasi notifikasi Amazon SNS untuk CloudTrail
- Menerima file CloudTrail log dari beberapa wilayah dan Menerima file CloudTrail log dari beberapa akun

Semua AWS Transfer Family tindakan dicatat oleh CloudTrail dan didokumentasikan dalam ActionsAPI reference. Misalnya, panggilan ke`CreateServer`, `ListUsers` dan `StopServer` tindakan menghasilkan entri dalam file CloudTrail log.

Setiap peristiwa atau entri log berisi informasi tentang siapa yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan hal berikut:

- Baik permintaan tersebut dibuat dengan kredensial pengguna root atau AWS Identity and Access Management.
- Baik permintaan tersebut dibuat dengan kredensial keamanan sementara untuk peran atau pengguna gabungan.
- Apakah permintaan dibuat oleh layanan AWS lain.

Untuk informasi selengkapnya, lihat Elemen userIdentity CloudTrail .

Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara berkelanjutan ke bucket Amazon S3, termasuk acara untuk. AWS Transfer Family Jika Anda tidak mengonfigurasi jejak, Anda masih dapat melihat peristiwa terbaru di CloudTrail konsol dalam Riwayat acara.

Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuatAWS Transfer Family, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk mempelajari selengkapnya CloudTrail, lihat Panduan AWS CloudTrail Pengguna.

Topik

- Aktifkan AWS CloudTrail pencatatan
- Contoh entri log untuk membuat server

Aktifkan AWS CloudTrail pencatatan

Anda dapat memantau panggilan AWS Transfer Family API menggunakanAWS CloudTrail. Dengan memantau panggilan API, Anda bisa mendapatkan informasi keamanan dan operasional yang berguna. Jika Anda mengaktifkan pencatatan level objek Amazon S3, `RoleSessionName` terdapat dalam bidang Peminta sebagai. `[AWS:Role Unique Identifier]/username.sessionid@server-id` Untuk informasi selengkapnya tentang pengidentifikasi unik peran AWS Identity and Access Management (IAM), lihat Pengidentifikasi unik di Panduan Pengguna. AWS Identity and Access Management

> **Important**
>
> Panjang maksimum `RoleSessionName` adalah 64 karakter. Jika `RoleSessionName` lebih panjang, `server-id` akan terpotong.

Contoh entri log untuk membuat server

Contoh berikut menunjukkan entri CloudTrail log (dalam format JSON) yang menunjukkan tindakan. `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"
}
```

CloudWatch Pencatatan Amazon untuk AWS Transfer Family

Amazon CloudWatch memantau AWS Transfer Family sumber daya Anda dan aplikasi yang Anda jalankan AWS secara real time. Anda dapat menggunakan CloudWatch untuk mengumpulkan dan melacak metrik, yang merupakan variabel yang dapat Anda ukur untuk sumber daya dan aplikasi Anda.

CloudWatch Halaman beranda secara otomatis menampilkan metrik tentang Transfer Family dan setiap AWS layanan lain yang Anda gunakan. Anda dapat membuat dasbor kustom tambahan untuk menampilkan metrik tentang aplikasi khusus Anda, dan menampilkan koleksi metrik-metrik kustom yang Anda pilih.

Anda dapat membuat alarm yang mengawasi metrik dan mengirimkan notifikasi atau secara otomatis melakukan perubahan pada sumber daya yang sedang dipantau ketika ada pelanggaran ambang batas. Misalnya, Anda dapat memantau file yang ditransfer ke server Transfer Family dan menggunakan data tersebut untuk menentukan apakah Anda perlu menggunakan server tambahan untuk menangani peningkatan beban. Anda juga dapat menggunakan data ini untuk menghentikan atau menghapus instance yang kurang digunakan untuk menghemat uang.

Topik

- Membuat, memperbarui, dan melihat logging untuk server
- Mengelola logging untuk alur kerja
- Konfigurasikan peran CloudWatch logging
- Melihat aliran log Transfer Family
- Membuat CloudWatch alarm Amazon
- Mencatat panggilan API Amazon S3 ke log akses S3
- Contoh untuk membatasi masalah wakil yang membingungkan

- CloudWatch struktur log untuk Transfer Family
- Contoh entri CloudWatch log
- Menggunakan CloudWatch metrik untuk Transfer Family
- Menggunakan Notifikasi Pengguna AWS dengan AWS Transfer Family

Membuat, memperbarui, dan melihat logging untuk server

Untuk semua AWS Transfer Family server, Anda dapat memilih di antara dua opsi untuk logging: `LoggingRole` (digunakan untuk mencatat alur kerja yang dilampirkan ke server) atau`StructuredLogDestinations`. Manfaat menggunakan `StructuredLogDestinations` meliputi:

- Menerima log dalam format JSON terstruktur.
- Kueri log Anda dengan Amazon CloudWatch Logs Insights, yang secara otomatis menemukan bidang berformat JSON.
- Bagikan grup log di seluruh AWS Transfer Family sumber daya memungkinkan Anda menggabungkan aliran log dari beberapa server ke dalam satu grup log, sehingga memudahkan pengelolaan konfigurasi pemantauan dan pengaturan penyimpanan log.
- Buat metrik dan visualisasi agregat yang dapat ditambahkan ke dasbor. CloudWatch
- Lacak data penggunaan dan kinerja dengan menggunakan grup log untuk membuat metrik log, visualisasi, dan dasbor terkonsolidasi.

Opsi untuk `LoggingRole` atau `StructuredLogDestinations` dikonfigurasi dan dikontrol secara terpisah. Untuk setiap server, Anda dapat mengatur satu atau kedua metode logging, atau mengonfigurasi server Anda agar tidak memiliki logging apa pun (meskipun ini tidak disarankan).

Jika Anda membuat server baru menggunakan konsol Transfer Family, logging diaktifkan secara default. Setelah membuat server, Anda dapat menggunakan panggilan `UpdateServer` API untuk mengubah konfigurasi logging Anda. Untuk detailnya, lihat StructuredLogDestinations.

Saat ini, untuk alur kerja, jika ingin logging diaktifkan, Anda harus menentukan peran logging:

- Jika Anda mengaitkan alur kerja dengan server, menggunakan panggilan `CreateServer` atau `UpdateServer` API, sistem tidak secara otomatis membuat peran logging. Jika Anda ingin mencatat peristiwa alur kerja Anda, Anda perlu melampirkan peran logging secara eksplisit ke server.

- Jika Anda membuat server menggunakan konsol Transfer Family dan melampirkan alur kerja, log akan dikirim ke grup log yang berisi ID server dalam nama. Formatnya`/aws/transfer/`*`server-id`*, misalnya,`/aws/transfer/s-1111aaaa2222bbbb3`. Log server dapat dikirim ke grup log yang sama atau yang lain.

Pertimbangan log untuk membuat dan mengedit server di konsol

- Server baru yang dibuat melalui konsol hanya mendukung logging JSON terstruktur, kecuali alur kerja dilampirkan ke server.
- Tidak ada logging bukanlah opsi untuk server baru yang Anda buat di konsol.
- Server yang ada dapat mengaktifkan logging JSON terstruktur melalui konsol kapan saja.
- Mengaktifkan logging JSON terstruktur melalui konsol menonaktifkan metode logging yang ada, agar tidak menagih pelanggan dua kali lipat. Pengecualiannya adalah jika alur kerja dilampirkan ke server.
- Jika Anda mengaktifkan logging JSON terstruktur, Anda tidak dapat menonaktifkannya nanti melalui konsol.
- Jika Anda mengaktifkan logging JSON terstruktur, Anda dapat mengubah tujuan grup log melalui konsol kapan saja.
- Jika Anda mengaktifkan logging JSON terstruktur, Anda tidak dapat mengedit peran logging melalui konsol jika Anda telah mengaktifkan kedua jenis logging melalui API. Pengecualiannya adalah jika server Anda memiliki alur kerja yang terpasang. Namun, peran logging terus muncul di Detail tambahan.

Pertimbangan logging untuk membuat dan mengedit server menggunakan API atau SDK

- Jika Anda membuat server baru melalui API, Anda dapat mengonfigurasi salah satu atau kedua jenis logging, atau memilih tidak ada pencatatan.
- Untuk server yang ada, aktifkan dan nonaktifkan logging JSON terstruktur kapan saja.
- Anda dapat mengubah grup log melalui API kapan saja.
- Anda dapat mengubah peran logging melalui API kapan saja.

Untuk mengaktifkan logging terstruktur, Anda harus masuk ke akun dengan izin berikut

- `logs:CreateLogDelivery`

- `logs:DeleteLogDelivery`
- `logs:DescribeLogGroups`
- `logs:DescribeResourcePolicies`
- `logs:GetLogDelivery`
- `logs:ListLogDeliveries`
- `logs:PutResourcePolicy`
- `logs:UpdateLogDelivery`

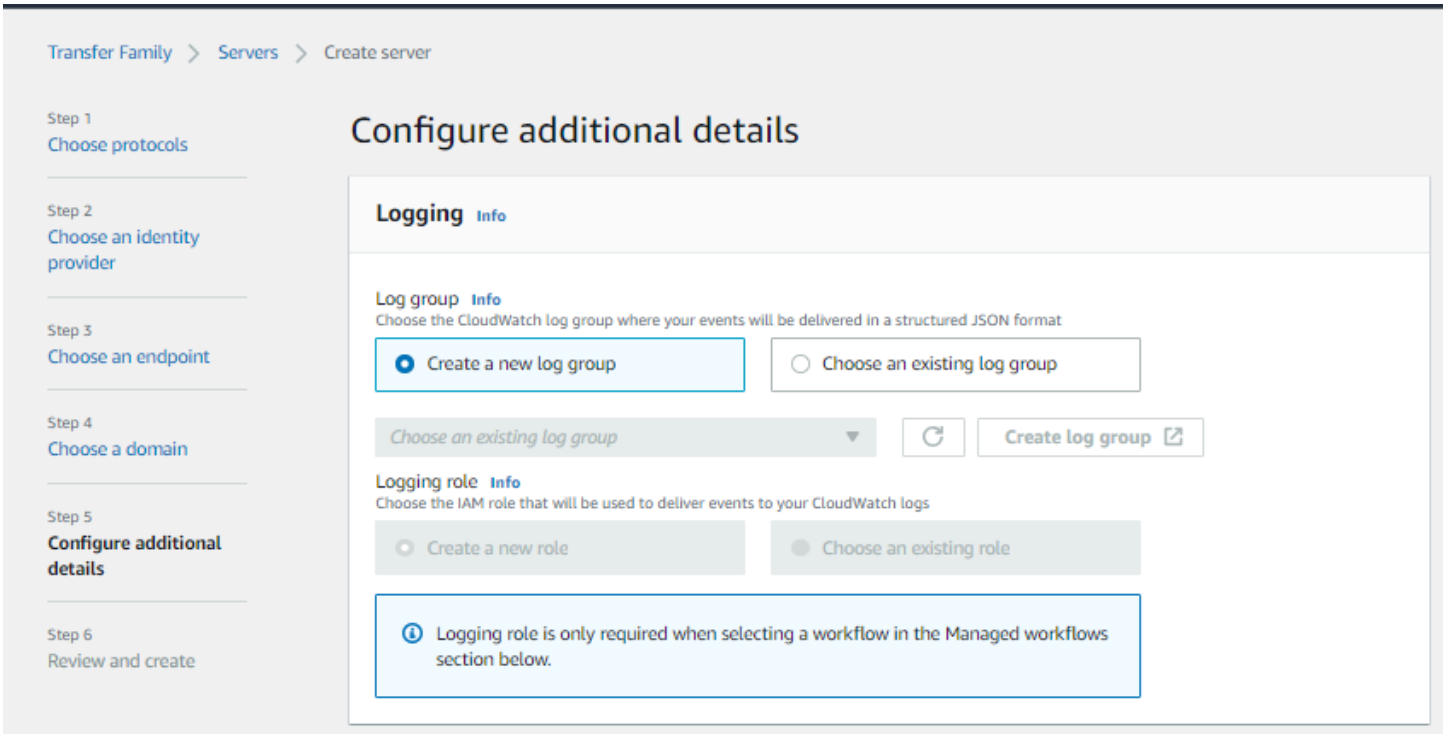
Contoh kebijakan tersedia di bagian iniKonfigurasikan peran CloudWatch logging.

Topik

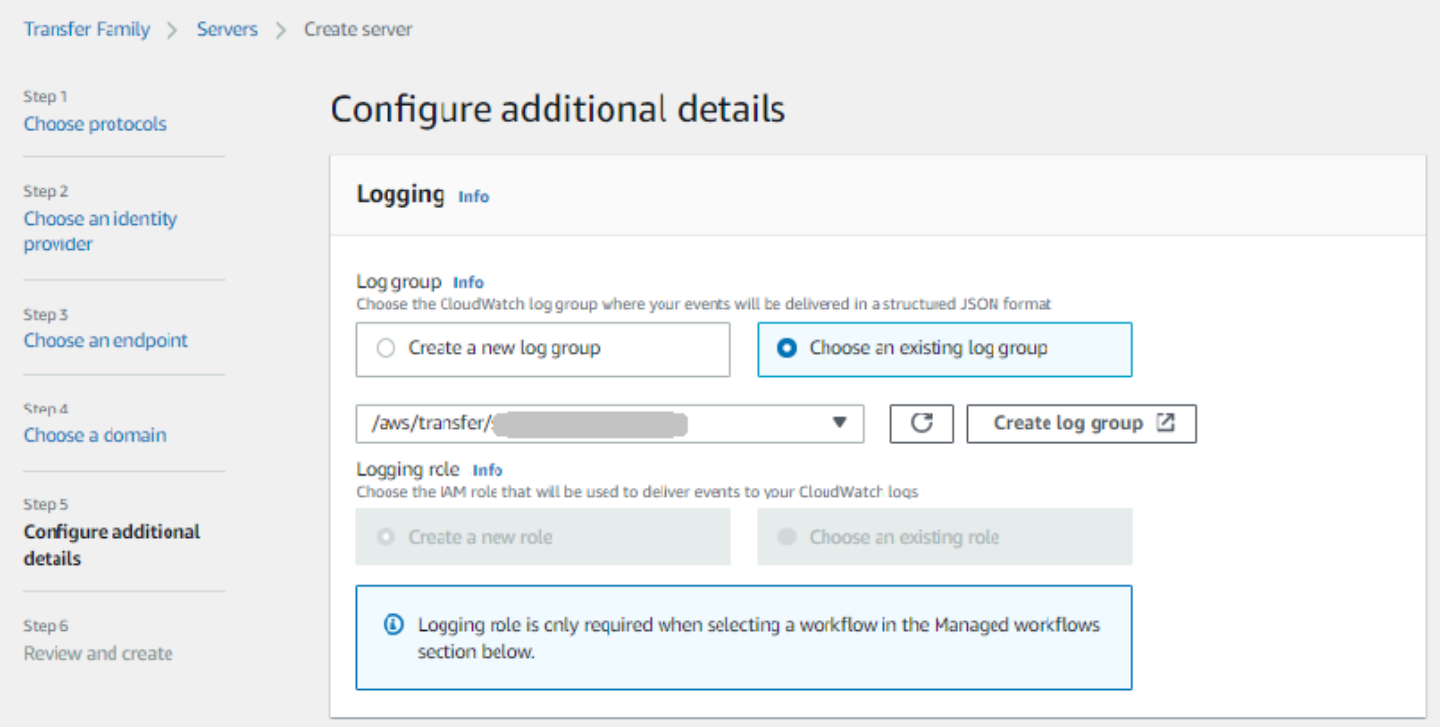
- Membuat logging untuk server
- Memperbarui logging untuk server
- Melihat konfigurasi server

Membuat logging untuk server

Saat Anda membuat server baru, pada halaman Konfigurasi detail tambahan, Anda dapat menentukan grup log yang ada, atau membuat yang baru.



Jika Anda memilih grup log yang ada, Anda harus memilih salah satu yang terkait dengan Anda Akun AWS.



Jika Anda memilih Buat grup log, CloudWatch konsol (https://console.aws.amazon.com/cloudwatch/) terbuka ke halaman Buat grup log. Untuk detailnya, lihat Membuat grup log di CloudWatch Log.

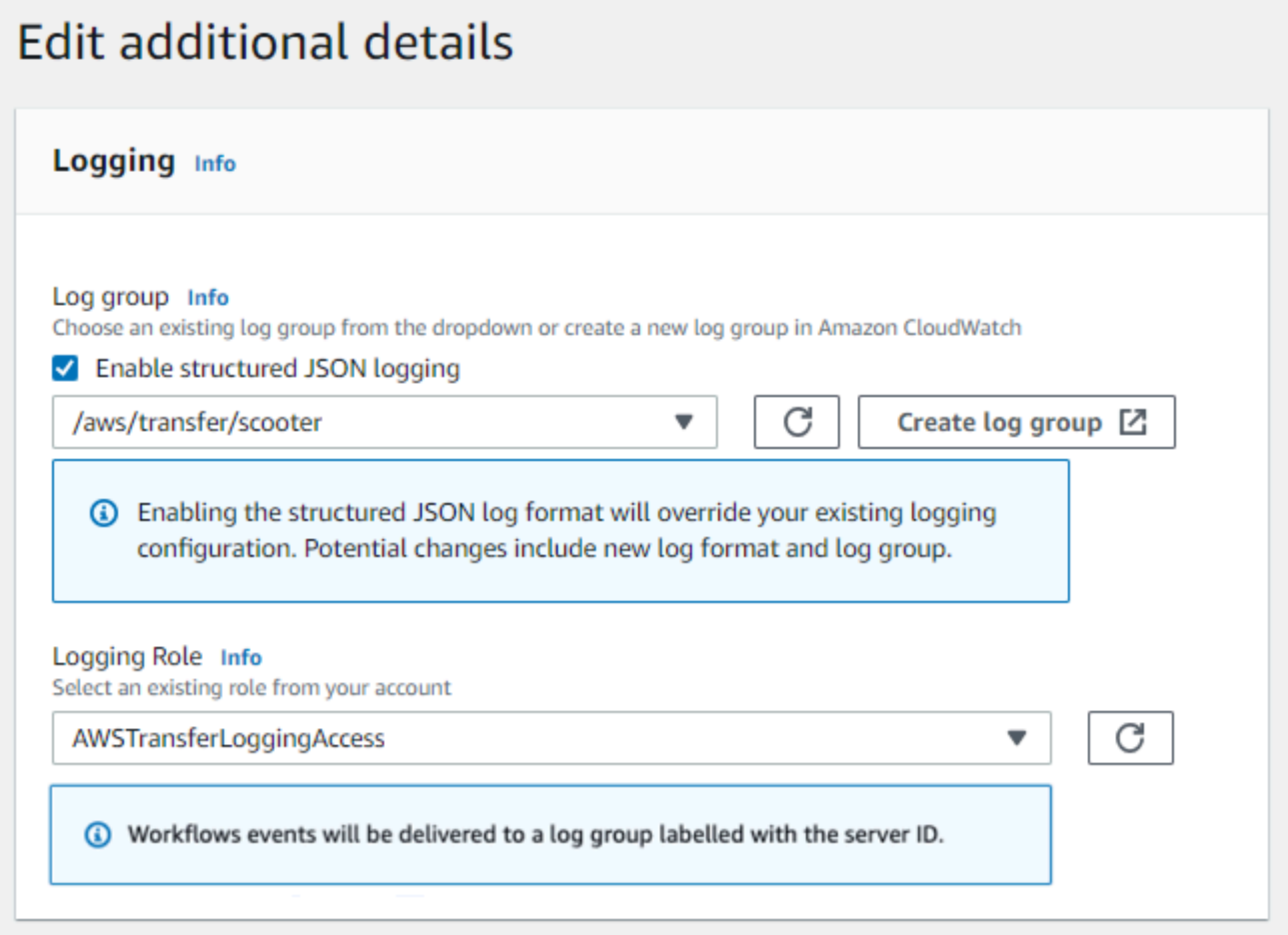
Memperbarui logging untuk server

Detail untuk pencatatan bergantung pada skenario pembaruan Anda.

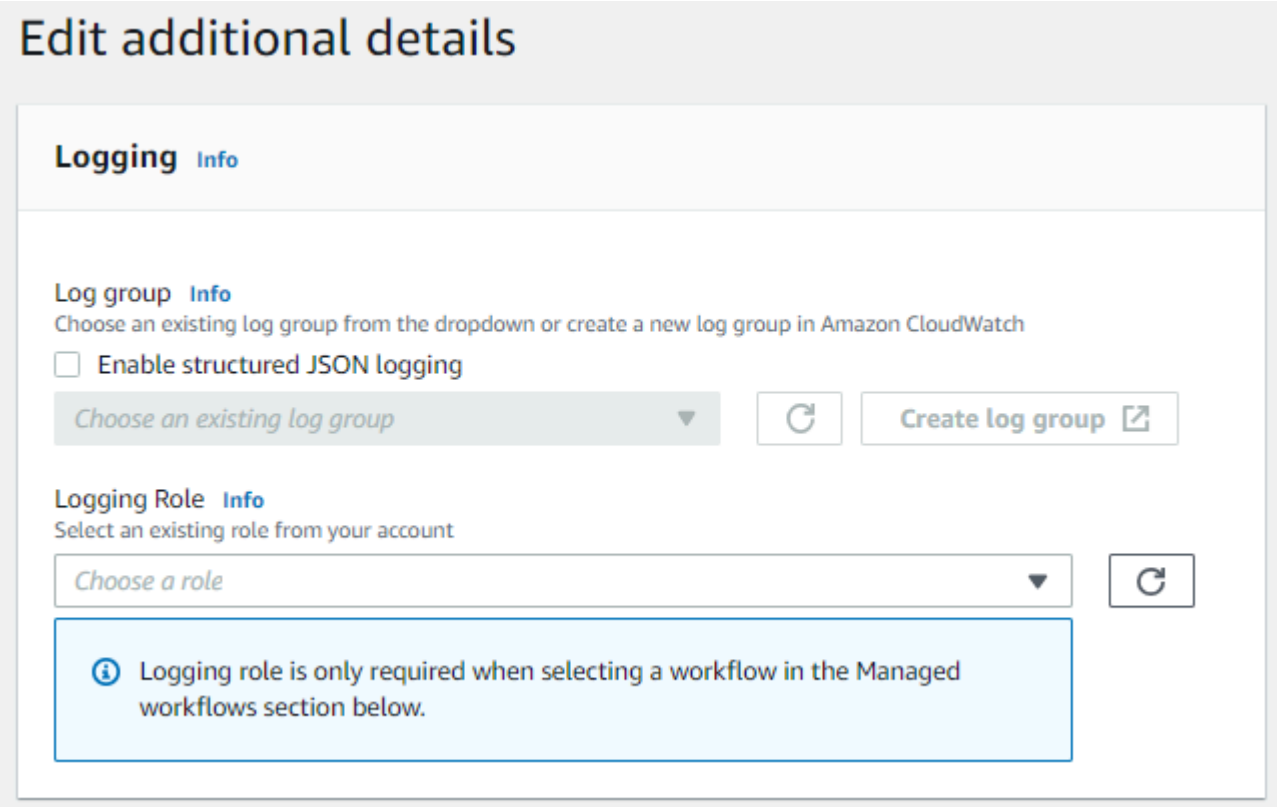
> **Note**
>
> Saat Anda memilih logging JSON terstruktur, mungkin ada penundaan, dalam kasus yang jarang terjadi, di mana Transfer Family berhenti masuk dalam format lama, tetapi membutuhkan waktu untuk mulai masuk dalam format JSON yang baru. Hal ini dapat mengakibatkan peristiwa yang tidak masuk log. Tidak akan ada gangguan layanan, tetapi Anda harus berhati-hati mentransfer file selama satu jam pertama setelah mengubah metode logging Anda, karena log dapat dihapus.

Jika Anda mengedit server yang ada, opsi Anda bergantung pada status server.

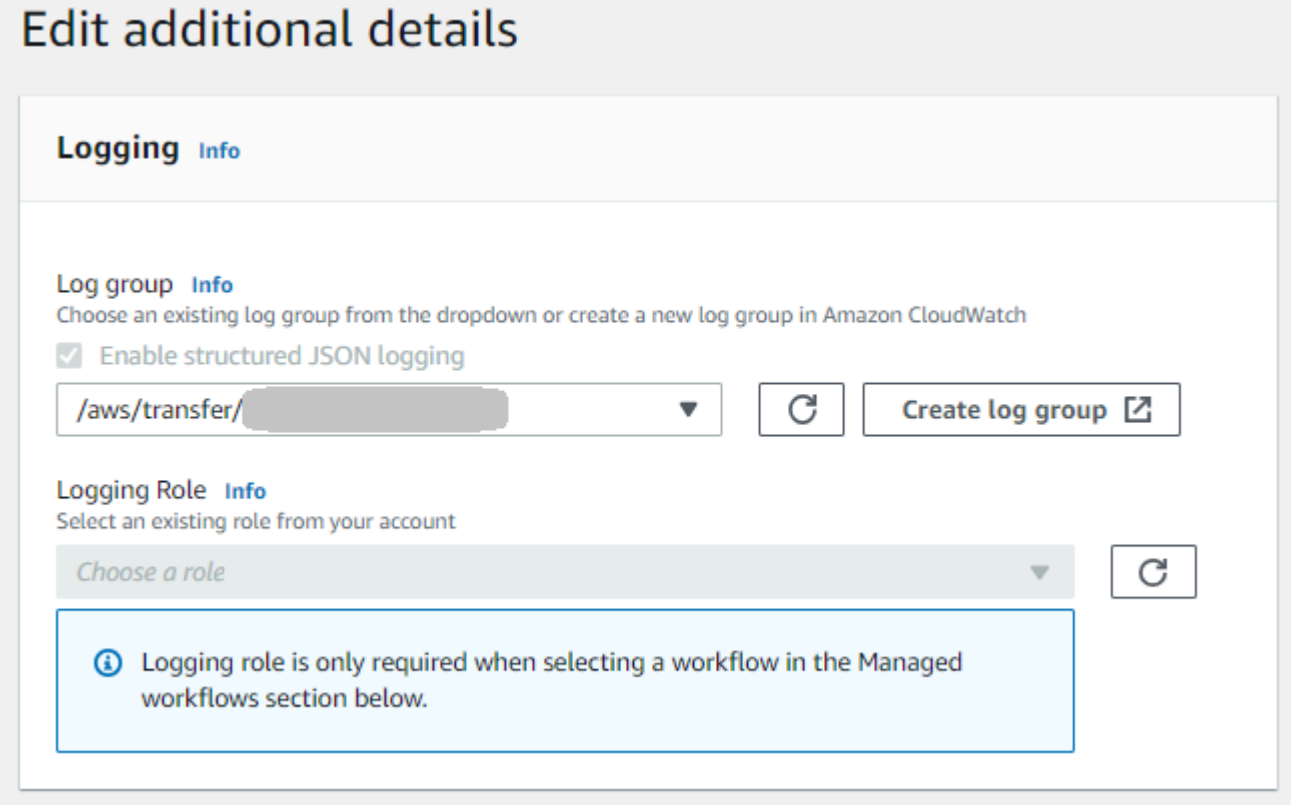
- Server sudah mengaktifkan peran logging, tetapi tidak mengaktifkan logging JSON Terstruktur.



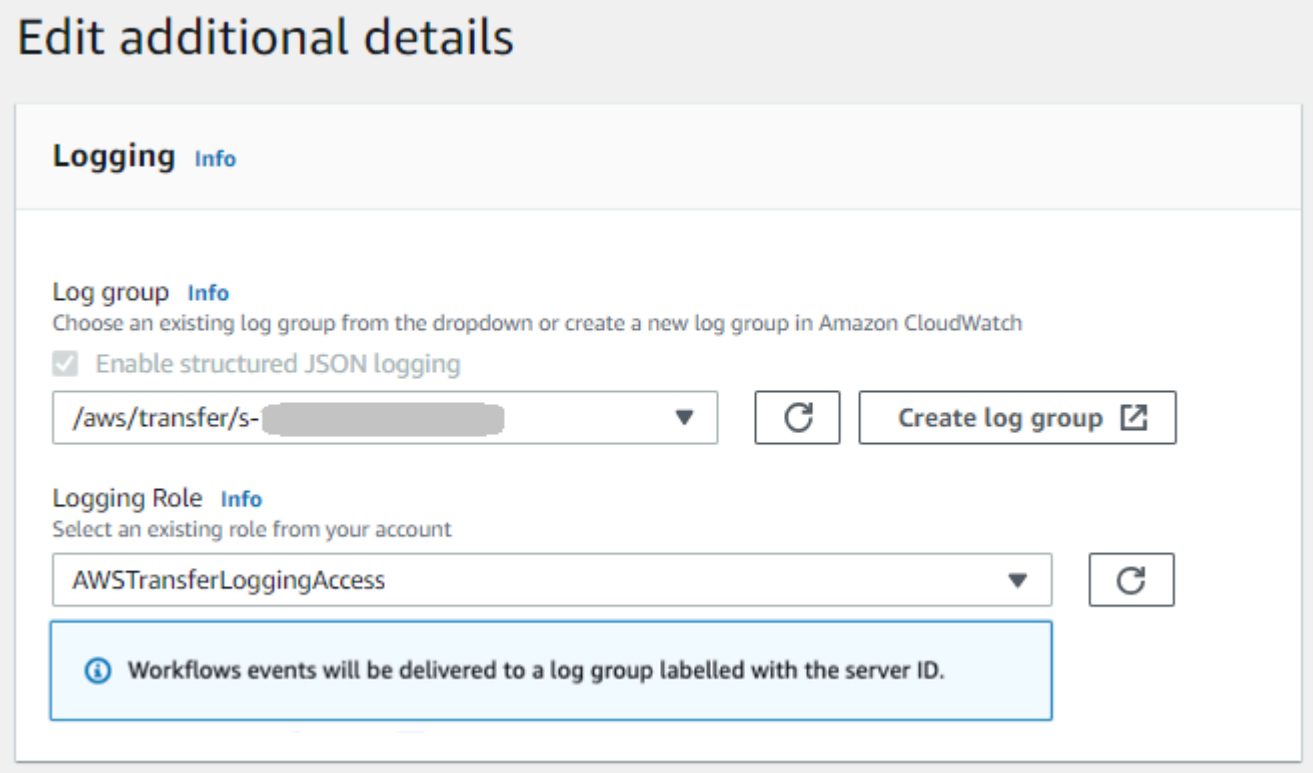
- Server tidak memiliki logging yang diaktifkan.



- Server sudah mengaktifkan logging JSON Terstruktur, tetapi tidak memiliki peran logging yang ditentukan.



- Server sudah mengaktifkan logging JSON Terstruktur, dan juga memiliki peran logging yang ditentukan.

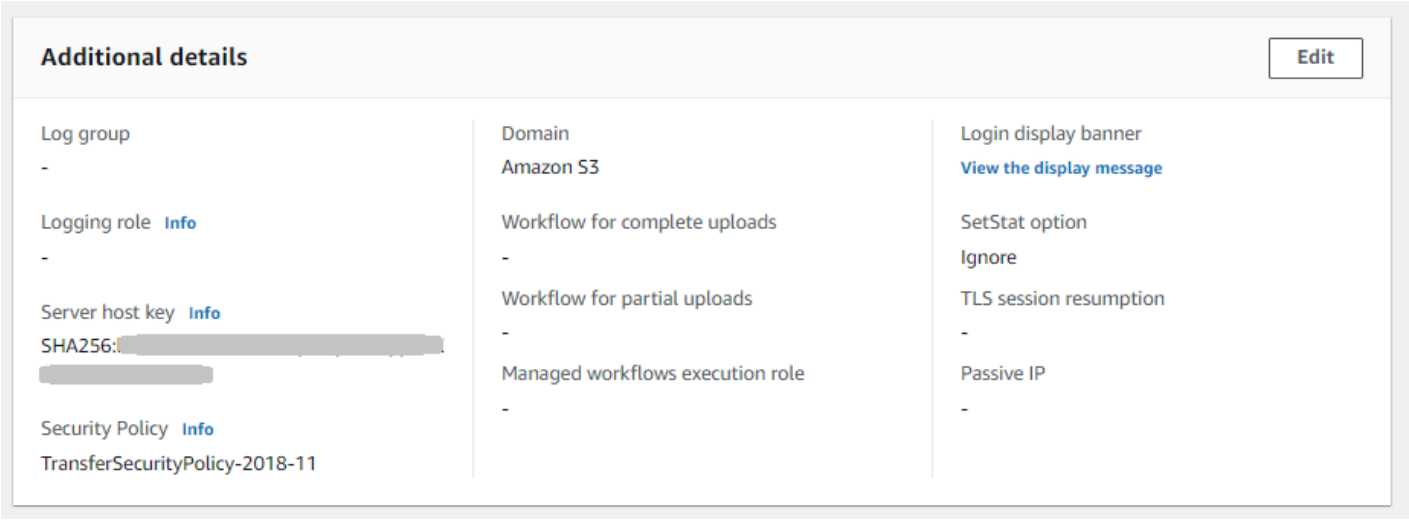


Melihat konfigurasi server

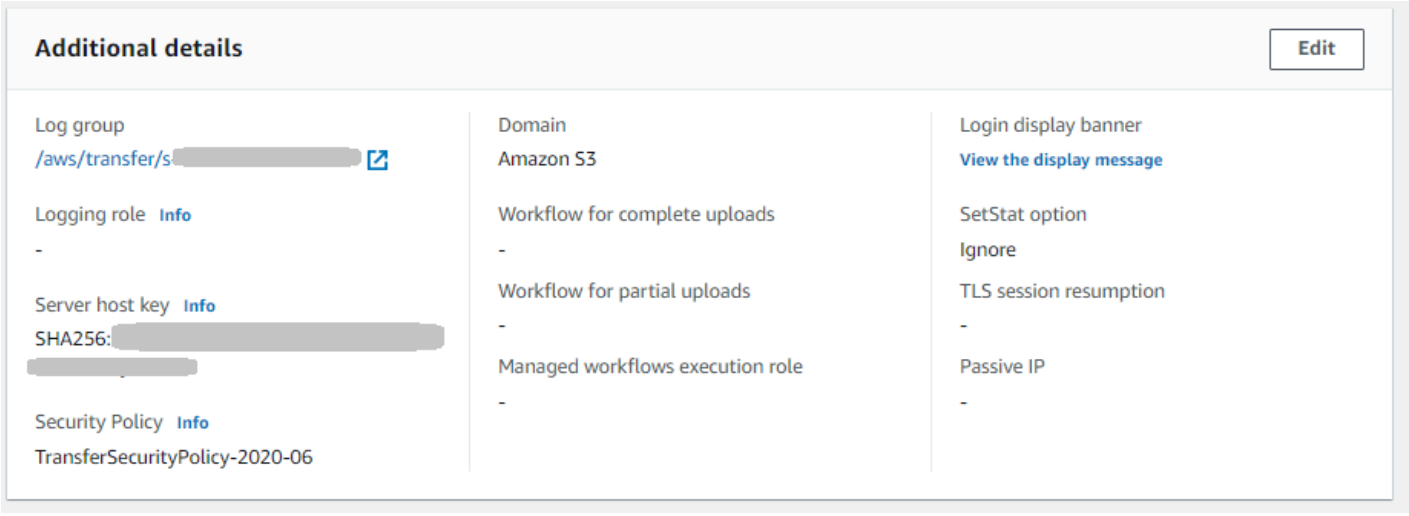
Detail untuk halaman konfigurasi server bergantung pada skenario Anda:

Bergantung pada skenario Anda, halaman konfigurasi server mungkin terlihat seperti salah satu contoh berikut:

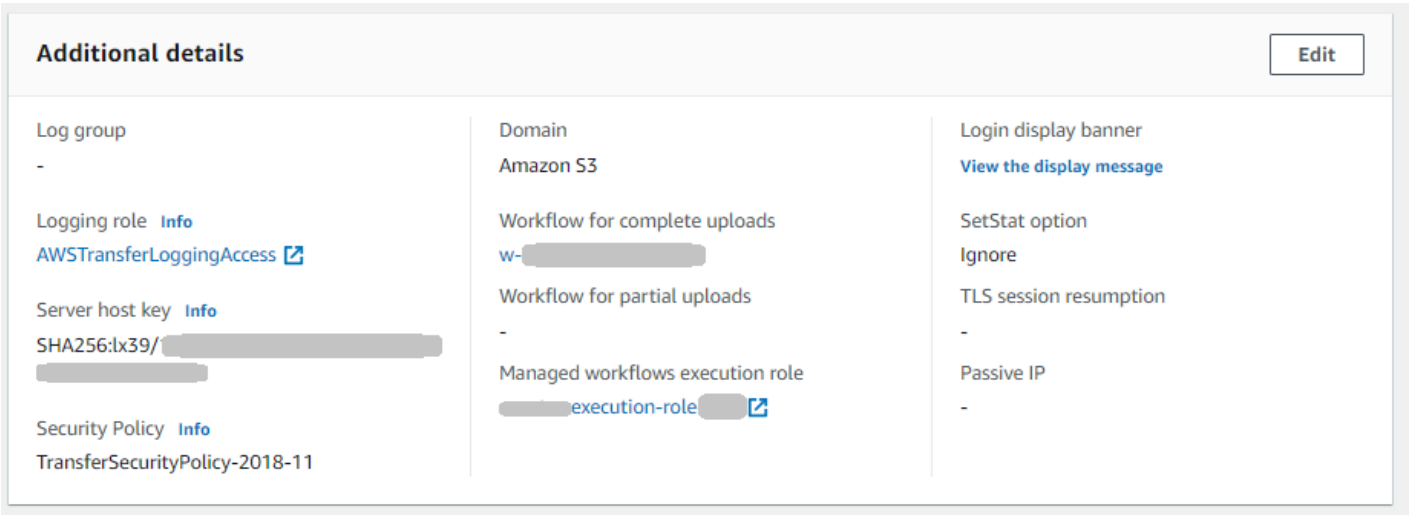
- Tidak ada logging yang diaktifkan.



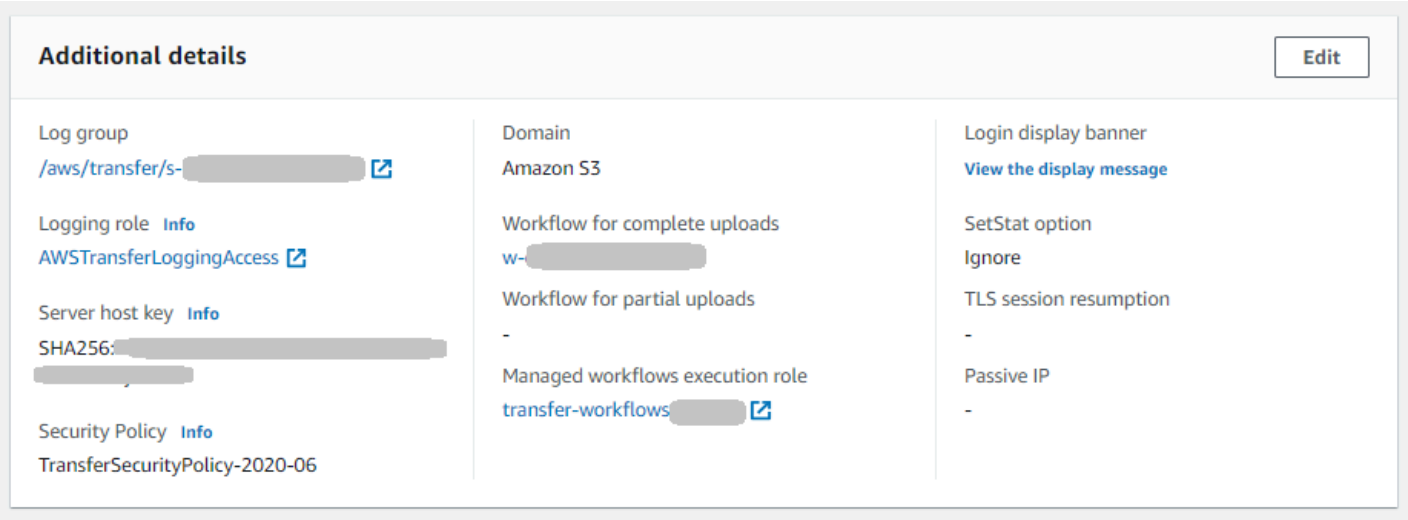
- Pencatatan JSON terstruktur diaktifkan.



- Peran logging diaktifkan, tetapi logging JSON terstruktur tidak diaktifkan.



- Kedua jenis logging (peran logging dan logging JSON terstruktur) diaktifkan.



Mengelola logging untuk alur kerja

CloudWatch menyediakan audit dan pencatatan terkonsolidasi untuk kemajuan dan hasil alur kerja. Selain itu, AWS Transfer Family menyediakan beberapa metrik untuk alur kerja. Anda dapat melihat metrik berapa banyak eksekusi alur kerja yang dimulai, diselesaikan dengan sukses, dan gagal pada menit sebelumnya. Semua CloudWatch metrik untuk Transfer Family dijelaskan dalam[Menggunakan CloudWatch metrik untuk Transfer Family](#).

Lihat CloudWatch log Amazon untuk alur kerja

1. Buka CloudWatch konsol Amazon di https://console.aws.amazon.com/cloudwatch/.
2. Di panel navigasi kiri, pilih Log, lalu pilih Grup log.
3. Pada halaman Grup log, pada bilah navigasi, pilih Wilayah yang benar untuk AWS Transfer Family server Anda.
4. Pilih grup log yang sesuai dengan server Anda.

   Misalnya, jika ID server Anda`s-1234567890abcdef0`, grup log Anda adalah`/aws/transfer/s-1234567890abcdef0`.
5. Pada halaman detail grup log untuk server Anda, aliran log terbaru ditampilkan. Ada dua aliran log untuk pengguna yang Anda jelajahi:

   - Satu untuk setiap sesi Secure Shell (SSH) File Transfer Protocol (SFTP).

- Satu untuk alur kerja yang sedang dijalankan untuk server Anda. Format untuk aliran log untuk alur kerja adalah*username*.*workflowID*.*uniqueStreamSuffix*.

Misalnya, jika pengguna Andamary-major, Anda memiliki aliran log berikut:

```
mary-major-east.1234567890abcdef0
mary.w-abcdef01234567890.021345abcdef6789
```

> **Note**
>
> Pengidentifikasi alfanumerik 16 digit yang tercantum dalam contoh ini adalah fiktif. Nilai yang Anda lihat di Amazon CloudWatch berbeda.

Halaman peristiwa Log untuk mary-major-usa-east.1234567890abcdef0 menampilkan detail untuk setiap sesi pengguna, dan aliran mary.w-abcdef01234567890.021345abcdef6789 log berisi detail untuk alur kerja.

Berikut ini adalah contoh aliran log untukmary.w-abcdef01234567890.021345abcdef6789, berdasarkan alur kerja (w-abcdef01234567890) yang berisi langkah salin.

```
{
    "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"
    }
}
```

Konfigurasikan peran CloudWatch logging

Untuk menetapkan akses, Anda membuat kebijakan IAM berbasis sumber daya dan peran IAM yang menyediakan informasi akses tersebut.

Untuk mengaktifkan CloudWatch pencatatan Amazon, Anda mulai dengan membuat kebijakan IAM yang memungkinkan CloudWatch pencatatan log. Anda kemudian membuat peran IAM dan melampirkan kebijakan ke dalamnya. Anda dapat melakukan ini ketika Anda membuat server atau dengan mengedit server yang ada. Untuk informasi selengkapnya CloudWatch, lihat Apa itu Amazon CloudWatch? dan Apa itu CloudWatch log Amazon? di Panduan CloudWatch Pengguna Amazon.

Gunakan contoh berikut kebijakan IAM untuk mengizinkan CloudWatch pencatatan.

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:Akun AWS:log-group:/aws/transfer/*"
        }
    ]
}
```

Dalam kebijakan contoh sebelumnya, untuk**Resource**, ganti *region-id* dan *Akun AWS*dengan nilai Anda. Misalnya, **"Resource": "arn:aws::logs:us-east-1:111122223333:log-group:/aws/transfer/*"**

Anda kemudian membuat peran dan melampirkan kebijakan CloudWatch Log yang Anda buat.

Untuk membuat peran IAM dan melampirkan kebijakan

1. Di panel navigasi, pilih Peran, lalu pilih Buat peran.

   Pada halaman Buat peran, pastikan bahwa AWS layanan dipilih.

2. Pilih Transfer dari daftar layanan, lalu pilih Berikutnya: Izin. Ini membangun hubungan kepercayaan antara AWS Transfer Family dan peran IAM. Selain itu, tambahkan `aws:SourceAccount` dan `aws:SourceArn` kondisikan kunci untuk melindungi diri Anda dari masalah wakil yang membingungkan. Lihat dokumentasi berikut untuk lebih jelasnya:

   - Prosedur untuk membangun hubungan kepercayaan dengan AWS Transfer Family: Untuk membangun hubungan kepercayaan
   - Deskripsi untuk masalah wakil yang bingung: masalah wakil yang bingung

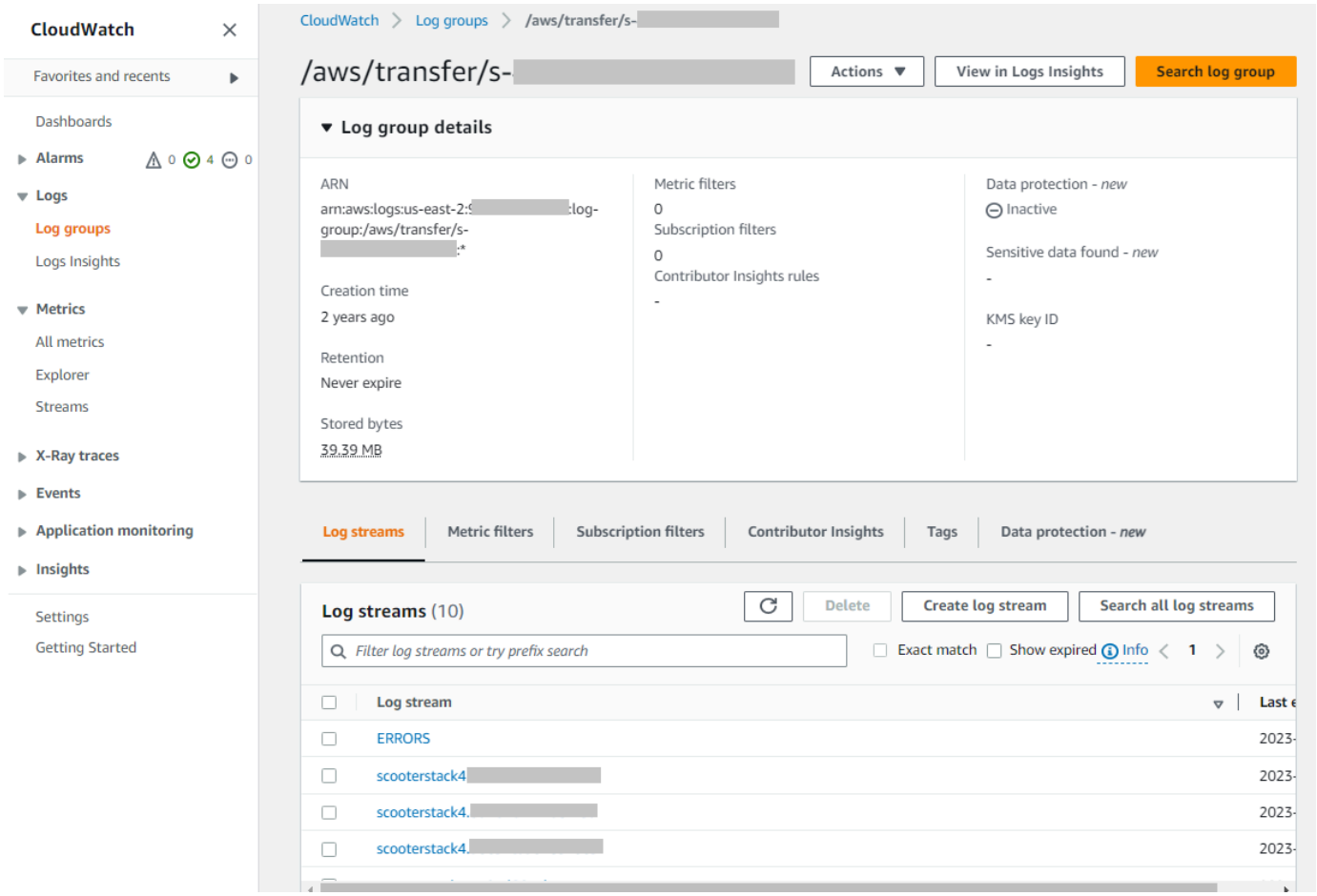
3. Di bagian Lampirkan kebijakan izin, cari dan pilih kebijakan CloudWatch Log yang baru saja Anda buat, lalu pilih Berikutnya: Tag.
4. (Opsional) Masukkan kunci dan nilai untuk tag, dan pilih Berikutnya: Tinjau.
5. Pada halaman Tinjauan, masukkan nama dan deskripsi untuk peran baru Anda, lalu pilih Buat peran.
6. Untuk melihat log, pilih ID Server untuk membuka halaman konfigurasi server, dan pilih Lihat log. Anda diarahkan ke CloudWatch konsol tempat Anda dapat melihat aliran log Anda.

Pada CloudWatch halaman server Anda, Anda dapat melihat catatan otentikasi pengguna (keberhasilan dan kegagalan), unggahan data (`PUT`operasi), dan unduhan data (`GET`operasi).

Melihat aliran log Transfer Family

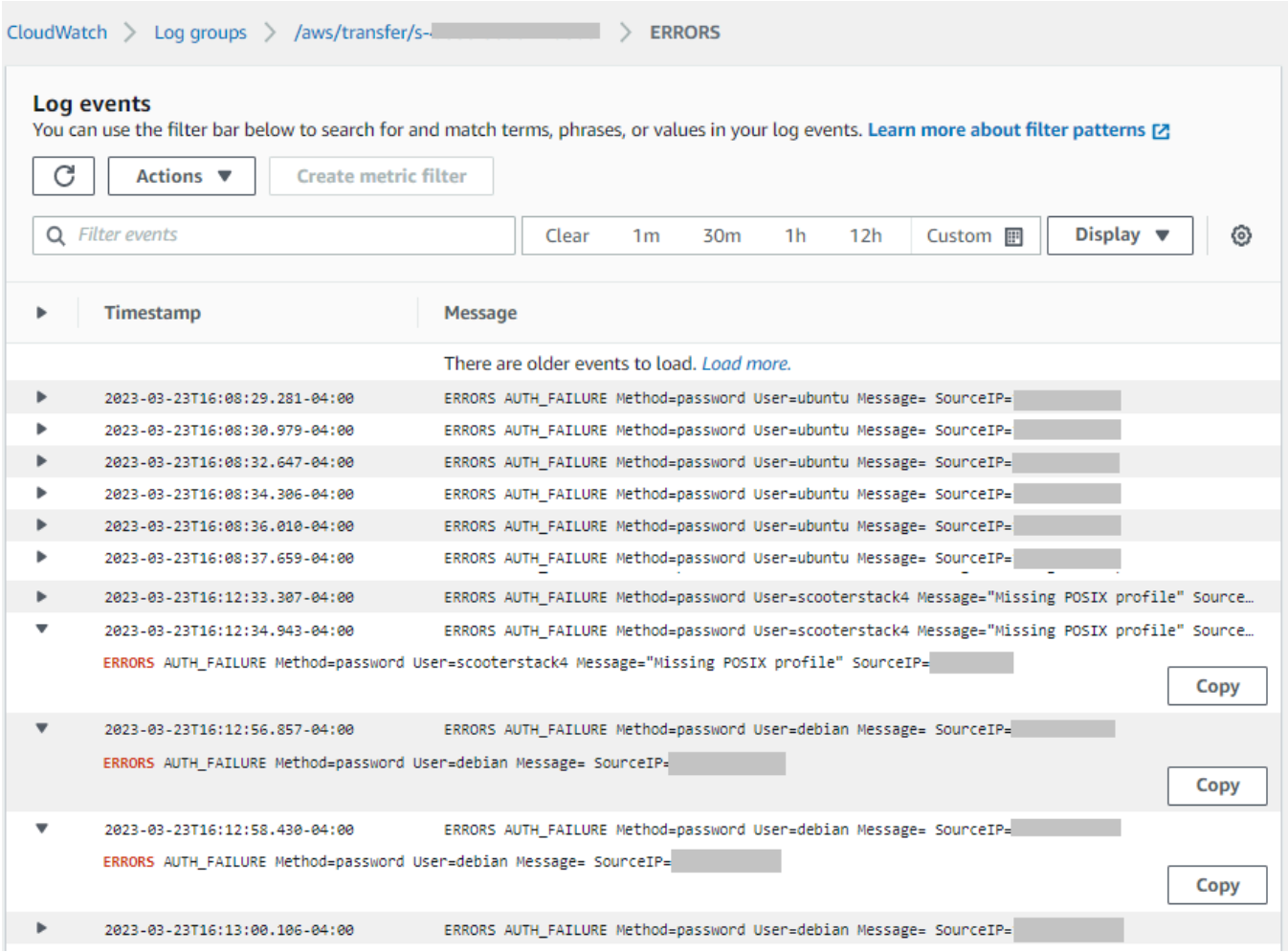
Untuk melihat log server Transfer Family

1. Arahkan ke halaman detail untuk server.
2. Pilih Lihat log. Ini membuka Amazon CloudWatch.
3. Grup log untuk server yang Anda pilih akan ditampilkan.

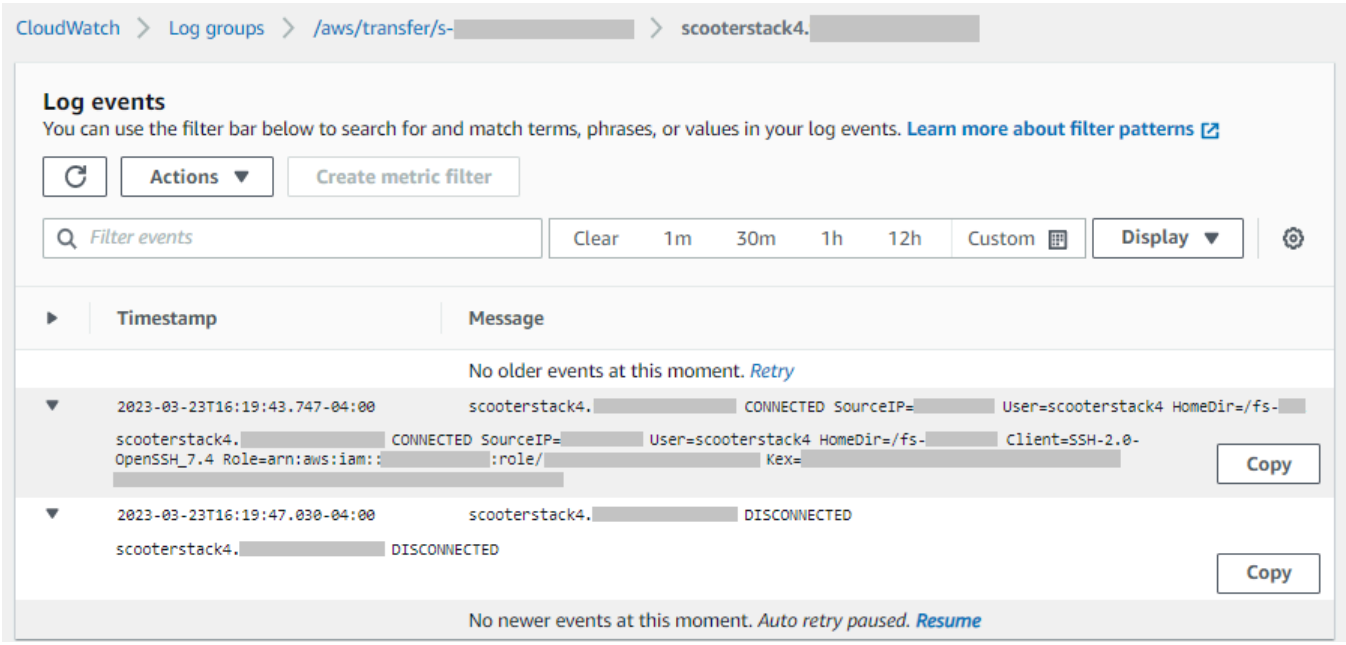


4. Anda dapat memilih aliran log untuk menampilkan detail dan entri individual untuk aliran.

   - Jika ada daftar untuk ERROR, Anda dapat memilihnya untuk melihat detail kesalahan terbaru untuk server.

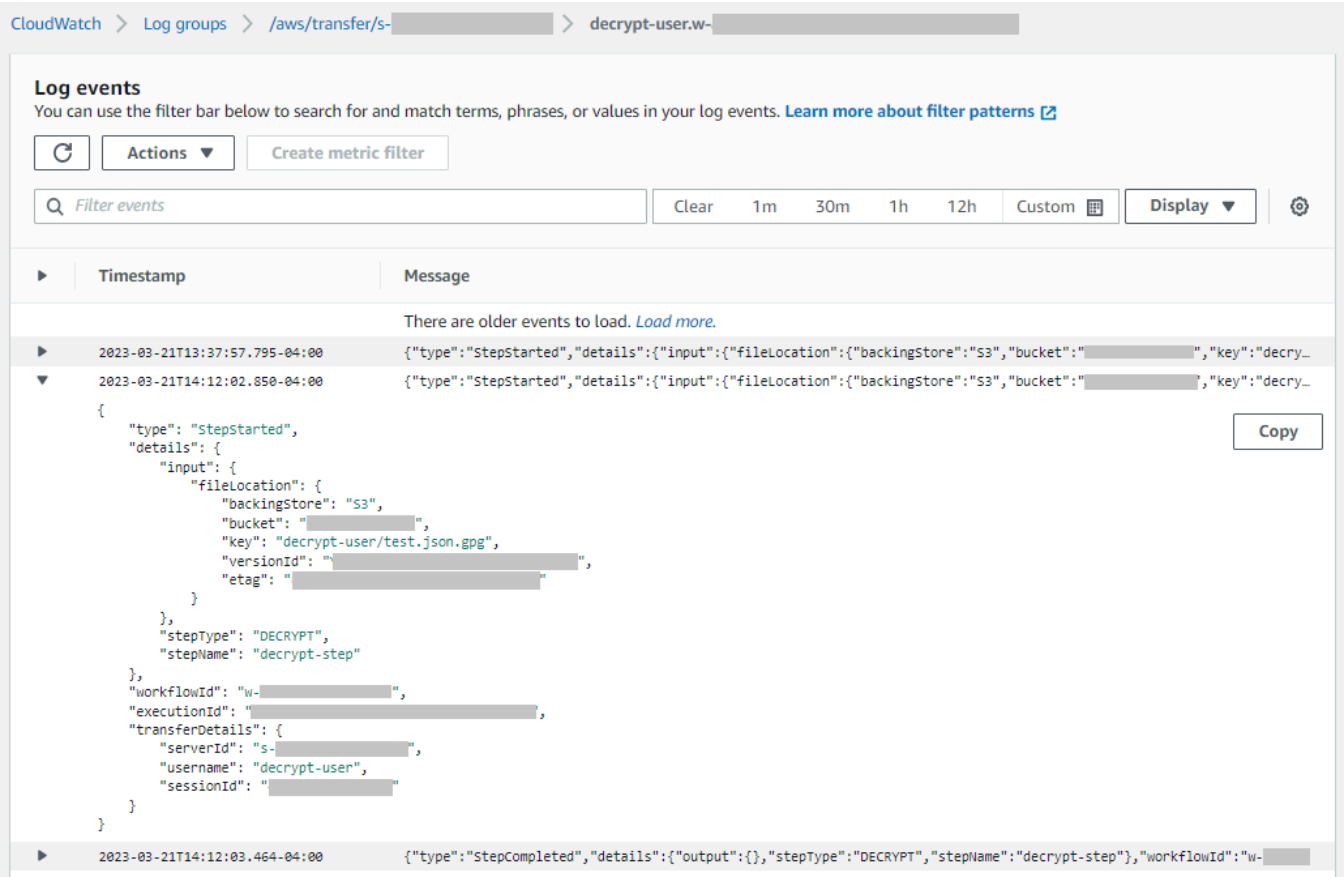


- Pilih entri lain untuk melihat contoh aliran log.



- Jika server Anda memiliki alur kerja terkelola yang terkait dengannya, Anda dapat melihat log untuk menjalankan alur kerja.

> **Note**
>
> Format untuk aliran log untuk alur kerja adalah*username*.*workflowId*.*uniqueStreamSuffix*. Misalnya, decrypt-user.w-a1111222233334444.aaaa1111bbbb2222 bisa menjadi nama aliran log untuk pengguna dan alur kerja. **`decrypt-user w-a1111222233334444`**



> **Note**
>
> Untuk entri log yang diperluas, Anda dapat menyalin entri ke clipboard dengan memilih Salin. Untuk detail selengkapnya tentang CloudWatch log, lihat Melihat data log.

Membuat CloudWatch alarm Amazon

Contoh berikut menunjukkan cara membuat CloudWatch alarm Amazon menggunakan AWS Transfer Family metrik,`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
```

Mencatat panggilan API Amazon S3 ke log akses S3

Jika Anda menggunakan log akses Amazon S3 untuk mengidentifikasi permintaan S3 yang dibuat atas nama pengguna transfer file Anda, `RoleSessionName` digunakan untuk menampilkan peran

IAM mana yang diasumsikan untuk melayani transfer file. Ini juga menampilkan informasi tambahan seperti nama pengguna, id sesi, dan server-id yang digunakan untuk transfer. Formatnya adalah `[AWS:Role Unique Identifier]/username.sessionid@server-id` dan terkandung dalam bidang Pemohon. Misalnya, berikut ini adalah konten untuk bidang Peminta sampel dari log akses S3 untuk file yang disalin ke bucket S3.

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

Di bidang Pemohon di atas, ini menunjukkan Peran IAM yang disebut. `IamRoleName` Untuk informasi selengkapnya tentang pengidentifikasi unik peran IAM, lihat Pengidentifikasi unik di Panduan Pengguna.AWS Identity and Access Management

Contoh untuk membatasi masalah wakil yang membingungkan

Masalah deputi yang bingung adalah masalah keamanan di mana entitas yang tidak memiliki izin untuk melakukan tindakan dapat memaksa entitas yang lebih istimewa untuk melakukan tindakan. Pada tahun AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil yang membingungkan. Untuk detail selengkapnya, lihat Pencegahan confused deputy lintas layanan.

> **Note**
>
> Dalam contoh-contoh berikut, ganti setiap *`placeholder input pengguna`* dengan informasi Anda sendiri.
> Dalam contoh ini, Anda dapat menghapus detail ARN untuk alur kerja jika server Anda tidak memiliki alur kerja yang melekat padanya.

Contoh kebijakan logging/pemanggilan berikut memungkinkan server apa pun (dan alur kerja) di akun untuk mengambil peran.

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

Contoh kebijakan logging/pemanggilan berikut memungkinkan server tertentu (dan alur kerja) untuk mengambil peran.

```
{
    "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 struktur log untuk Transfer Family

Topik ini menjelaskan bidang yang diisi dalam log Transfer Family: baik untuk entri log terstruktur JSON maupun entri log lama.

Topik

- Log terstruktur JSON untuk Transfer Family
- Log lama untuk Transfer Family

Log terstruktur JSON untuk Transfer Family

Tabel berikut berisi rincian untuk bidang entri log untuk tindakan Transfer Family SFTP/FTP/FTPS, dalam format log terstruktur JSON yang baru.

Bidang	Deskripsi	Entri contoh
activity-type	The action by the user	BUKA | TUTUP | PARTIAL_CLOSE | TERPUTUS | TERHUBUNG
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 Algoritma kriptografi)	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	/user-home-bucket/test

Bidang	Deskripsi	Entri contoh
	have a logical home directory, this value is always `/`	
kex	Specifies the negotiated SSH key exchange (KEX) for the connection (available KEX are listed in Algoritma kriptografi)	diffie-hellman-group14-sha256
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-timur laut-1:12346789012: server/s-1234567890akeu2js2
role	The IAM role of the user	arn:aws:iam: :0293883675: peran/testuser-role
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

Bidang	Deskripsi	Entri contoh
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

Log lama untuk Transfer Family

Tabel berikut berisi rincian untuk entri log untuk berbagai tindakan Transfer Family.

> **Note**
>
> Entri ini tidak dalam format log terstruktur JSON yang baru.

Tabel berikut berisi rincian untuk entri log untuk berbagai tindakan Transfer Family, dalam format log terstruktur JSON yang baru.

Tindakan	Log yang sesuai dalam CloudWatch Log Amazon
Kegagalan otentikasi	KESALAHAN AUTH_FAILURE METHOD=PublicKey User="RSA SHA256:lfz3r2nmly4rak+b7rb1rsvuibae+a+hxg0c7l1jiZ0" sourceIP=3.8.172.211
SALIN/TANDA/HAPUS/DEKRIPSI alur kerja	{"type”:” StepStarted “, "details": {"input”: {"FileLocation”: {"backingStore” :"EFS”, "fileSystemId” :"fs-12345678", "path”:” /lhr/regex.py “}}, "stepType” :"TAG”, "stepName” :"successful_tag_step "}, "workFlowid” :"workFlowid” :"successful_tag_step "}, "workFlowid” :"workFlowid” :"stepName” :"successful_tag_step "}, "workFlowid” :"workFlowid” :"stepName” :"successful_tag_step

Tindakan	Log yang sesuai dalam CloudWatch Log Amazon
	"}, "workFlowid” :"workFlowid” :"stepNam e” "w-1111aaaa22bbbb3", "Executio nID” :"81234abcd-1234-efgh-5678-ijklmnopq r90", "transferDetails”: {"serverID” :"s-1234a bcd5678efghi”, "nama pengguna” :"lhr”, "sessionID” :"123456767890abcdef0"}}
Alur kerja langkah kustom	{"type”:” CustomStepInvoked “, "details”: {"output”: {"token” :"MZM4MjG5YWUTYT EzMy 00 YjIz LWI3OG MtYz U4OGI2 ZjQyMz E5"}, "StepType” :"CUSTOM”, "stepName ” :"efs-s3_copy_2"}, "workFlowid” :"w-9283e 49d33297c3f7", "executionID” :"1234abc d-1234-efgh-5678-ijklmnopqr90", "transfer Details”: {"serverID” :"s-zzzzzz11aaaa22 223", "username” :"lhr”, "sessionID” :"1234567 890abcdef0"}}
Menghapus	lhr.33a8fb495ffb383b HAPUS PATH=/ember/ pengguna/123.jpg
Unduh	lhr.33a8fb495ffb383b JALUR TERBUKA =/ ember/pengguna/mode 123.jpg =Baca llhr.33a8fb495ffb383b JALUR TUTUP =/ember/ pengguna/123.jpg = 3618546 BytesOut
Login/Logout	user.914984e553bcddb6 SUMBER TERHUBUNG = 1.22.111.222 pengguna = LHR = KLIEN LOGIS = SSH-2.0-openssh_7.4 peran = arn:aws: :iam: :123456789012: peran/sftp-s3-akses HomeDir user.914984e553bcddb6 TERPUTUS

Tindakan	Log yang sesuai dalam CloudWatch Log Amazon
Mengganti nama	lhr.33a8fb495ffb383b GANTI NAMA PATH=/bucket/user/lambo.png =/ember/pengguna/ferrari.png NewPath
Contoh log kesalahan alur kerja	{"type”:” StepErrored “, "details”: {"ErrorType” :"BAD_REQUEST”, "ErrorMessage” :"Tidak dapat menandai berkas Efs”, "stepType ” :"stepName” :"successful_tag_step "}, "workFlowid” :"w-1234abcd5678efghi”, "ExecutionID” :"81234abcd5678efghi”, "ExecutionID” :"81234abcd5678efghi” :"ExecutionID” :"81234abcd5678efghi” :"ExecutionID” :"81234abcd5678efghi” :"ExecutionId” :"81234abcd534abcd-1234-efgh-5678-ijklmnopqr90", "transferDetails”: {"serverID” :"s-1234abcd5678efghi”, "namapengguna” :"lhr”, "sessionID” :"1234567890abcdef0"}}
Symlink	lhr.eb49cf7b8651e6d5 CREATE_SYMLINK =/fs-12345678/lhr/pqr.jpg =abc.jpg LinkPath TargetPath
Unggahan	lhr.33a8fb495ffb383b JALUR TERBUKA =/bucket/pengguna/mode 123.jpg =Buat | memotong|tulis lhr.33a8fb495ffb383b JALUR TUTUP =/ember/pengguna/123.jpg = 3618546 BytesIn

Tindakan	Log yang sesuai dalam CloudWatch Log Amazon
Alur Kerja	{"type”:” ExecutionStarted “, "details”: {"input”: {” “: {" backingStore” :"EFS”, "fileSyst emId” :"fs-12345678", "path”:” /lhr/regex.py initialFileLocation “}}}, "workFlowid” :"w-1111a aaa22bbbb3", "ExecutionId” :"1234abcd-1234- efgh-5678-ijklmnopqr90", "transferDetails”: {"serverID” :"s-zzzz1111aaaa22223", "nama pengguna” :"lhr”, "sessionID” :"1234567 890abcdef0"}} {"type”:” StepStarted “, "details”: {"input”: {"FileLocation”: {"backingStore” :"EFS”, "fileSyst emId” :"fs-12345678", "path”:” /lhr/regex.py “}}, "stepType” :"CUSTOM”, "stepName ” :"efs-s3_copy_2"}, "workFlowid” :"w-9283e 49d33297c3f7", "ExecutionID” :"1234abc d-1234-efgh-5678-ijklmnopqr90", "transfer Details”: {"serverID” :"s-18ca49dce5d842 e0b”, "username” :"lhr”, "sessionID” :"1234567 890abcdef0"}}

Contoh entri CloudWatch log

Topik ini menyajikan contoh entri log.

Topik

- Contoh entri log sesi transfer
- Contoh entri log untuk konektor SFTP
- Contoh entri log untuk kegagalan algoritma pertukaran Kunci

Contoh entri log sesi transfer

Dalam contoh ini, pengguna SFTP terhubung ke server Transfer Family, mengunggah file, lalu memutuskan sambungan dari sesi.

Entri log berikut mencerminkan pengguna SFTP yang terhubung ke server Transfer Family.

```
{
   "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"
}
```

Entri log berikut mencerminkan pengguna SFTP yang mengunggah file ke bucket Amazon S3 mereka.

```
{
   "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"
}
```

Entri log berikut mencerminkan pengguna SFTP yang terputus dari sesi SFTP mereka. Pertama, klien menutup koneksi ke bucket, dan kemudian klien memutuskan sesi SFTP.

```
{
   "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"
}
```

Contoh entri log untuk konektor SFTP

Bagian ini berisi contoh log untuk transfer yang berhasil dan tidak berhasil. Log dihasilkan ke grup log bernama`/aws/transfer/`*`connector-id`*, di mana *`connector-id`* adalah pengenal untuk konektor SFTP Anda.

> **Note**
>
> Entri log untuk konektor SFTP hanya dihasilkan saat Anda menjalankan perintah. `StartFileTransfer`

Entri log ini untuk transfer yang berhasil diselesaikan.

```
{
    "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"
}
```

Entri log ini untuk transfer yang waktunya habis, dan dengan demikian tidak berhasil diselesaikan.

```
{
    "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"
}
```

Deskripsi untuk beberapa bidang kunci dalam contoh log sebelumnya.

- `timestamp`mewakili saat log ditambahkan ke CloudWatch. `start-time`dan `end-time` sesuai dengan kapan konektor benar-benar memulai dan menyelesaikan transfer.
- `transfer-id`adalah pengidentifikasi unik yang ditetapkan untuk setiap `start-file-transfer` permintaan. Jika pengguna melewati beberapa jalur file dalam satu panggilan `start-file-transfer` API, semua file berbagi yang sama`transfer-id`.
- `file-transfer-id`adalah nilai unik yang dihasilkan untuk setiap file yang ditransfer. Perhatikan bahwa bagian awal `file-transfer-id` adalah sama dengan`transfer-id`.

Contoh entri log untuk kegagalan algoritma pertukaran Kunci

Bagian ini berisi contoh log di mana algoritma pertukaran Kunci (KEX) gagal. Ini adalah contoh dari aliran log ERRORS untuk log terstruktur.

Entri log ini adalah contoh di mana ada kesalahan jenis kunci host.

```
{
    "activity-type": "KEX_FAILURE",
    "source-ip": "999.999.999.999",
    "resource-arn": "arn:aws:transfer:us-east-1:999999999999:server/
s-99999999999999999",
    "message": "no matching 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"
}
```

Entri log ini adalah contoh di mana ada ketidakcocokan KEX.

```
{
    "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"
}
```

Menggunakan CloudWatch metrik untuk Transfer Family

> **Note**
>
> Anda juga bisa mendapatkan metrik untuk Transfer Family dari dalam konsol Transfer Family itu sendiri. Untuk detailnya, lihat Memantau penggunaan di konsol

Anda bisa mendapatkan informasi tentang server Anda menggunakan CloudWatch metrik. Metrik mewakili kumpulan titik data yang diurutkan waktu yang dipublikasikan ke CloudWatch. Saat menggunakan metrik, Anda harus menentukan namespace Transfer Family, nama metrik, dan dimensi. Untuk informasi selengkapnya tentang metrik, lihat Metrik di CloudWatch Panduan Pengguna Amazon.

Tabel berikut menjelaskan CloudWatch metrik untuk Transfer Family.

Namespace	Metrik	Deskripsi
`AWS/Transfer`	`BytesIn`	Jumlah total byte yang ditransfer ke server. Unit: Hitungan Periode: 5 menit

Namespace	Metrik	Deskripsi
	`BytesOut`	Jumlah total byte yang ditransfer keluar dari server. Unit: Jumlah Periode: 5 menit
	`FilesIn`	Jumlah total file yang ditransfer ke server. Untuk server yang menggunakan protokol AS2, metrik ini mewakili jumlah pesan yang diterima. Unit: Hitungan Periode: 5 menit
	`FilesOut`	Jumlah total file yang ditransfer keluar dari server. Unit: Hitungan Periode: 5 menit
	`InboundMessage`	Jumlah total pesan AS2 yang berhasil diterima dari mitra dagang. Unit: Hitungan Periode: 5 menit
	`InboundFailedMessage`	Jumlah total pesan AS2 yang tidak berhasil diterima dari mitra dagang. Artinya, mitra dagang mengirim pesan, tetapi server Transfer Family tidak berhasil memproses nya. Unit: Hitungan Periode: 5 menit

Namespace	Metrik	Deskripsi
	`OnPartialUploadExecutionsStarted`	Jumlah total eksekusi on-partial-upload alur kerja dimulai di server. Unit: Hitungan Periode: 1 menit
	`OnPartialUploadExecutionsSuccess`	Jumlah total eksekusi on-partial-upload alur kerja yang berhasil di server. Unit: Hitungan Periode: 1 menit
	`OnPartialUploadExecutionsFailed`	Jumlah total eksekusi on-partial-upload alur kerja yang gagal di server. Unit: Hitungan Periode: 1 menit
	`OnUploadExecutionsStarted`	Jumlah total eksekusi alur kerja dimulai di server. Unit: Hitungan Periode: 1 menit
	`OnUploadExecutionsSuccess`	Jumlah total eksekusi alur kerja yang berhasil di server. Unit: Hitungan Periode: 1 menit
	`OnUploadExecutionsFailed`	Jumlah total eksekusi alur kerja yang gagal di server. Unit: Hitungan Periode: 1 menit

Dimensi Transfer Family

Dimensi adalah pasangan nama/nilai yang merupakan bagian dari identitas metrik. Untuk informasi selengkapnya tentang dimensi, lihat Dimensi di Panduan CloudWatch Pengguna Amazon.

Tabel berikut menjelaskan CloudWatch dimensi Transfer Family.

Dimensi	Deskripsi
`ServerId`	ID unik dari server.

Menggunakan Notifikasi Pengguna AWS dengan AWS Transfer Family

Untuk mendapatkan pemberitahuan tentang AWS Transfer Family acara, Anda dapat menggunakan Notifikasi Pengguna AWSuntuk mengatur berbagai saluran pengiriman. Jika acara cocok dengan aturan yang Anda tentukan, Anda menerima pemberitahuan.

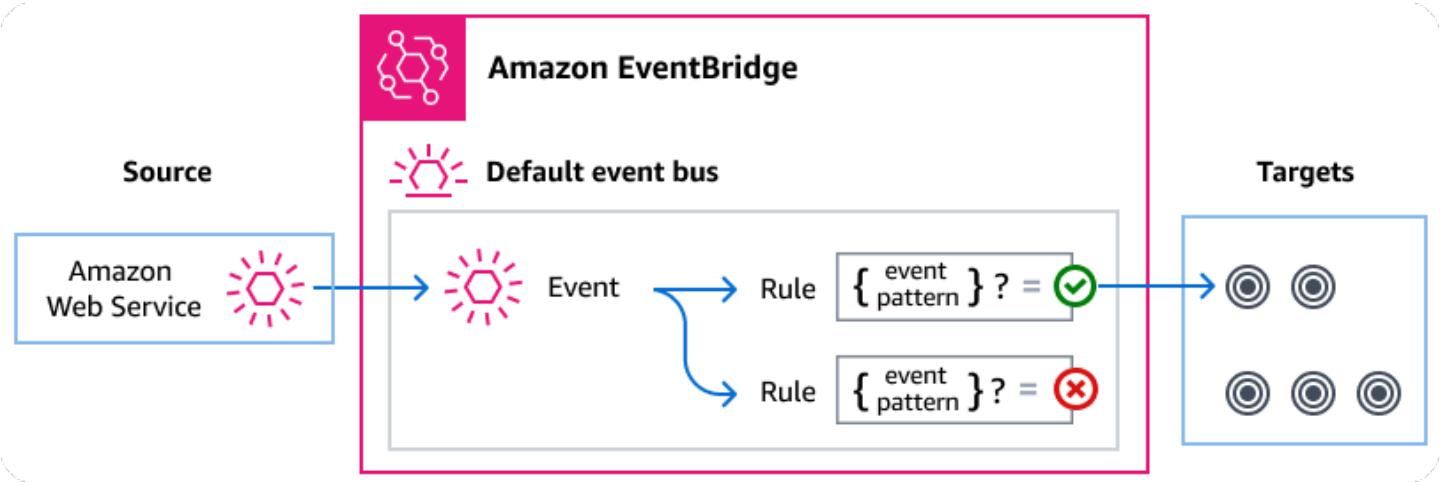
Anda dapat menerima notifikasi untuk peristiwa melalui beberapa saluran, termasuk email, notifikasi obrolan AWS Chatbot, atau notifikasi push AWS Console Mobile Application. Anda juga dapat melihat notifikasi di Pusat Pemberitahuan Konsol. Notifikasi Pengguna mendukung agregasi, yang dapat mengurangi jumlah pemberitahuan yang Anda terima selama acara tertentu.

Untuk informasi selengkapnya, lihat Menyesuaikan pemberitahuan pengiriman file menggunakan posting blog alur kerja AWS Transfer Family terkelola, dan Apa itu? Notifikasi Pengguna AWS dalam Notifikasi Pengguna AWS User Guide.

Mengelola Transfer Family acara menggunakan Amazon EventBridge

Amazon EventBridge adalah layanan tanpa server yang menggunakan peristiwa untuk menghubungkan komponen aplikasi bersama-sama, yang dapat memudahkan Anda membangun aplikasi berbasis peristiwa yang dapat diskalakan. Arsitektur berbasis peristiwa adalah gaya membangun sistem perangkat lunak yang digabungkan secara longgar yang bekerja sama dengan memancarkan dan menanggapi peristiwa. Peristiwa mewakili perubahan dalam sumber daya atau lingkungan.

Seperti banyak AWS layanan, Transfer Family menghasilkan dan mengirim acara ke bus acara EventBridge default. Perhatikan bahwa bus acara default secara otomatis disediakan di setiap AWS akun. Bus acara adalah router yang menerima acara dan mengirimkannya ke nol atau lebih tujuan, atau target. Anda menentukan aturan untuk bus acara yang mengevaluasi peristiwa saat mereka tiba. Setiap aturan memeriksa apakah suatu peristiwa cocok dengan pola acara aturan. Jika acara cocok, bus acara mengirimkan acara ke satu atau lebih target yang ditentukan.



Topik

- Transfer Family acara
- Mengirim Transfer Family acara dengan menggunakan EventBridge aturan
- Amazon EventBridge izin
- EventBridge Sumber daya tambahan
- Transfer Family referensi detail acara

Transfer Family acara

Transfer Family secara otomatis mengirimkan acara ke bus EventBridge acara default. Anda dapat membuat aturan pada bus acara di mana setiap aturan mencakup pola acara dan satu atau lebih target. Peristiwa yang cocok dengan pola acara aturan dikirimkan ke target yang ditentukan berdasarkan upaya terbaik, namun, beberapa peristiwa mungkin dikirimkan secara tidak berurutan.

Peristiwa berikut dihasilkan oleh Transfer Family. Untuk informasi selengkapnya, lihat EventBridge peristiwa di Panduan Amazon EventBridge Pengguna.

Acara server SFTP, FTPS, dan FTP

Jenis detail acara	Deskripsi
Unduhan Server File FTP Selesai	Sebuah file telah berhasil diunduh untuk protokol FTP.
Unduhan Server File FTP Gagal	Upaya untuk mengunduh file gagal untuk protokol FTP.
Unggahan Server File FTP Selesai	Sebuah file telah berhasil diunggah untuk protokol FTP.
Unggahan Server File FTP Gagal	Upaya untuk mengunggah file gagal untuk protokol FTP.
Unduhan Server File FTPS Selesai	Sebuah file telah berhasil diunduh untuk protokol FTPS.
Unduhan Server File FTPS Gagal	Upaya untuk mengunduh file gagal untuk protokol FTPS.
Unggahan Server File FTPS Selesai	Sebuah file telah berhasil diunggah untuk protokol FTPS.
Unggahan Server File FTPS Gagal	Upaya untuk mengunggah file gagal untuk protokol FTPS.

Jenis detail acara	Deskripsi
Unduhan File Server SFTP Selesai	Sebuah file telah berhasil diunduh untuk protokol SFTP.
Unduhan File Server SFTP Gagal	Upaya untuk mengunduh file gagal untuk protokol SFTP.
Unggahan File Server SFTP Selesai	Sebuah file telah berhasil diunggah untuk protokol SFTP.
Unggahan File Server SFTP Gagal	Upaya untuk mengunggah file gagal untuk protokol SFTP.

Acara konektor SFTP

Jenis detail acara	Deskripsi
Kirim File Konektor SFTP Selesai	Transfer file dari konektor ke server SFTP jarak jauh telah berhasil diselesaikan.
Kirim File Konektor SFTP Gagal	Transfer file dari konektor ke server SFTP jarak jauh telah gagal.
Pengambilan File Konektor SFTP Selesai	Transfer file dari server SFTP jarak jauh ke konektor telah berhasil diselesaikan.
Pengambilan File Konektor SFTP Gagal	Transfer file dari server SFTP jarak jauh ke konektor gagal.

Acara A2S

Jenis detail acara	Deskripsi
Terima Muatan AS2 Selesai	Muatan untuk pesan AS2 telah diterima.

Jenis detail acara	Deskripsi
Penerimaan Muatan AS2 Gagal	Muatan untuk pesan AS2 belum diterima.
Kirim Muatan AS2 Selesai	Payload untuk pesan AS2 telah berhasil dikirim.
Kirim Muatan AS2 Gagal	Muatan untuk pesan AS2 gagal dikirim.
AS2 MDN Terima Selesai	Pemberitahuan disposisi pesan untuk pesan AS2 telah diterima.
AS2 MDN Menerima Gagal	Pemberitahuan disposisi pesan untuk pesan AS2 belum diterima.
AS2 MDN Kirim Selesai	Pemberitahuan disposisi pesan untuk pesan AS2 telah berhasil dikirim.
AS2 MDN Kirim Gagal	Pemberitahuan disposisi pesan untuk pesan AS2 gagal dikirim.

Mengirim Transfer Family acara dengan menggunakan EventBridge aturan

Jika Anda ingin bus acara EventBridge default mengirim Transfer Family acara ke target, Anda harus membuat aturan yang berisi pola peristiwa yang cocok dengan data dalam Transfer Family acara yang Anda inginkan.

Anda dapat membuat aturan dengan mengikuti langkah-langkah umum ini:

1. Buat pola acara untuk aturan yang menentukan berikut:
   - Transfer Family adalah sumber peristiwa yang dievaluasi oleh aturan.
   - (Opsional) Setiap data acara lain untuk mencocokkannya.

   Untuk informasi selengkapnya, lihat ???.
2. (Opsional) Buat transformator input yang menyesuaikan data dari peristiwa sebelum EventBridge mengirim informasi ke target aturan.

   Untuk informasi selengkapnya, lihat Transformasi input di Panduan EventBridge Pengguna.
3. Tentukan target yang EventBridge ingin Anda sampaikan acara yang cocok dengan pola acara.

Target dapat berupa AWS layanan lain, aplikasi perangkat lunak sebagai layanan (SaaS), tujuan API, atau titik akhir kustom lainnya. Untuk informasi lebih lanjut, lihat Target di Panduan Pengguna EventBridge .

Untuk petunjuk komprehensif tentang cara membuat aturan bus acara, lihat Membuat aturan yang bereaksi terhadap peristiwa di Panduan EventBridge Pengguna.

Membuat pola acara untuk Transfer Family acara

Saat Transfer Family mengirimkan acara ke bus acara default, EventBridge gunakan pola acara yang ditentukan untuk setiap aturan untuk menentukan apakah acara harus dikirim ke target aturan. Pola peristiwa cocok dengan data dalam Transfer Family peristiwa yang diinginkan. Setiap pola acara adalah objek JSON yang berisi berikut:

- `source`Atribut yang mengidentifikasi layanan yang mengirim acara. Untuk Transfer Family acara, sumbernya adalah`aws.transfer`.
- (Opsional) `detail-type` Atribut yang berisi array dari jenis acara untuk dicocokkan.
- (Opsional) `detail` Atribut yang berisi data peristiwa lain yang cocok.

Misalnya, pola acara berikut cocok dengan semua peristiwa dari Transfer Family:

```
{
  "source": ["aws.transfer"]
}
```

Contoh pola peristiwa berikut cocok dengan semua peristiwa konektor SFTP:

```
{
  "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"]
}
```

Contoh pola acara berikut cocok dengan semua peristiwa gagal Transfer Family:

```
{
```

```
  "source": ["aws.transfer"],
  "detail-type": [{"wildcard", "*Failed"}]
}
```

Contoh pola peristiwa berikut cocok dengan unduhan SFTP yang berhasil untuk nama pengguna pengguna:

```
{
  "source": ["aws.transfer"],
  "detail-type": ["SFTP Server File Download Completed"],
  "detail": {
    "username": [username]
  }
}
```

Untuk informasi selengkapnya tentang penulisan pola acara, lihat Pola acara di Panduan EventBridge Pengguna.

Menguji pola acara untuk Transfer Family acara di EventBridge

Anda dapat menggunakan EventBridge Sandbox untuk mendefinisikan dan menguji pola peristiwa dengan cepat, tanpa harus menyelesaikan proses pembuatan atau pengeditan aturan yang lebih luas. Menggunakan Sandbox, Anda dapat menentukan pola peristiwa dan menggunakan contoh peristiwa untuk mengonfirmasi bahwa pola tersebut cocok dengan peristiwa yang diinginkan. EventBridge memberi Anda pilihan untuk membuat aturan baru dengan menggunakan pola acara itu langsung dari kotak pasir.

Untuk informasi selengkapnya, lihat Menguji pola peristiwa menggunakan EventBridge Kotak Pasir di Panduan EventBridge Pengguna.

Amazon EventBridge izin

Transfer Family tidak memerlukan izin tambahan untuk mengirimkan acara ke Amazon EventBridge.

Target yang Anda tentukan mungkin memerlukan izin atau konfigurasi tertentu. Untuk detail selengkapnya tentang penggunaan layanan khusus untuk target, lihat Amazon EventBridge target di Panduan Amazon EventBridge Pengguna.

EventBridge Sumber daya tambahan

Lihat topik-topik berikut di Panduan Amazon EventBridge Pengguna untuk informasi lebih lanjut tentang cara menggunakan EventBridge untuk memproses dan mengelola acara.

- Untuk informasi rinci tentang cara kerja bus acara, lihat bus Amazon EventBridge acara.
- Untuk informasi tentang struktur acara, lihat Acara.
- Untuk informasi tentang membuat pola peristiwa untuk EventBridge digunakan saat mencocokkan peristiwa dengan aturan, lihat Pola acara.
- Untuk informasi tentang membuat aturan untuk menentukan EventBridge proses peristiwa, lihat Aturan.
- Untuk informasi tentang cara menentukan layanan atau tujuan lain yang EventBridge mengirimkan peristiwa yang cocok, lihat Target.

Transfer Family referensi detail acara

Semua peristiwa dari AWS layanan memiliki seperangkat bidang umum yang berisi metadata tentang acara tersebut. Metadata ini dapat mencakup AWS layanan yang merupakan sumber acara, waktu acara dibuat, akun dan Wilayah tempat acara berlangsung, dan lainnya. Untuk definisi bidang umum ini, lihat Referensi struktur acara di Panduan Amazon EventBridge Pengguna.

Selain itu, setiap acara memiliki `detail` bidang yang berisi data khusus untuk peristiwa tertentu. Referensi berikut mendefinisikan bidang detail untuk berbagai Transfer Family acara.

Saat Anda menggunakan EventBridge untuk memilih dan mengelola Transfer Family acara, pertimbangkan hal berikut:

- `source`Bidang untuk semua acara dari Transfer Family diatur ke`aws.transfer`.
- `detail-type`Bidang menentukan jenis acara.

  Misalnya, `FTP File Server Download Completed`.
- `detail`Bidang berisi data yang spesifik untuk peristiwa tertentu.

Untuk informasi tentang membuat pola peristiwa yang memungkinkan aturan untuk mencocokkan Transfer Family peristiwa, lihat Pola acara di Panduan Amazon EventBridge Pengguna.

Untuk informasi selengkapnya tentang peristiwa dan cara EventBridge memprosesnya, lihat Amazon EventBridge peristiwa di Panduan Amazon EventBridge Pengguna.

Topik

- Acara server SFTP, FTPS, dan FTP
- Acara konektor SFTP
- Acara AS2

Acara server SFTP, FTPS, dan FTP

Berikut ini adalah bidang detail untuk acara server SFTP, FTPS, dan FTP:

- Unduhan Server File FTP Selesai
- Unduhan Server File FTP Gagal
- Unggahan Server File FTP Selesai
- Unggahan Server File FTP Gagal
- Unduhan Server File FTPS Selesai
- Unduhan Server File FTPS Gagal
- Unggahan Server File FTPS Selesai
- Unggahan Server File FTPS Gagal
- Unduhan File Server SFTP Selesai
- Unduhan File Server SFTP Gagal
- Unggahan File Server SFTP Selesai
- Unggahan File Server SFTP Gagal

`detail-type`Bidang `source` dan disertakan di bawah ini karena mengandung nilai khusus untuk Transfer Family acara. Untuk definisi bidang metadata lain yang disertakan dalam semua peristiwa, lihat Referensi struktur acara di Amazon EventBridge Panduan Pengguna.

```
{
  . . .,
  "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`

Mengidentifikasi jenis acara.

Untuk acara ini, nilainya adalah salah satu nama acara server SFTP, FTPS, atau FTP yang tercantum sebelumnya.

`source`

Mengidentifikasi layanan yang menghasilkan peristiwa. Untuk acara Transfer Family, nilai ini adalah`aws.transfer`.

`detail`

Objek JSON yang berisi informasi tentang peristiwa. Layanan yang menghasilkan acara menentukan konten bidang ini.

Untuk acara ini, data meliputi yang berikut:

`failure-code`

Kategori mengapa transfer gagal. Nilai: `PARTIAL_UPLOAD` | `PARTIAL_DOWNLOAD` | `UNKNOWN_ERROR`

`status-code`

Apakah transfer berhasil. Nilai:`COMPLETED` | `FAILED`.

`protocol`

Protokol yang digunakan untuk transfer. Nilai: `SFTP | FTPS | FTP`

`bytes`

Jumlah byte yang ditransfer.

`client-ip`

Alamat IP untuk klien yang terlibat dalam transfer

`failure-message`

Untuk transfer yang gagal, detail mengapa transfer gagal.

`end-timestamp`

Untuk transfer yang berhasil, stempel waktu ketika file selesai diproses.

`etag`

Tag entitas (hanya digunakan untuk file Amazon S3).

`file-path`

Jalur ke file yang ditransfer.

`server-id`

ID unik untuk server Transfer Family.

`username`

Pengguna yang melakukan transfer.

`session-id`

Pengenal unik untuk sesi transfer.

`start-timestamp`

Untuk transfer yang berhasil, stempel waktu saat pemrosesan file dimulai.

Example Unduhan File Server SFTP Gagal contoh peristiwa

Contoh berikut menunjukkan peristiwa di mana unduhan gagal di server SFTP (Amazon EFS adalah penyimpanan yang digunakan).

```
{
    "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 Unggah Server File FTP Contoh acara yang lengkap

Contoh berikut menunjukkan peristiwa di mana unggahan berhasil diselesaikan pada server FTP (Amazon S3 adalah penyimpanan yang digunakan).

```
{
    "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"
    }
}
```

Acara konektor SFTP

Berikut ini adalah bidang detail untuk acara konektor SFTP:

- Kirim File Konektor SFTP Selesai
- Kirim File Konektor SFTP Gagal
- Pengambilan File Konektor SFTP Selesai
- Pengambilan File Konektor SFTP Gagal

`detail-type`Bidang `source` dan disertakan di bawah ini karena mengandung nilai khusus untuk Transfer Family acara. Untuk definisi bidang metadata lain yang disertakan dalam semua peristiwa, lihat Referensi struktur acara di Amazon EventBridge Panduan Pengguna.

```
{
  . . .,
  "detail-type": "string",
  "source": "aws.transfer",
  . . .,
  "detail": {
    "operation" : "string",
    "connector-id" : "string",
    "transfer-id" : "string",
    "file-transfer-id" : "string",
    "url" : "string",
    "file-path" : "string",
    "status-code" : "string",
```

```
    "failure-code" : "string",
    "failure-message" : "string",
    "start-timestamp" : "string",
    "end-timestamp" : "string",
    "local-directory-path" : "string",
    "remote-directory-path" : "string"
    "bytes" : "number",
    "local-file-location" : {
      "domain" : "string",
      "bucket" : "string",
      "key" : "string"
    },
  }
}
```

`detail-type`

Mengidentifikasi jenis acara.

Untuk acara ini, nilainya adalah salah satu nama acara konektor SFTP yang tercantum sebelumnya.

`source`

Mengidentifikasi layanan yang menghasilkan peristiwa. Untuk Transfer Family acara, nilai ini adalah`aws.transfer`.

`detail`

Objek JSON yang berisi informasi tentang peristiwa. Layanan yang menghasilkan acara menentukan konten bidang ini.

Untuk acara ini, data meliputi yang berikut:

`operation`

Apakah `StartFileTransfer` permintaan mengirim atau mengambil file. Nilai:`SEND` | `RETRIEVE`.

`connector-id`

Pengidentifikasi unik untuk konektor SFTP yang digunakan.

`transfer-id`

Pengenal unik untuk acara transfer (`StartFileTransfer`permintaan).

`file-transfer-id`

Pengidentifikasi unik untuk file yang ditransfer.

`url`

URL titik akhir AS2 atau SFTP mitra.

`file-path`

Lokasi dan file yang sedang dikirim atau diambil.

`status-code`

Apakah transfer berhasil. Nilai:`FAILED | COMPLETED`.

`failure-code`

Untuk transfer yang gagal, kode alasan mengapa transfer gagal.

`failure-message`

Untuk transfer yang gagal, detail mengapa transfer gagal.

`start-timestamp`

Untuk transfer yang berhasil, stempel waktu saat pemrosesan file dimulai.

`end-timestamp`

Untuk transfer yang berhasil, stempel waktu saat pemrosesan file selesai.

`local-directory-path`

Untuk `RETRIEVE` permintaan, lokasi di mana untuk menempatkan file yang diambil.

`remote-directory-path`

Untuk `SEND` permintaan, direktori file tempat menempatkan file di server SFTP mitra. Ini adalah nilai untuk `RemoteDirectoryPath` yang diteruskan pengguna ke `StartFileTransfer` permintaan. Anda dapat menentukan direktori default di server SFTP mitra. Jika demikian, bidang ini kosong.

`bytes`

Jumlah byte yang ditransfer. Nilainya adalah 0 untuk transfer yang gagal.

`local-file-location`

Parameter ini berisi rincian lokasi file AWS penyimpanan.

`domain`

Penyimpanan yang digunakan. Saat ini, satu-satunya nilai adalah`S3`.

`bucket`

Wadah untuk objek di Amazon S3.

`key`

Nama yang ditetapkan untuk objek di Amazon S3.

Example SFTP Connector File Kirim contoh peristiwa gagal

Contoh berikut menunjukkan peristiwa di mana konektor SFTP gagal saat mencoba mengirim file ke server SFTP jarak jauh.

```
{
    "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 File Konektor SFTP Mengambil contoh acara yang lengkap

Contoh berikut menunjukkan peristiwa di mana konektor SFTP berhasil mengambil file yang dikirim dari server SFTP jarak jauh.

```
{
    "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"
        }
```

```
    }
}
```

Acara AS2

Berikut ini adalah bidang detail untuk acara AS2:

- Terima Muatan AS2 Selesai
- Penerimaan Muatan AS2 Gagal
- Kirim Muatan AS2 Selesai
- Kirim Muatan AS2 Gagal
- AS2 MDN Terima Selesai
- AS2 MDN Menerima Gagal
- AS2 MDN Kirim Selesai
- AS2 MDN Kirim Gagal

`detail-type`Bidang `source` dan disertakan di bawah ini karena mengandung nilai khusus untuk Transfer Family acara. Untuk definisi bidang metadata lain yang disertakan dalam semua peristiwa, lihat Referensi struktur acara di Amazon EventBridge Panduan Pengguna.

```
{
   . . .,
  "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`

Mengidentifikasi jenis acara.

Untuk acara ini, nilainya adalah salah satu peristiwa AS2 yang tercantum sebelumnya.

`source`

Mengidentifikasi layanan yang menghasilkan peristiwa. Untuk Transfer Family acara, nilai ini adalah`aws.transfer`.

`detail`

Objek JSON yang berisi informasi tentang peristiwa. Layanan yang menghasilkan acara menentukan konten bidang ini.

`s3-attributes`

Mengidentifikasi bucket dan kunci Amazon S3 untuk file yang ditransfer. Untuk acara MDN, ini juga mengidentifikasi bucket dan kunci untuk file MDN.

`file-bucket`

Wadah untuk objek di Amazon S3.

`file-key`

Nama yang ditetapkan untuk objek di Amazon S3.

`json-bucket`

Untuk transfer SELESAI atau GAGAL, wadah untuk file JSON.

`json-key`

Untuk transfer SELESAI atau GAGAL, nama yang ditetapkan ke file JSON di Amazon S3.

`mdn-bucket`

Untuk acara MDN, wadah untuk file MDN.

`mdn-key`

Untuk peristiwa MDN, nama ditetapkan ke file MDN di Amazon S3.

`mdn-subject`

Untuk acara MDN, deskripsi teks untuk disposisi pesan.

`mdn-message-id`

Untuk acara MDN, ID unik untuk pesan MDN.

`disposition`

Untuk acara MDN, kategori untuk disposisi.

`bytes`

Jumlah byte dalam pesan.

`as2-from`

Mitra dagang AS2 yang mengirim pesan.

`as2-message-id`

Pengenal unik untuk pesan AS2 yang ditransfer.

`as2-to`

Mitra dagang AS2 yang menerima pesan.

`connector-id`

Untuk pesan AS2 yang dikirim dari server Transfer Family ke mitra dagang, pengenal unik untuk konektor AS2 digunakan.

`client-ip`

Untuk kejadian server (transfer dari mitra dagang ke server Transfer Family), alamat IP untuk klien yang terlibat dalam transfer.

`agreement-id`

Untuk peristiwa server, pengenal unik untuk perjanjian AS2.

`server-id`

Untuk kejadian server, ID unik hanya untuk server Transfer Family.

`requester-file-name`

Untuk acara payload, nama asli untuk file yang diterima selama transfer.

`message-subject`

Deskripsi teks untuk subjek pesan.

`start-timestamp`

Untuk transfer yang berhasil, stempel waktu saat pemrosesan file dimulai.

`end-timestamp`

Untuk transfer yang berhasil, stempel waktu saat pemrosesan file selesai.

`status-code`

Kode yang sesuai dengan keadaan proses transfer pesan AS2. Nilai yang valid:`COMPLETED | FAILED | PROCESSING`.

`failure-code`

Untuk transfer yang gagal, kategori mengapa transfer gagal.

`failure-message`

Untuk transfer yang gagal, detail mengapa transfer gagal.

`transfer-id`

Pengenal unik untuk acara transfer.

Example AS2 Payload Receive Contoh acara Selesai

```
{
```

```
    "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 Menerima contoh peristiwa Gagal

```
{
  "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"
  }
}
```

Keamanan di AWS Transfer Family

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. Model tanggung jawab bersama menjelaskan hal ini sebagai keamanan dari cloud dan keamanan di cloud:

Untuk mempelajari apakah an Layanan AWS berada dalam lingkup program kepatuhan tertentu, lihat Layanan AWS di Lingkup oleh Program Kepatuhan Layanan AWS dan pilih program kepatuhan yang Anda minati. Untuk informasi umum, lihat Program AWS Kepatuhan Program AWS .

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat Mengunduh Laporan di AWS Artifact .

Tanggung jawab kepatuhan Anda saat menggunakan Layanan AWS ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- Panduan Memulai Cepat Keamanan dan Kepatuhan — Panduan penerapan ini membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar AWS yang berfokus pada keamanan dan kepatuhan.
- Arsitektur untuk Keamanan dan Kepatuhan HIPAA di Amazon Web Services — Whitepaper ini menjelaskan bagaimana perusahaan dapat menggunakan AWS untuk membuat aplikasi yang memenuhi syarat HIPAA.

  > **Note**
  >
  > Tidak semua memenuhi Layanan AWS syarat HIPAA. Untuk informasi selengkapnya, lihat Referensi Layanan yang Memenuhi Syarat HIPAA.

- AWS Sumber Daya AWS — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- AWS Panduan Kepatuhan Pelanggan - Memahami model tanggung jawab bersama melalui lensa kepatuhan. Panduan ini merangkum praktik terbaik untuk mengamankan Layanan AWS dan memetakan panduan untuk kontrol keamanan di berbagai kerangka kerja (termasuk Institut

Standar dan Teknologi Nasional (NIST), Dewan Standar Keamanan Industri Kartu Pembayaran (PCI), dan Organisasi Internasional untuk Standardisasi (ISO)).

- Mengevaluasi Sumber Daya dengan Aturan dalam Panduan AWS Config Pengembang — AWS Config Layanan menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- AWS Security Hub— Ini Layanan AWS memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS. Security Hub menggunakan kontrol keamanan untuk mengevaluasi sumber daya AWS Anda dan memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik. Untuk daftar layanan dan kontrol yang didukung, lihat Referensi kontrol Security Hub.
- AWS Audit ManagerIni Layanan AWS membantu Anda terus mengaudit AWS penggunaan Anda untuk menyederhanakan cara Anda mengelola risiko dan kepatuhan terhadap peraturan dan standar industri.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan AWS Transfer Family. Topik berikut menunjukkan cara mengonfigurasi AWS Transfer Family untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga belajar cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan AWS Transfer Family sumber daya Anda.

Kami menawarkan lokakarya yang menyediakan panduan preskriptif dan lab langsung tentang bagaimana Anda dapat membangun arsitektur transfer file yang terukur dan aman AWS tanpa perlu memodifikasi aplikasi yang ada atau mengelola infrastruktur server. Anda dapat melihat detail untuk lokakarya ini di sini.

Topik

- Kebijakan keamanan untuk AWS Transfer Family server
- Kebijakan keamanan untuk konektor AWS Transfer Family SFTP
- Menggunakan pertukaran kunci pasca-kuantum hibrida dengan AWS Transfer Family
- Perlindungan data di AWS Transfer Family
- Identitas dan manajemen akses untuk AWS Transfer Family
- Validasi kepatuhan untuk AWS Transfer Family
- Ketahanan di AWS Transfer Family
- Keamanan infrastruktur di AWS Transfer Family
- Tambahkan firewall aplikasi web

- Pencegahan confused deputy lintas layanan
- AWS kebijakan terkelola untuk AWS Transfer Family

Kebijakan keamanan untuk AWS Transfer Family server

Kebijakan keamanan server AWS Transfer Family memungkinkan Anda untuk membatasi set algoritma kriptografi (kode otentikasi pesan (MAC), pertukaran kunci (KEX), dan cipher suite) yang terkait dengan server Anda. Untuk daftar algoritma kriptografi yang didukung, lihat. Algoritma kriptografi Untuk daftar algoritme kunci yang didukung untuk digunakan dengan kunci host server dan kunci pengguna yang dikelola layanan, lihat. Algoritma yang didukung untuk kunci pengguna dan server

> **Note**
>
> Kami sangat menyarankan untuk memperbarui server Anda ke kebijakan keamanan terbaru kami. Kebijakan keamanan terbaru kami adalah default. Setiap pelanggan yang membuat server Transfer Family menggunakan CloudFormation dan menerima kebijakan keamanan default akan secara otomatis ditetapkan kebijakan terbaru. Jika Anda khawatir tentang kompatibilitas klien, harap sebutkan kebijakan keamanan mana yang ingin Anda gunakan saat membuat atau memperbarui server daripada menggunakan kebijakan default, yang dapat berubah sewaktu-waktu.
> Untuk mengubah kebijakan keamanan server, lihatEdit kebijakan keamanan.

Untuk informasi selengkapnya tentang keamanan di Transfer Family, lihat postingan blog, Bagaimana Transfer Family dapat membantu Anda membangun solusi transfer file terkelola yang aman dan sesuai.

Topik

- Algoritma kriptografi
- TransferSecurityPolicy-2024-01
- TransferSecurityPolicy-2023-05
- TransferSecurityPolicy-2022-03
- TransferSecurityPolicy-2020-06
- TransferSecurityPolicy-2018-11

- TransferSecurityPolicy-FIP-2024-01
- TransferSecurityPolicy-FIP-2023-05
- TransferSecurityPolicy-FIP-2020-06
- Pasca kebijakan keamanan Quantum

> **Note**
>
> `TransferSecurityPolicy-2024-01`adalah kebijakan keamanan default yang dilampirkan ke server Anda saat membuat server menggunakan konsol, API, atau CLI.

Algoritma kriptografi

Untuk kunci host, kami mendukung algoritma berikut:

- `rsa-sha2-256`
- `rsa-sha2-512`
- `ecdsa-sha2-nistp256`
- `ecdsa-sha2-nistp384`
- `ecdsa-sha2-nistp521`
- `ssh-ed25519`

Selain itu, kebijakan keamanan 2018 dan 2020 memungkinkan`ssh-rsa`.

> **Note**
>
> Penting untuk memahami perbedaan antara tipe kunci RSA — yang selalu `ssh-rsa` — dan algoritma kunci host RSA, yang dapat berupa salah satu algoritma yang didukung.

Berikut ini adalah daftar algoritma kriptografi yang didukung untuk setiap kebijakan keamanan.

> **Note**
>
> Dalam tabel dan kebijakan berikut, perhatikan penggunaan jenis algoritma berikut.

- Server SFTP hanya menggunakan algoritma di SshCiphers, SshKexs, dan bagian. SshMacs
- Server FTPS hanya menggunakan algoritma di bagian ini. TlsCiphers
- Server FTP, karena mereka tidak menggunakan enkripsi, tidak menggunakan algoritme ini.

Kebijakan keamanan	2024-01	2023-05	2022-03	2020-06	FIP-2024-01	FIP-2023-05	FIP-2020-06	2018-11
SshCiphers								
aes128-ctr	♦			♦	♦		♦	♦
aes128-gcm@openssh.com	♦	♦	♦	♦	♦	♦	♦	♦
aes192-ctr	♦	♦	♦	♦	♦	♦	♦	♦
aes256-ctr	♦	♦	♦	♦	♦	♦	♦	♦
aes256-gcm@openssh.com	♦	♦	♦	♦	♦	♦	♦	♦
chacha20-poly1305@openssh.com				♦				♦
SshKexs								

Kebijakan keamanan	2024-01	2023-05	2022-03	2020-06	FIP-2024-01	FIP-2023-05	FIP-2020-06	2018-11
kurva25519-sha256	♦	♦	♦					♦
curve25519-sha256@libssh.org	♦	♦	♦					♦
diffie-hellman-group14-sha1								♦
diffie-hellman-group14-sha256				♦			♦	♦
diffie-hellman-group16-sha512	♦	♦	♦	♦	♦	♦	♦	♦
diffie-hellman-group18-sha512	♦	♦	♦	♦	♦	♦	♦	♦

Kebijakan keamanan	2024-01	2023-05	2022-03	2020-06	FIP-2024-01	FIP-2023-05	FIP-2020-06	2018-11
diffie-hellman-group-exchange-sha256		♦	♦	♦		♦	♦	♦
ecdh-nist p256-kyber-512r3-sha256-d00 @openquantumsafe.org	♦				♦			
ecdh-nist p384-kyber-768r3-sha384-d00 @openquantumsafe.org	♦				♦			

Kebijakan keamanan	2024-01	2023-05	2022-03	2020-06	FIP-2024-01	FIP-2023-05	FIP-2020-06	2018-11
ecdh-nistp521-kyber-1024r3-sha512-d00@openquantumsafe.org	♦				♦			
ecdh-sha2-nistp256	♦			♦	♦		♦	♦
ecdh-sha2-nistp384	♦			♦	♦		♦	♦
ecdh-sha2-nistp521	♦			♦	♦		♦	♦
x25519-kyber-512r3-sha256-d00@amazon.com	♦							

SshMacs

Kebijakan keamanan	2024-01	2023-05	2022-03	2020-06	FIP-2024-01	FIP-2023-05	FIP-2020-06	2018-11
hmac-sha1								♦
hmac-sha1-etm@openssh.com								♦
hmac-sha2-256			♦	♦			♦	♦
hmac-sha2-256-etm@openssh.com	♦	♦	♦	♦	♦	♦	♦	♦
hmac-sha2-512			♦	♦			♦	♦
hmac-sha2-512-etm@openssh.com	♦	♦	♦	♦	♦	♦	♦	♦
umac-128-etm@openssh.com				♦				♦

Kebijakan keamanan	2024-01	2023-05	2022-03	2020-06	FIP-2024-01	FIP-2023-05	FIP-2020-06	2018-11
umac-128@openssh.com				♦				♦
umac-64-etm@openssh.com								♦
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	♦	♦	♦	♦	♦	♦	♦	♦

Kebijakan keamanan	2024-01	2023-05	2022-03	2020-06	FIP-2024-01	FIP-2023-05	FIP-2020-06	2018-11
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	♦	♦	♦	♦	♦	♦	♦	
TLS_RSA_WITH_AES_128_CBC_SHA256								♦

Kebijakan keamanan	2024-01	2023-05	2022-03	2020-06	FIP-2024-01	FIP-2023-05	FIP-2020-06	2018-11
TLS_RSA_WITH_AES_256_CBC_SHA256								♦

TransferSecurityPolicy-2024-01

Berikut ini menunjukkan kebijakan keamanan TransferSecurityPolicy -2024-01.

```
{
    "SecurityPolicy": {
        "Fips": false,
        "SecurityPolicyName": "TransferSecurityPolicy-2024-01",
        "SshCiphers": [
            "aes128-gcm@openssh.com",
            "aes256-gcm@openssh.com",
            "aes128-ctr",
            "aes256-ctr",
            "aes192-ctr"
        ],
        "SshKexs": [
            "ecdh-nistp384-kyber-768r3-sha384-d00@openquantumsafe.org",
            "x25519-kyber-512r3-sha256-d00@amazon.com",
            "ecdh-nistp256-kyber-512r3-sha256-d00@openquantumsafe.org",
            "ecdh-nistp521-kyber-1024r3-sha512-d00@openquantumsafe.org",
            "ecdh-sha2-nistp256",
            "ecdh-sha2-nistp384",
            "ecdh-sha2-nistp521",
            "curve25519-sha256",
            "curve25519-sha256@libssh.org",
            "diffie-hellman-group18-sha512",
            "diffie-hellman-group16-sha512",
            "diffie-hellman-group-exchange-sha256"
        ],
        "SshMacs": [
            "hmac-sha2-256-etm@openssh.com",
            "hmac-sha2-512-etm@openssh.com"
        ],
```

```
        "TlsCiphers": [
            "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384"
        ]
    }
}
```

TransferSecurityPolicy-2023-05

Berikut ini menunjukkan kebijakan keamanan TransferSecurityPolicy -2023-05.

```
{
    "SecurityPolicy": {
        "Fips": false,
        "SecurityPolicyName": "TransferSecurityPolicy-2023-05",
        "SshCiphers": [
            "aes256-gcm@openssh.com",
            "aes128-gcm@openssh.com",
            "aes256-ctr",
            "aes192-ctr"
        ],
        "SshKexs": [
            "curve25519-sha256",
            "curve25519-sha256@libssh.org",
            "diffie-hellman-group16-sha512",
            "diffie-hellman-group18-sha512",
            "diffie-hellman-group-exchange-sha256"
        ],
        "SshMacs": [
            "hmac-sha2-512-etm@openssh.com",
            "hmac-sha2-256-etm@openssh.com"
        ],
        "TlsCiphers": [
            "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
```

```
            "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384"
        ]
    }
}
```

TransferSecurityPolicy-2022-03

Berikut ini menunjukkan kebijakan keamanan TransferSecurityPolicy -2022-03.

```
{
  "SecurityPolicy": {
    "Fips": false,
    "SecurityPolicyName": "TransferSecurityPolicy-2022-03",
    "SshCiphers": [
      "aes256-gcm@openssh.com",
      "aes128-gcm@openssh.com",
      "aes256-ctr",
      "aes192-ctr"
    ],
    "SshKexs": [
      "curve25519-sha256",
      "curve25519-sha256@libssh.org",
      "diffie-hellman-group16-sha512",
      "diffie-hellman-group18-sha512",
      "diffie-hellman-group-exchange-sha256"
    ],
    "SshMacs": [
      "hmac-sha2-512-etm@openssh.com",
      "hmac-sha2-256-etm@openssh.com",
      "hmac-sha2-512",
      "hmac-sha2-256"
    ],
    "TlsCiphers": [
      "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
      "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
      "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256",
      "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
      "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
      "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
      "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384",
```

```
      "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384"
    ]
  }
}
```

TransferSecurityPolicy-2020-06

Berikut ini menunjukkan kebijakan keamanan TransferSecurityPolicy -2020-06.

```
{
  "SecurityPolicy": {
    "Fips": false,
    "SecurityPolicyName": "TransferSecurityPolicy-2020-06",
    "SshCiphers": [
      "chacha20-poly1305@openssh.com",
      "aes128-ctr",
      "aes192-ctr",
      "aes256-ctr",
      "aes128-gcm@openssh.com",
      "aes256-gcm@openssh.com"
    ],
    "SshKexs": [
      "ecdh-sha2-nistp256",
      "ecdh-sha2-nistp384",
      "ecdh-sha2-nistp521",
      "diffie-hellman-group-exchange-sha256",
      "diffie-hellman-group16-sha512",
      "diffie-hellman-group18-sha512",
      "diffie-hellman-group14-sha256"
    ],
    "SshMacs": [
      "umac-128-etm@openssh.com",
      "hmac-sha2-256-etm@openssh.com",
      "hmac-sha2-512-etm@openssh.com",
      "umac-128@openssh.com",
      "hmac-sha2-256",
      "hmac-sha2-512"
    ],
    "TlsCiphers": [
      "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
      "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
      "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256",
      "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
```

```
      "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
      "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
      "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384",
      "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384"
    ]
  }
}
```

TransferSecurityPolicy-2018-11

Berikut ini menunjukkan kebijakan keamanan TransferSecurityPolicy -2018-11.

```
{
  "SecurityPolicy": {
    "Fips": false,
    "SecurityPolicyName": "TransferSecurityPolicy-2018-11",
    "SshCiphers": [
      "chacha20-poly1305@openssh.com",
      "aes128-ctr",
      "aes192-ctr",
      "aes256-ctr",
      "aes128-gcm@openssh.com",
      "aes256-gcm@openssh.com"
    ],
    "SshKexs": [
      "curve25519-sha256",
      "curve25519-sha256@libssh.org",
      "ecdh-sha2-nistp256",
      "ecdh-sha2-nistp384",
      "ecdh-sha2-nistp521",
      "diffie-hellman-group-exchange-sha256",
      "diffie-hellman-group16-sha512",
      "diffie-hellman-group18-sha512",
      "diffie-hellman-group14-sha256",
      "diffie-hellman-group14-sha1"
    ],
    "SshMacs": [
      "umac-64-etm@openssh.com",
      "umac-128-etm@openssh.com",
      "hmac-sha2-256-etm@openssh.com",
      "hmac-sha2-512-etm@openssh.com",
      "hmac-sha1-etm@openssh.com",
      "umac-64@openssh.com",
```

```
      "umac-128@openssh.com",
      "hmac-sha2-256",
      "hmac-sha2-512",
      "hmac-sha1"
    ],
    "TlsCiphers": [
      "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
      "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
      "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256",
      "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
      "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
      "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
      "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384",
      "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384",
      "TLS_RSA_WITH_AES_128_CBC_SHA256",
      "TLS_RSA_WITH_AES_256_CBC_SHA256"
    ]
  }
}
```

TransferSecurityPolicy-FIP-2024-01

Berikut ini menunjukkan kebijakan keamanan TransferSecurityPolicy -FIPS-2024-01.

> **Note**
>
> Titik akhir layanan FIPS dan kebijakan keamanan TransferSecurityPolicy -FIPS-2024-01 hanya tersedia di beberapa Wilayah. AWS Untuk informasi selengkapnya, lihat AWS Transfer Family titik akhir dan kuota di. Referensi Umum AWS

```
{
    "SecurityPolicy": {
        "Fips": true,
        "SecurityPolicyName": "TransferSecurityPolicy-FIPS-2024-01",
        "SshCiphers": [
            "aes128-gcm@openssh.com",
            "aes256-gcm@openssh.com",
            "aes128-ctr",
            "aes256-ctr",
            "aes192-ctr"
        ],
```

```
        "SshKexs": [
            "ecdh-nistp384-kyber-768r3-sha384-d00@openquantumsafe.org",
            "ecdh-nistp256-kyber-512r3-sha256-d00@openquantumsafe.org",
            "ecdh-nistp521-kyber-1024r3-sha512-d00@openquantumsafe.org",
            "ecdh-sha2-nistp256",
            "ecdh-sha2-nistp384",
            "ecdh-sha2-nistp521",
            "diffie-hellman-group18-sha512",
            "diffie-hellman-group16-sha512",
            "diffie-hellman-group-exchange-sha256"
        ],
        "SshMacs": [
            "hmac-sha2-256-etm@openssh.com",
            "hmac-sha2-512-etm@openssh.com"
        ],
        "TlsCiphers": [
            "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384"
        ]
    }
}
```

TransferSecurityPolicy-FIP-2023-05

Detail sertifikasi FIPS untuk AWS Transfer Family dapat ditemukan di https://csrc.nist.gov/projects/cryptographic-module-validation-program/validated-modules/search/all

Berikut ini menunjukkan kebijakan keamanan TransferSecurityPolicy -FIPS-2023-05.

> **Note**
>
> Titik akhir layanan FIPS dan kebijakan keamanan TransferSecurityPolicy -FIPS-2023-05 hanya tersedia di beberapa Wilayah. AWS Untuk informasi selengkapnya, lihat AWS Transfer Family titik akhir dan kuota di. Referensi Umum AWS

```
{
    "SecurityPolicy": {
        "Fips": true,
        "SecurityPolicyName": "TransferSecurityPolicy-FIPS-2023-05",
        "SshCiphers": [
            "aes256-gcm@openssh.com",
            "aes128-gcm@openssh.com",
            "aes256-ctr",
            "aes192-ctr"
        ],
        "SshKexs": [
            "diffie-hellman-group16-sha512",
            "diffie-hellman-group18-sha512",
            "diffie-hellman-group-exchange-sha256"
        ],
        "SshMacs": [
            "hmac-sha2-256-etm@openssh.com",
            "hmac-sha2-512-etm@openssh.com"
        ],
        "TlsCiphers": [
            "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384"
        ]
    }
}
```

TransferSecurityPolicy-FIP-2020-06

Detail sertifikasi FIPS untuk AWS Transfer Family dapat ditemukan di https://csrc.nist.gov/projects/cryptographic-module-validation-program/validated-modules/search/all

Berikut ini menunjukkan kebijakan keamanan TransferSecurityPolicy -FIPS-2020-06.

> **Note**
>
> Titik akhir layanan FIPS dan kebijakan keamanan TransferSecurityPolicy -FIPS-2020-06 hanya tersedia di beberapa Wilayah. AWS Untuk informasi selengkapnya, lihat AWS Transfer Family titik akhir dan kuota di. Referensi Umum AWS

```
{
  "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"
```

```
    ]
  }
}
```

Pasca kebijakan keamanan Quantum

Tabel ini mencantumkan algoritme untuk kebijakan keamanan kuantum pasca Transfer Family. Kebijakan ini dijelaskan secara rinci dalamMenggunakan pertukaran kunci pasca-kuantum hibrida dengan AWS Transfer Family.

Daftar kebijakan mengikuti tabel.

Kebijakan keamanan	TransferSecurityPolicy-PQ-SSH-Percobaan-2023-04	TransferSecurityPolicy-PQ-SSH-FIP-Percobaan-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	♦	♦
ecdh-nistp384-kyber-768r3-sha384-d00@openquantumsafe.org	♦	♦
ecdh-nistp521-kyber-1024r3-sha512-d00@openquantumsafe.org	♦	♦

Kebijakan keamanan	TransferSecurityPolicy-PQ-SSH-Percobaan-2023-04	TransferSecurityPolicy-PQ-SSH-FIP-Percobaan-2023-04
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	♦	
kurva25519-sha256	♦	
MACs		
hmac-sha2-256-etm@openssh.com	♦	♦
hmac-sha2-256	♦	♦
hmac-sha2-512-etm@openssh.com	♦	♦
hmac-sha2-512	♦	♦
TLS ciphers		

Kebijakan keamanan	TransferSecurityPolicy-PQ-SSH-Percobaan-2023-04	TransferSecurityPolicy-PQ-SSH-FIP-Percobaan-2023-04
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	♦	♦
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	♦	♦
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384	♦	♦
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384	♦	♦
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	♦	♦
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	♦	♦
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384	♦	♦
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384	♦	♦

TransferSecurityPolicy-PQ-SSH-Percobaan-2023-04

Berikut ini menunjukkan kebijakan keamanan TransferSecurityPolicy -PQ-SSH-Experimental-2023-04.

```
{
    "SecurityPolicy": {
        "Fips": false,
        "SecurityPolicyName": "TransferSecurityPolicy-PQ-SSH-Experimental-2023-04",
        "SshCiphers": [
            "aes256-gcm@openssh.com",
            "aes128-gcm@openssh.com",
            "aes256-ctr",
```

```
            "aes192-ctr"
        ],
        "SshKexs": [
            "ecdh-nistp384-kyber-768r3-sha384-d00@openquantumsafe.org",
            "x25519-kyber-512r3-sha256-d00@amazon.com",
            "ecdh-nistp256-kyber-512r3-sha256-d00@openquantumsafe.org",
            "ecdh-nistp521-kyber-1024r3-sha512-d00@openquantumsafe.org",
            "curve25519-sha256",
            "curve25519-sha256@libssh.org",
            "diffie-hellman-group16-sha512",
            "diffie-hellman-group18-sha512",
            "diffie-hellman-group-exchange-sha256"
        ],
        "SshMacs": [
            "hmac-sha2-512-etm@openssh.com",
            "hmac-sha2-256-etm@openssh.com",
            "hmac-sha2-512",
            "hmac-sha2-256"
        ],
        "TlsCiphers": [
            "TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256",
            "TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384",
            "TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384",
            "TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384"
        ]
    }
}
```

TransferSecurityPolicy-PQ-SSH-FIP-Percobaan-2023-04

Berikut ini menunjukkan kebijakan keamanan TransferSecurityPolicy -PQ-SSH-FIPS-eksperimental-2023-04.

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

Kebijakan keamanan untuk konektor AWS Transfer Family SFTP

Kebijakan keamanan konektor SFTP AWS Transfer Family memungkinkan Anda membatasi rangkaian algoritme kriptografi (kode otentikasi pesan (MAC), pertukaran kunci (KEX), dan rangkaian

sandi) yang terkait dengan konektor SFTP Anda. Berikut ini adalah daftar algoritma kriptografi yang didukung untuk setiap kebijakan keamanan konektor SFTP.

> **Note**
>
> `TransferSFTPConnectorSecurityPolicy-2024-03`adalah kebijakan keamanan default yang diterapkan pada konektor SFTP.

Kebijakan keamanan	TransfersFTP Connector SecurityPolicy -2024-03	TransfersFTP Connector SecurityPolicy -2023-07
Ciphers		
aes128-ctr		♦
aes128-gcm@openssh.com	♦	♦
aes192-ctr	♦	♦
aes256-ctr	♦	♦
aes256-gcm@openssh.com	♦	♦
Kexs		
kurva25519-sha256	♦	♦
curve25519-sha256@libssh.org	♦	♦
diffie-hellman-group14-sha1		♦
diffie-hellman-group16-sha512	♦	♦
diffie-hellman-group18-sha512	♦	♦
diffie-hellman-group-exchange-sha256	♦	♦
Macs		

Kebijakan keamanan	TransfersFTP Connector SecurityPolicy -2024-03	TransfersFTP Connector SecurityPolicy -2023-07
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		♦

Menggunakan pertukaran kunci pasca-kuantum hibrida dengan AWS Transfer Family

AWS Transfer Family mendukung opsi pembentukan kunci pasca-kuantum hibrida untuk protokol Secure Shell (SSH). Pembentukan kunci pasca-kuantum diperlukan karena sudah memungkinkan untuk merekam lalu lintas jaringan dan menyimpannya untuk dekripsi di masa depan oleh komputer kuantum, yang disebut serangan. store-now-harvest-later

Anda dapat menggunakan opsi ini saat terhubung ke Transfer Family untuk transfer file aman ke dalam dan keluar dari penyimpanan Amazon Simple Storage Service (Amazon S3) atau Amazon Elastic File System (Amazon EFS). Pembentukan kunci hibrida pasca-kuantum di SSH memperkenalkan mekanisme pembentukan kunci pasca-kuantum, yang digunakannya bersama dengan algoritma pertukaran kunci klasik. Kunci SSH yang dibuat dengan cipher suite klasik aman dari serangan brute-force dengan teknologi saat ini. Namun, enkripsi klasik diperkirakan tidak akan tetap aman setelah munculnya komputasi kuantum skala besar di masa depan.

Jika organisasi Anda bergantung pada kerahasiaan data jangka panjang yang diteruskan melalui koneksi Transfer Family, Anda harus mempertimbangkan rencana untuk bermigrasi ke kriptografi pasca-kuantum sebelum komputer kuantum skala besar tersedia untuk digunakan.

Untuk melindungi data yang dienkripsi hari ini terhadap potensi serangan future, AWS berpartisipasi dengan komunitas kriptografi dalam pengembangan algoritma tahan kuantum atau pasca-kuantum. Kami telah menerapkan rangkaian cipher pertukaran kunci pasca-kuantum hibrida di Transfer Family yang menggabungkan elemen klasik dan pasca-kuantum.

Suite sandi hibrida ini tersedia untuk digunakan pada beban kerja produksi Anda di sebagian besar Wilayah. AWS Namun, karena karakteristik kinerja dan persyaratan bandwidth suite cipher hybrid berbeda dengan mekanisme pertukaran kunci klasik, kami sarankan Anda mengujinya pada koneksi Transfer Family Anda.

Cari tahu lebih lanjut tentang kriptografi pasca-kuantum di posting blog keamanan Post-Quantum Cryptography.

Daftar Isi

- Tentang pertukaran kunci hibrida pasca-kuantum di SSH
- Cara kerja pembentukan kunci hibrida pasca-kuantum di Transfer Family
  - Mengapa Kyber?
  - Pertukaran kunci SSH hibrida pasca-kuantum dan persyaratan kriptografi (FIPS 140)
- Menguji pertukaran kunci hibrida pasca-kuantum di Transfer Family
  - Aktifkan pertukaran kunci hybrid pasca-kuantum di titik akhir SFTP Anda
  - Siapkan klien SFTP yang mendukung pertukaran kunci hybrid pasca-kuantum
  - Konfirmasikan pertukaran kunci hibrida pasca-kuantum di SFTP

Tentang pertukaran kunci hibrida pasca-kuantum di SSH

Transfer Family mendukung suite cipher pertukaran kunci hibrida pasca-kuantum, yang menggunakan algoritme pertukaran kunci Elliptic Curve Diffie-Hellman (ECDH) klasik, dan CRYSTALS Kyber. Kyber adalah enkripsi kunci publik pasca-kuantum dan algoritma pembentukan kunci yang telah ditetapkan oleh National Institute for Standards and Technology (NIST) sebagai algoritma perjanjian kunci pasca-kuantum standar pertama.

Klien dan server masih melakukan pertukaran kunci ECDH. Selain itu, server merangkum rahasia bersama pasca-kuantum ke kunci publik KEM pasca-kuantum klien, yang diiklankan dalam pesan pertukaran kunci SSH klien. Strategi ini menggabungkan jaminan tinggi pertukaran kunci klasik dengan keamanan pertukaran kunci pasca-kuantum yang diusulkan, untuk membantu memastikan bahwa jabat tangan dilindungi selama ECDH atau rahasia bersama pasca-kuantum tidak dapat dipatahkan.

Cara kerja pembentukan kunci hibrida pasca-kuantum di Transfer Family

AWS baru-baru ini mengumumkan dukungan untuk pertukaran kunci pasca-kuantum dalam transfer file SFTP di. AWS Transfer Family Transfer Family dengan aman menskalakan transfer business-to-business file ke layanan AWS Storage menggunakan SFTP dan protokol lainnya. SFTP adalah versi yang lebih aman dari File Transfer Protocol (FTP) yang berjalan melalui SSH. Dukungan pertukaran kunci pasca-kuantum dari Transfer Family meningkatkan bilah keamanan untuk transfer data melalui SFTP.

Dukungan SFTP pertukaran kunci hibrida pasca-kuantum di Transfer Family termasuk menggabungkan algoritma pasca-kuantum Kyber-512, Kyber-768, dan Kyber-1024, dengan ECDH lebih dari kurva P256, P384, P521, atau Curve25519. Metode pertukaran kunci SSH yang sesuai berikut ini ditentukan dalam draf pertukaran kunci SSH hibrida pasca-kuantum.

- `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**
>
> Metode pertukaran kunci baru ini dapat berubah saat draf berkembang menuju standardisasi, atau ketika NIST meratifikasi algoritma Kyber.

Mengapa Kyber?

AWS berkomitmen untuk mendukung standar, algoritma interoperable. Kyber adalah algoritma enkripsi pasca-kuantum pertama yang dipilih untuk standardisasi oleh proyek NIST Post-Quantum Cryptography. Beberapa badan standar sudah mengintegrasikan Kyber ke dalam protokol. AWS sudah mendukung Kyber di TLS di beberapa titik akhir AWS API.

Sebagai bagian dari komitmen ini, AWS telah mengajukan draf proposal ke IETF untuk kriptografi pasca-kuantum yang menggabungkan Kyber dengan kurva yang disetujui NIST seperti P256 untuk SSH. Untuk membantu meningkatkan keamanan bagi pelanggan kami, AWS implementasi pertukaran kunci pasca-kuantum di SFTP dan SSH mengikuti rancangan itu. Kami berencana untuk mendukung pembaruan masa depan sampai proposal kami diadopsi oleh IETF dan menjadi standar.

Metode pertukaran kunci baru (tercantum di bagianCara kerja pembentukan kunci hibrida pasca-kuantum di Transfer Family) mungkin berubah saat draf berkembang menuju standardisasi atau ketika NIST meratifikasi algoritma Kyber.

> **Note**
>
> Dukungan algoritme pasca-kuantum saat ini tersedia untuk pertukaran kunci hibrida pasca-kuantum di TLS untuk AWS KMS (lihat Menggunakan TLS pasca-kuantum hibrida dengan AWS KMS),,AWS Certificate Manager dan titik akhir API. AWS Secrets Manager

Pertukaran kunci SSH hibrida pasca-kuantum dan persyaratan kriptografi (FIPS 140)

Untuk pelanggan yang memerlukan kepatuhan FIPS, Transfer Family menyediakan kriptografi yang disetujui FIPS di SSH dengan menggunakan perpustakaan kriptografi open-source bersertifikat AWS FIPS 140, -LC. AWSMetode pertukaran kunci hibrida pasca-kuantum yang didukung dalam TransferSecurityPolicy -PQ-SSH-FIPS-Experimental-2023-04 di Transfer Family disetujui FIPS menurut SP 800-56Cr2 NIST (bagian 2). Kantor Federal Jerman untuk Keamanan Informasi (BSI) dan Agence nationale de la sécurité des systèmes d'information (ANSSI) Prancis juga merekomendasikan metode pertukaran kunci hibrida pasca-kuantum tersebut.

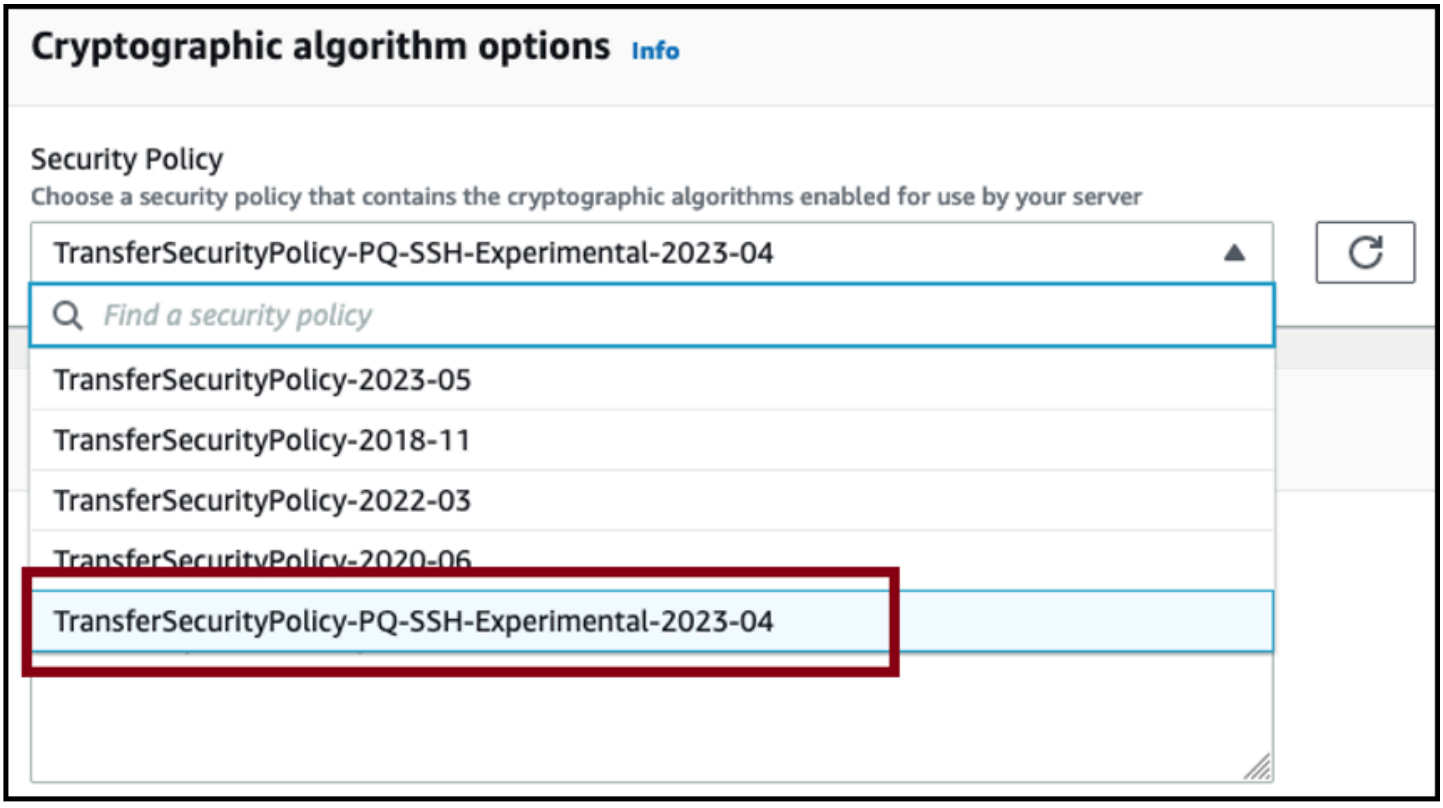
Menguji pertukaran kunci hibrida pasca-kuantum di Transfer Family

Bagian ini menjelaskan langkah-langkah yang Anda ambil untuk menguji pertukaran kunci hibrida pasca-kuantum.

1. Aktifkan pertukaran kunci hybrid pasca-kuantum di titik akhir SFTP Anda.
2. Gunakan klien SFTP (seperti Siapkan klien SFTP yang mendukung pertukaran kunci hybrid pasca-kuantum) yang mendukung pertukaran kunci hibrida pasca-kuantum dengan mengikuti panduan dalam spesifikasi draf yang disebutkan di atas.
3. Transfer file menggunakan server Transfer Family.
4. Konfirmasikan pertukaran kunci hibrida pasca-kuantum di SFTP.

Aktifkan pertukaran kunci hybrid pasca-kuantum di titik akhir SFTP Anda

Anda dapat memilih kebijakan SSH saat membuat endpoint server SFTP baru di Transfer Family, atau dengan mengedit opsi algoritma Cryptographic di endpoint SFTP yang ada. Snapshot berikut menunjukkan contoh AWS Management Console tempat Anda memperbarui kebijakan SSH.



Nama kebijakan SSH yang mendukung pertukaran kunci pasca-kuantum adalah -PQ-SSH-eksperimental-2023-04 dan TransferSecurityPolicy-PQ-SSH-FIPS-Experimental-2023-04.

TransferSecurityPolicy Untuk detail selengkapnya tentang kebijakan Transfer Family, lihatKebijakan keamanan untuk AWS Transfer Family server.

Siapkan klien SFTP yang mendukung pertukaran kunci hybrid pasca-kuantum

Setelah memilih kebijakan SSH pasca-kuantum yang benar di titik akhir Transfer Family SFTP, Anda dapat bereksperimen dengan SFTP pasca-kuantum di Transfer Family. Anda dapat menggunakan klien SFTP (seperti OQS OpenSSH) iyang mendukung pertukaran kunci hybrid pasca-kuantum dengan mengikuti panduan dalam spesifikasi draf yang disebutkan di atas.

OQS OpenSSH adalah fork open-source OpenSSH yang menambahkan kriptografi kuantum aman ke SSH dengan menggunakan. `liboqs` `liboqs`adalah pustaka C open-source yang mengimplementasikan algoritma kriptografi tahan kuantum. OQS OpenSSH dan `liboqs` merupakan bagian dari proyek Open Quantum Safe (OQS).

Untuk menguji pertukaran kunci hybrid pasca-kuantum di Transfer Family SFTP dengan OQS OpenSSH, Anda perlu membangun OQS OpenSSH seperti yang dijelaskan dalam README proyek. Setelah Anda membangun OQS OpenSSH, Anda dapat menjalankan contoh klien SFTP untuk terhubung ke titik akhir SFTP Anda (misalnya,`s-1111aaaa2222bbbb3.server.transfer.us-west-2.amazonaws.com`) dengan menggunakan metode pertukaran kunci hybrid pasca-kuantum, seperti yang ditunjukkan pada perintah berikut.

```
./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
```

Pada perintah sebelumnya, ganti item berikut dengan informasi Anda sendiri:

- Ganti *UserName_Private_KEY_PEM_FILE dengan file kunci pribadi pengguna SFTP* yang dikodekan PEM
- Ganti *nama pengguna dengan nama pengguna* SFTP
- Ganti *server-id dengan ID* server Transfer Family
- Ganti *region-id* dengan wilayah sebenarnya di mana server Transfer Family Anda berada

Konfirmasikan pertukaran kunci hibrida pasca-kuantum di SFTP

Untuk mengonfirmasi bahwa pertukaran kunci hybrid pasca-kuantum digunakan selama koneksi SSH untuk SFTP ke Transfer Family, periksa output klien. Secara opsional, Anda dapat menggunakan

program pengambilan paket. Jika Anda menggunakan klien OpenSSH OpenSSH Open Quantum Safe, outputnya akan terlihat mirip dengan yang berikut (menghilangkan informasi yang tidak relevan untuk singkatnya):

```
$./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>
```

Output menunjukkan bahwa negosiasi klien terjadi menggunakan `ecdh-nistp384-kyber-768r3-sha384-d00@openquantumsafe.org` metode hibrida pasca-kuantum dan berhasil membuat sesi SFTP.

Perlindungan data di AWS Transfer Family

Model tanggung jawab AWS bersama model berlaku untuk perlindungan data di AWS Transfer Family (Transfer Family). Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Konten ini mencakup konfigurasi keamanan dan tugas manajemen untuk AWS layanan yang Anda gunakan. Untuk informasi selengkapnya tentang privasi data, lihat FAQ privasi data. Untuk informasi tentang perlindungan data di Eropa, lihat model tanggung jawab AWS bersama dan posting blog GDPR di Blog AWS Keamanan.

Untuk tujuan perlindungan data, kami menyarankan Anda untuk melindungi kredensyal AWS akun dan menyiapkan akun pengguna individu dengan AWS Identity and Access Management (IAM). Dengan cara ini, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugas mereka. Kami juga merekomendasikan agar Anda mengamankan data Anda dengan cara-cara berikut ini:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan sumber daya. AWS Kami mendukung TLS 1.2.
- Siapkan API dan logging aktivitas pengguna dengan AWS CloudTrail.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default dalam AWS layanan.
- Gunakan layanan keamanan terkelola lanjutan seperti Amazon Macie, yang membantu menemukan dan mengamankan data pribadi yang disimpan di Amazon Simple Storage Service (Amazon S3).
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-2 ketika mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Untuk informasi selengkapnya tentang titik akhir FIPS yang tersedia, lihat Standar pemrosesan informasi federal (FIPS) 140-2.

Kami sangat merekomendasikan agar Anda tidak memasukkan informasi identifikasi sensitif apapun, seperti nomor rekening pelanggan Anda, ke dalam kolom isian teks bebas seperti kolom Nama. Ini termasuk saat Anda bekerja dengan Transfer Family atau AWS layanan lain menggunakan konsol, API AWS CLI, atau AWS SDK. Data konfigurasi apa pun yang Anda masukkan ke dalam konfigurasi layanan Transfer Family, atau konfigurasi layanan lain, dapat diambil untuk dimasukkan dalam log diagnostik. Saat Anda memberikan URL ke server eksternal, jangan sertakan informasi kredensial di URL untuk memvalidasi permintaan Anda ke server tersebut.

Sebaliknya, data dari operasi unggah dan pengunduhan masuk dan keluar dari server Transfer Family diperlakukan sebagai sepenuhnya pribadi dan tidak pernah ada di luar saluran terenkripsi — seperti koneksi SFTP atau FTPS. Data ini hanya dapat diakses oleh orang yang berwenang.

Topik

- Enkripsi data di Amazon S3
- Manajemen kunci

Enkripsi data di Amazon S3

AWS Transfer Family menggunakan opsi enkripsi default yang Anda tetapkan untuk bucket Amazon S3 untuk mengenkripsi data Anda. Saat Anda mengaktifkan enkripsi pada bucket, semua objek dienkripsi saat disimpan di bucket. Objek dienkripsi dengan menggunakan enkripsi sisi server dengan kunci terkelola Amazon S3 (SSE-S3) atau () kunci terkelola (SSE-KMS). AWS Key Management Service AWS KMS Untuk informasi tentang enkripsi sisi server, lihat Melindungi data menggunakan enkripsi sisi server di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Langkah-langkah berikut menunjukkan cara mengenkripsi data di AWS Transfer Family.

Untuk memungkinkan enkripsi di AWS Transfer Family

1. Aktifkan enkripsi default untuk bucket Amazon S3 Anda. Untuk petunjuknya, lihat enkripsi default Amazon S3 untuk bucket S3 di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.
2. Perbarui kebijakan peran AWS Identity and Access Management (IAM) yang dilampirkan ke pengguna untuk memberikan izin required AWS Key Management Service (AWS KMS).
3. Jika Anda menggunakan kebijakan sesi untuk pengguna, kebijakan sesi harus memberikan AWS KMS izin yang diperlukan.

Contoh berikut menunjukkan kebijakan IAM yang memberikan izin minimum yang diperlukan saat menggunakan bucket Amazon S3 AWS Transfer Family yang diaktifkan untuk enkripsi. AWS KMS Sertakan kebijakan contoh ini dalam kebijakan peran IAM pengguna dan kebijakan sesi, jika Anda menggunakannya.

```
{
 "Sid": "Stmt1544140969635",
 "Action": [
  "kms:Decrypt",
  "kms:Encrypt",
  "kms:GenerateDataKey"
 ],
 "Effect": "Allow",
 "Resource": "arn:aws:kms:region:account-id:key/kms-key-id"
}
```

> **Note**
>
> ID kunci KMS yang Anda tentukan dalam kebijakan ini harus sama dengan yang ditentukan untuk enkripsi default pada langkah 1.
> Root, atau peran IAM yang digunakan untuk pengguna, harus diizinkan dalam kebijakan AWS KMS kunci. Untuk informasi tentang kebijakan AWS KMS utama, lihat Menggunakan kebijakan utama di AWS KMS di Panduan AWS Key Management Service Pengembang.

Manajemen kunci

Di bagian ini, Anda dapat menemukan informasi tentang kunci SSH, termasuk cara membuatnya dan cara memutarnya. Untuk detail tentang penggunaan Transfer Family with AWS Lambda to manage keys, lihat posting blog Mengaktifkan manajemen kunci layanan mandiri pengguna dengan A AWS Transfer Family dan. AWS Lambda

> **Note**
>
> AWS Transfer Family menerima kunci RSA, ECDSA, dan ED25519.

Bagian ini juga mencakup cara membuat dan mengelola kunci Pretty Good Privacy (PGP).

Topik

- Algoritma yang didukung untuk kunci pengguna dan server
- Buat kunci SSH untuk pengguna yang dikelola layanan
- Putar tombol SSH
- Buat dan kelola kunci PGP
- Klien PGP yang didukung

Algoritma yang didukung untuk kunci pengguna dan server

Algoritma kunci berikut didukung untuk pasangan kunci pengguna dan server di dalamnya. AWS Transfer Family

> **Note**
>
> Untuk algoritme yang dapat digunakan dengan dekripsi PGP dalam alur kerja, lihat Algoritma yang didukung untuk pasangan kunci PGP.

- Untuk ED25519: `ssh-ed25519`
- Untuk RSA:
  - `rsa-sha2-256`
  - `rsa-sha2-512`
- Untuk ECDSA:
  - `ecdsa-sha2-nistp256`
  - `ecdsa-sha2-nistp384`
  - `ecdsa-sha2-nistp521`

> **Note**
>
> Kami mendukung `ssh-rsa` SHA1 untuk kebijakan keamanan lama kami. Untuk detailnya, lihat Algoritma kriptografi.

Buat kunci SSH untuk pengguna yang dikelola layanan

Anda dapat mengatur server Anda untuk mengautentikasi pengguna menggunakan metode otentikasi terkelola layanan, tempat nama pengguna dan kunci SSH disimpan dalam layanan. Kunci SSH publik pengguna diunggah ke server sebagai properti pengguna. Kunci ini digunakan oleh server sebagai bagian dari proses otentikasi berbasis kunci standar. Setiap pengguna dapat memiliki beberapa kunci SSH publik pada file dengan server individual. Untuk batasan jumlah kunci yang dapat disimpan per pengguna, lihat AWS Transfer Family titik akhir dan kuota di. Referensi Umum Amazon Web

Sebagai alternatif dari metode otentikasi terkelola layanan, Anda dapat mengautentikasi pengguna menggunakan penyedia identitas kustom, atau. AWS Directory Service for Microsoft Active Directory Untuk informasi selengkapnya, lihat Bekerja dengan penyedia identitas khusus atau Menggunakan penyedia identitas AWS Directory Service.

Server hanya dapat mengautentikasi pengguna menggunakan satu metode (layanan dikelola, layanan direktori, atau penyedia identitas kustom), dan metode itu tidak dapat diubah setelah server dibuat.

Topik

- Membuat kunci SSH di macOS, Linux, atau Unix
- Membuat kunci SSH di Microsoft Windows
- Mengkonversi kunci publik SSH2 ke format PEM

Membuat kunci SSH di macOS, Linux, atau Unix

Pada sistem operasi macOS, Linux, atau Unix, Anda menggunakan `ssh-keygen` perintah untuk membuat kunci publik SSH dan kunci pribadi SSH yang juga dikenal sebagai key pair.

Untuk membuat kunci SSH pada sistem operasi macOS, Linux, atau Unix

1. Pada sistem operasi macOS, Linux, atau Unix, buka terminal perintah.
2. AWS Transfer Family menerima kunci berformat RSA-, ECDSA-, dan ED25519. Pilih perintah yang sesuai berdasarkan jenis pasangan kunci yang Anda hasilkan.

> **Note**
>
> Dalam contoh berikut, kami tidak menentukan frasa sandi: dalam hal ini, alat meminta Anda untuk memasukkan frasa sandi Anda dan kemudian mengulanginya untuk memverifikasi. Membuat frasa sandi menawarkan perlindungan yang lebih baik untuk kunci pribadi Anda, dan mungkin juga meningkatkan keamanan sistem secara keseluruhan. Anda tidak dapat memulihkan frasa sandi Anda: jika Anda lupa, Anda harus membuat kunci baru.
> Namun, jika Anda membuat kunci host server, Anda harus menentukan frasa sandi kosong, dengan menentukan `-N ""` opsi dalam perintah (atau dengan menekan **`Enter`** dua kali saat diminta), karena server Transfer Family tidak dapat meminta kata sandi saat start-up.

- Untuk menghasilkan key pair RSA 4096-bit:

```
ssh-keygen -t rsa -b 4096 -f key_name
```

- Untuk menghasilkan pasangan kunci ECDSA 521-bit (ECDSA memiliki ukuran bit 256, 384, dan 521):

```
ssh-keygen -t ecdsa -b 521 -f key_name
```

- Untuk menghasilkan key pair ED25519:

```
ssh-keygen -t ed25519 -f key_name
```

> **Note**
>
> *key_name*adalah nama file key pair SSH.

Berikut ini menunjukkan contoh `ssh-keygen` output.

```
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**
>
> Ketika Anda menjalankan `ssh-keygen` perintah seperti yang ditunjukkan sebelumnya, itu menciptakan kunci publik dan pribadi sebagai file dalam direktori saat ini.

Key pair SSH Anda sekarang siap digunakan. Ikuti langkah 3 dan 4 untuk menyimpan kunci publik SSH untuk pengguna yang dikelola layanan Anda. Pengguna ini menggunakan kunci ketika mereka mentransfer file pada endpoint server Transfer Family.

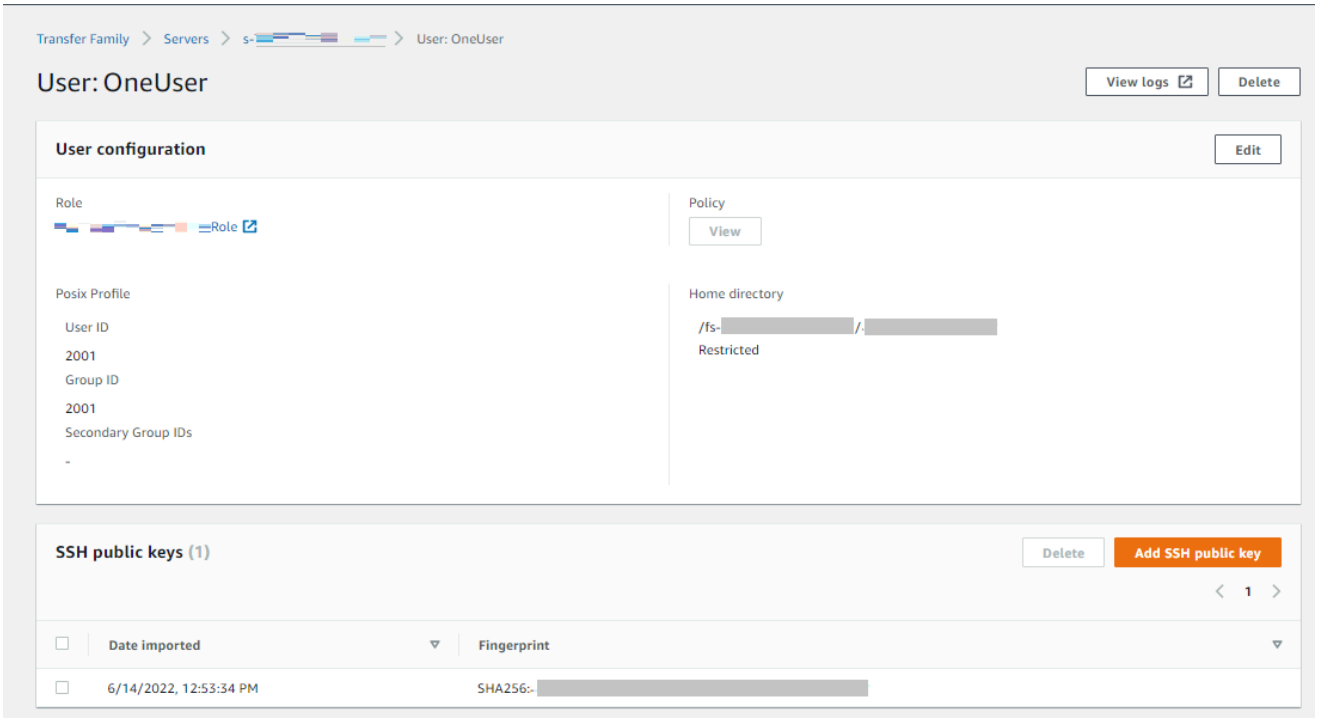
3. Arahkan ke *`key_name`*`.pub` file dan buka.
4. Salin teks dan tempel di kunci publik SSH untuk pengguna yang dikelola layanan.

   a. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/, lalu pilih Server dari panel navigasi.

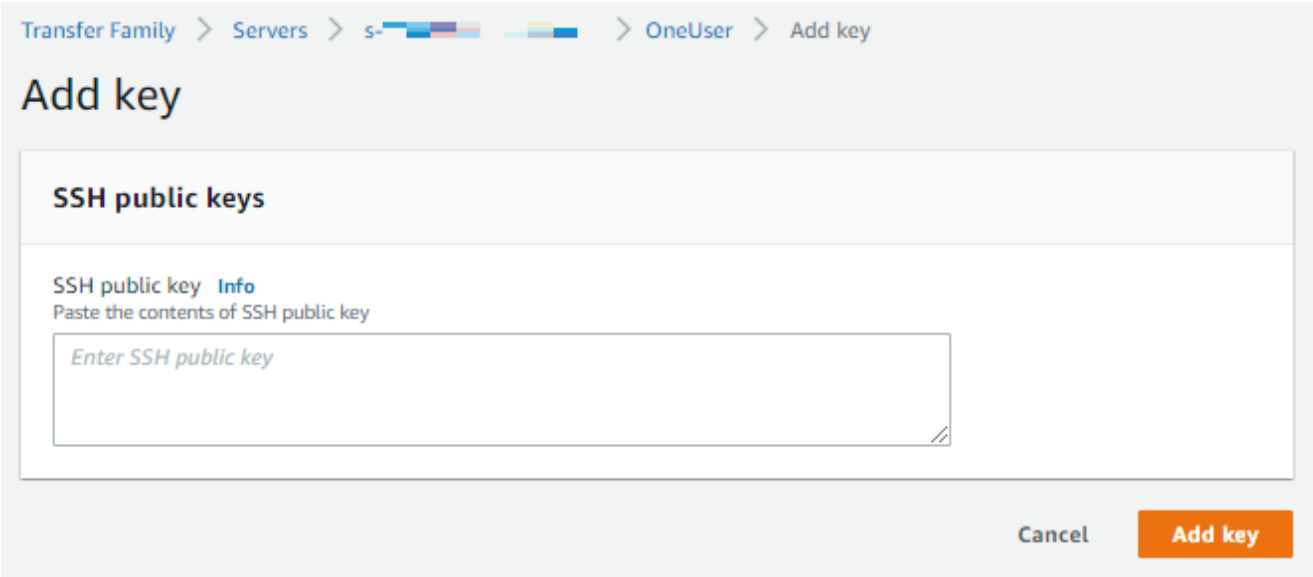
   b. Pada halaman Server, pilih ID Server untuk server yang berisi pengguna yang ingin Anda perbarui.

   c. Pilih pengguna yang Anda tambahkan kunci publiknya.

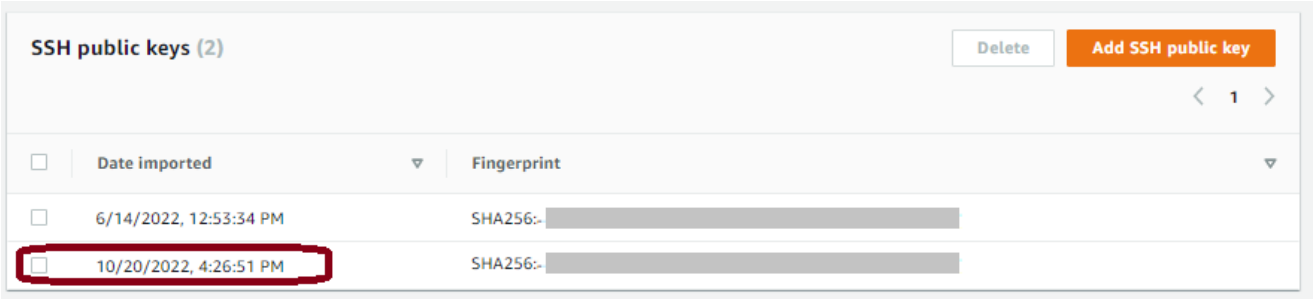
   d. Di panel kunci publik SSH, pilih Tambahkan kunci publik SSH.



e. Tempelkan teks kunci publik yang Anda hasilkan ke dalam kotak teks kunci publik SSH, lalu pilih Tambah kunci.



Kunci baru tercantum di panel kunci publik SSH.



Membuat kunci SSH di Microsoft Windows

Windows menggunakan format SSH key pair yang sedikit berbeda. Kunci publik harus dalam PUB format, dan kunci pribadi harus dalam PPK format. Di Windows, Anda dapat menggunakan PuttyGen untuk membuat key pair SSH dalam format yang sesuai. Anda juga dapat menggunakan PuttyGen untuk mengonversi kunci pribadi yang dihasilkan `ssh-keygen` menggunakan ke file. `.ppk`

> **Note**
>
> Jika Anda menyajikan WinSCP dengan file kunci pribadi yang tidak `.ppk` dalam format, klien itu menawarkan untuk mengubah kunci `.ppk` menjadi format untuk Anda.

Untuk tutorial tentang membuat kunci SSH dengan menggunakan PuttyGen di Windows, lihat situs web SSH.com.

Mengkonversi kunci publik SSH2 ke format PEM

AWS Transfer Family hanya menerima kunci publik berformat PEM. Jika Anda memiliki kunci publik SSH2, Anda perlu mengonversinya. Kunci publik SSH2 memiliki format berikut:

```
---- BEGIN SSH2 PUBLIC KEY ----
Comment: "rsa-key-20160402"
AAAAB3NzaC1yc2EAAAABJQAAAgEAiL0jjDdFqK/kYThqKt7THrjABTPWvXmB3URI
:
:
---- END SSH2 PUBLIC KEY ----
```

Kunci publik PEM memiliki format berikut:

```
ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAA...
```

Jalankan perintah berikut untuk mengonversi kunci publik berformat SSH2 menjadi kunci publik berformat PEM. Ganti *ssh2-key* dengan nama kunci SSH2 Anda, dan *pem-key dengan nama kunci* PEM Anda.

```
ssh-keygen -i -f ssh2-key.pub > PEM-key.pub
```

Putar tombol SSH

Untuk keamanan, kami merekomendasikan praktik terbaik untuk memutar kunci SSH Anda. Biasanya, rotasi ini ditentukan sebagai bagian dari kebijakan keamanan dan diimplementasikan dalam beberapa cara otomatis. Tergantung pada tingkat keamanan, untuk komunikasi yang sangat sensitif, key pair SSH mungkin hanya digunakan sekali. Melakukan hal ini menghilangkan risiko apa pun karena kunci yang disimpan. Namun, jauh lebih umum untuk menyimpan kredensyal SSH untuk jangka waktu tertentu dan menetapkan interval yang tidak menempatkan beban yang tidak semestinya pada pengguna. Interval waktu tiga bulan adalah hal biasa.

Ada dua metode yang digunakan untuk melakukan rotasi kunci SSH:

- Di konsol, Anda dapat mengunggah kunci publik SSH baru dan menghapus kunci publik SSH yang ada.
- Dengan menggunakan API, Anda dapat memperbarui pengguna yang ada dengan menggunakan DeleteSshPublicKeyAPI untuk menghapus kunci publik Secure Shell (SSH) pengguna dan ImportSshPublicKeyAPI untuk menambahkan kunci publik Secure Shell (SSH) baru ke akun pengguna.

Console

Untuk melakukan rotasi kunci di konsol

1. Buka AWS Transfer Family konsol di https://console.aws.amazon.com/transfer/.
2. Arahkan ke halaman Server.
3. Pilih pengenal di kolom ID Server untuk melihat halaman Detail Server.
4. Di bawah Pengguna, pilih kotak centang pengguna yang kunci publik SSH yang ingin Anda putar, lalu pilih Tindakan, lalu pilih Tambah kunci untuk melihat halaman Tambah kunci.

   atau

Pilih nama pengguna untuk melihat halaman Detail pengguna, lalu pilih Tambahkan kunci publik SSH untuk melihat halaman Tambah kunci.

5. Masukkan kunci publik SSH baru dan pilih Tambah kunci.

> **Important**
>
> Format kunci publik SSH tergantung pada jenis kunci yang Anda hasilkan.
>
> - Untuk kunci RSA, formatnya adalah`ssh-rsa` *`string`*.
> - Untuk tombol ED25519, formatnya adalah. `ssh-ed25519` *`string`*
> - Untuk kunci ECDSA, kunci dimulai dengan`ecdsa-sha2-nistp256`,`ecdsa-sha2-nistp384`, atau`ecdsa-sha2-nistp521`, tergantung pada ukuran kunci yang Anda hasilkan. String awal kemudian diikuti oleh*`string`*, mirip dengan jenis kunci lainnya.

Anda dikembalikan ke halaman detail Pengguna, dan kunci publik SSH baru yang baru saja Anda masukkan muncul di bagian kunci publik SSH.

6. Pilih kotak centang kunci lama yang ingin Anda hapus dan kemudian pilih Hapus.
7. Konfirmasikan operasi penghapusan dengan memasukkan kata**`delete`**, lalu pilih Hapus.

API

Untuk melakukan rotasi kunci menggunakan API

1. Pada sistem operasi macOS, Linux, atau Unix, buka terminal perintah.
2. Ambil kunci SSH yang ingin Anda hapus dengan memasukkan perintah berikut. Untuk menggunakan perintah ini, ganti *`serverID`* dengan ID server untuk server Transfer Family Anda, dan ganti *`username`* dengan nama pengguna Anda.

```
aws transfer describe-user --server-id='serverID' --user-name='username'
```

Perintah mengembalikan rincian tentang pengguna. Salin isi `"SshPublicKeyId":` bidang. Anda harus memasukkan nilai ini nanti dalam prosedur ini.

```
"SshPublicKeys": [ { "SshPublicKeyBody": "public-key", "SshPublicKeyId":
 "keyID",
   "DateImported": 1621969331.072 } ],
```

3. Selanjutnya, impor kunci SSH baru untuk pengguna Anda. Di perintah , masukkan perintah berikut. Untuk menggunakan perintah ini, ganti *serverID* dengan ID server untuk server Transfer Family Anda, ganti *username* dengan nama pengguna Anda, dan ganti *public-key* dengan sidik jari kunci publik baru Anda.

   ```
   aws transfer import-ssh-public-key --server-id='serverID' --user-name='username'
      --ssh-public-key-body='public-key'
   ```

   Jika perintah berhasil, tidak ada output yang dikembalikan.

4. Terakhir, hapus kunci lama dengan menjalankan perintah berikut. Untuk menggunakan perintah ini, ganti *serverID* dengan ID server untuk server Transfer Family Anda, ganti *username* dengan nama pengguna Anda, dan ganti *keyID-from-step-2* dengan nilai ID kunci yang Anda salin di langkah 2 prosedur ini

   ```
   aws transfer delete-ssh-public-key --server-id='serverID' --user-name='username'
      --ssh-public-key-id='keyID-from-step-2'
   ```

5. (Opsional) Untuk mengonfirmasi bahwa kunci lama tidak ada lagi, ulangi langkah 2.

Buat dan kelola kunci PGP

Anda dapat menggunakan dekripsi Pretty Good Privacy (PGP) dengan file yang diproses Transfer Family dengan alur kerja. Untuk menggunakan dekripsi dalam langkah alur kerja, Anda harus memberikan kunci PGP.

Blog AWS penyimpanan memiliki posting yang menjelaskan cara mengenkripsi dan mendekripsi file, mengenkripsi dan mendekripsi file dengan PGP dan. AWS Transfer Family

Hasilkan kunci PGP

Metode yang Anda gunakan untuk menghasilkan kunci PGP Anda tergantung pada sistem operasi Anda dan versi perangkat lunak generasi kunci yang Anda gunakan.

Jika Anda menggunakan Linux atau Unix, gunakan penginstal paket Anda untuk menginstal. gpg Tergantung pada distribusi Linux Anda, salah satu perintah berikut akan bekerja untuk Anda.

```
sudo yum install gnupg
```

```
sudo apt-get install gnupg
```

Untuk Windows atau macOS, Anda dapat mengunduh apa yang Anda butuhkan dari https://gnupg.org/download/.

Setelah Anda menginstal perangkat lunak generator kunci PGP Anda, Anda menjalankan `gpg --gen-key` perintah `gpg --full-gen-key` or untuk menghasilkan key pair.

> **Note**
>
> Jika Anda menggunakan GnuPG versi 2.3.0 atau yang lebih baru, Anda harus menjalankannya. `gpg --full-gen-key` Ketika diminta untuk jenis kunci yang akan dibuat, pilih RSA atau ECC. Namun, jika Anda memilih ECC, pastikan untuk memilih salah satu NIST atau BrainPool untuk kurva elips. Jangan memilihCurve 25519.

Algoritma yang didukung untuk pasangan kunci PGP

Algoritma berikut didukung untuk pasangan kunci PGP:

- RSA
- Elgamal
- ECC:
  - NIST
  - BrainPool

> **Note**
>
> Tombol Curve25519 tidak didukung.

gpgSubperintah yang berguna

Berikut ini adalah beberapa subperintah yang berguna untuk`gpg`:

- `gpg --help`— Perintah ini mencantumkan opsi yang tersedia dan mungkin menyertakan beberapa contoh.
- `gpg --list-keys`— Perintah ini mencantumkan detail untuk semua pasangan kunci yang telah Anda buat.
- `gpg --fingerprint`— Perintah ini mencantumkan detail untuk semua pasangan kunci Anda, termasuk sidik jari masing-masing tombol.
- `gpg --export -a` *`user-name`*— Perintah ini mengekspor bagian kunci publik dari kunci untuk *`user-name`* yang digunakan saat kunci dihasilkan.

Kelola kunci PGP

Untuk mengelola kunci PGP Anda, Anda harus menggunakan. AWS Secrets Manager

> **Note**
>
> Nama rahasia Anda termasuk ID server Transfer Family Anda. Ini berarti Anda seharusnya sudah mengidentifikasi atau membuat server sebelum Anda dapat menyimpan informasi kunci PGP Anda. AWS Secrets Manager

Jika Anda ingin menggunakan satu kunci dan frasa sandi untuk semua pengguna Anda, Anda dapat menyimpan informasi blok kunci PGP di bawah nama rahasia`aws/transfer/`*`server-id`*`/@pgp-default`, di mana ID untuk server Transfer *`server-id`* Family Anda. Kunci default ini digunakan jika tidak ada kunci di mana *`user-name`* cocok dengan pengguna yang menjalankan alur kerja.

Atau, Anda dapat membuat kunci untuk pengguna tertentu. Dalam hal ini, format untuk nama rahasia adalah`aws/transfer/`*`server-id`*`/`*`user-name`*, di mana *`user-name`* cocok dengan pengguna yang menjalankan alur kerja untuk server Transfer Family.

> **Note**
>
> Anda dapat menyimpan maksimal 3 kunci pribadi PGP, per server Transfer Family, per pengguna.

Untuk mengkonfigurasi kunci PGP untuk digunakan dengan dekripsi

1. Bergantung pada versi GPG yang Anda gunakan, jalankan salah satu perintah berikut untuk menghasilkan key pair PGP yang tidak menggunakan algoritma enkripsi Curve 25519.

   - Jika Anda menggunakan **GnuPG** versi 2.3.0 atau yang lebih baru, jalankan perintah berikut:

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

     Anda dapat memilih**RSA**, atau, jika Anda memilih**ECC**, Anda dapat memilih salah satu **NIST** atau **BrainPool** untuk kurva elips. Jika Anda `gpg --gen-key` menjalankannya, Anda membuat key pair yang menggunakan algoritma enkripsi ECC Curve 25519, yang saat ini tidak kami dukung untuk kunci PGP.

   - Untuk versi **GnuPG** sebelum 2.3.0, Anda dapat menggunakan perintah berikut, karena RSA adalah jenis enkripsi default.

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

   > **Important**
   >
   > Selama proses pembuatan kunci, Anda harus memberikan frasa sandi dan alamat email. Pastikan untuk mencatat nilai-nilai ini. Anda harus memberikan frasa sandi ketika Anda memasukkan detail kunci ke dalam prosedur ini AWS Secrets Manager nanti. Dan Anda harus memberikan alamat email yang sama untuk mengekspor kunci pribadi di langkah berikutnya.

2. Jalankan perintah berikut untuk mengekspor kunci pribadi. Untuk menggunakan perintah ini, ganti *`private.pgp`* dengan nama file untuk menyimpan blok kunci pribadi, dan *`marymajor@example.com`* dengan alamat email yang Anda gunakan saat Anda membuat key pair.

   ```
   gpg --output private.pgp --armor --export-secret-key marymajor@example.com
   ```

3. Gunakan AWS Secrets Manager untuk menyimpan kunci PGP Anda.

   a. Masuk ke AWS Management Console dan buka AWS Secrets Manager konsol di https://console.aws.amazon.com/secretsmanager/.

b. Pada panel navigasi kiri, pilih Rahasia.

c. Pada halaman Rahasia, pilih Simpan rahasia baru.

d. Pada halaman Choose secret type, untuk Secret type, pilih Other type of secret.

e. Di bagian pasangan kunci/Nilai, pilih tab kunci/Nilai.

- Kunci — Masukkan**`PGPPrivateKey`**.

> **Note**
>
> Anda harus memasukkan **`PGPPrivateKey`** string dengan tepat: jangan menambahkan spasi sebelum atau di antara karakter.

- nilai — Tempelkan teks kunci pribadi Anda ke bidang nilai. Anda dapat menemukan teks kunci pribadi Anda dalam file (misalnya,`private.pgp`) yang Anda tentukan saat Anda mengekspor kunci Anda sebelumnya dalam prosedur ini. Kuncinya dimulai dengan `-----BEGIN PGP PRIVATE KEY BLOCK-----` dan diakhiri dengan`-----END PGP PRIVATE KEY BLOCK-----`.

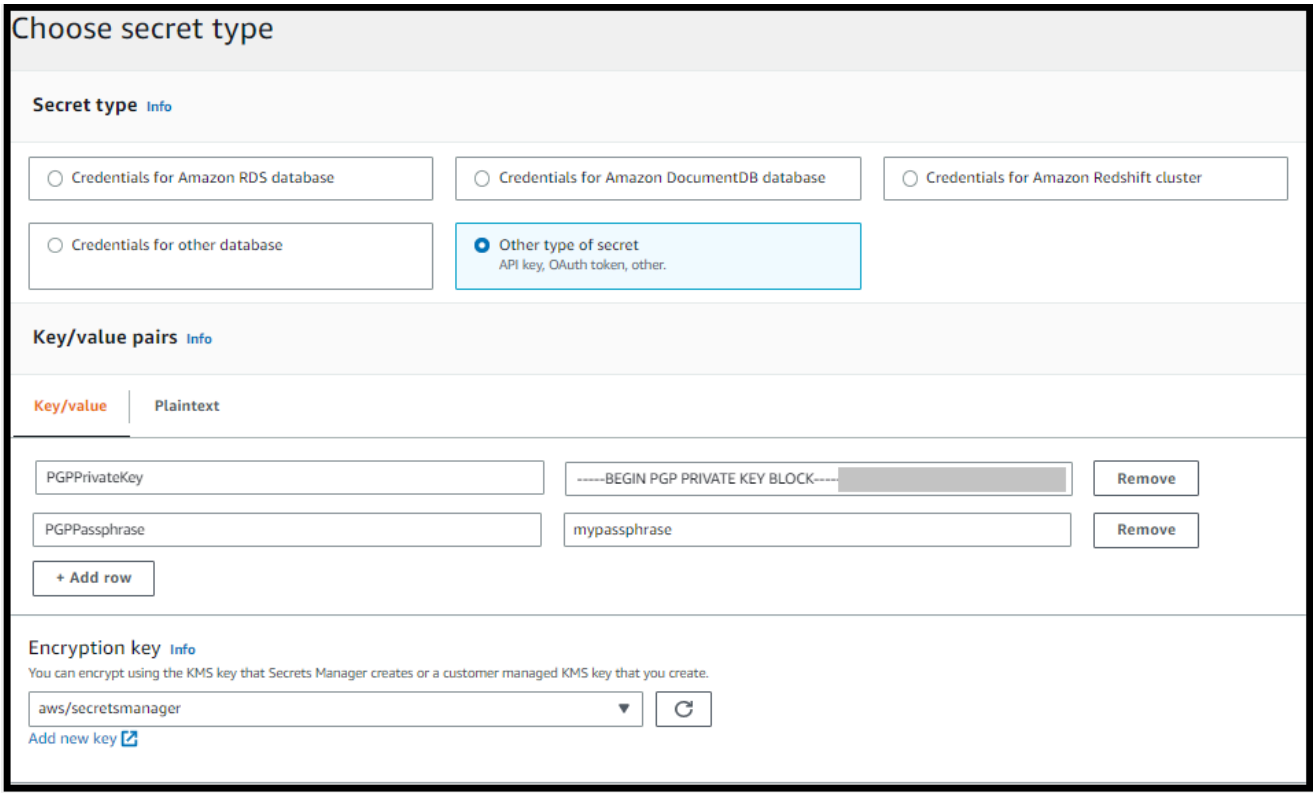
> **Note**
>
> Pastikan bahwa blok teks hanya berisi kunci pribadi dan tidak mengandung kunci publik juga.

f. Pilih Tambahkan baris dan di bagian pasangan kunci/Nilai, pilih tab kunci/Nilai.

- Kunci — Masukkan**`PGPPassphrase`**.

> **Note**
>
> Anda harus memasukkan **`PGPPassphrase`** string dengan tepat: jangan menambahkan spasi sebelum atau di antara karakter.

- value — Masukkan kata sandi yang Anda gunakan saat membuat key pair PGP Anda.



> **Note**
>
> Anda dapat menambahkan hingga 3 set kunci dan frasa sandi. Untuk menambahkan set kedua, tambahkan dua baris baru, dan masukkan **`PGPPrivateKey2`** dan **`PGPPassphrase2`** untuk kunci, dan tempel di kunci pribadi dan frasa sandi lainnya. Untuk menambahkan set ketiga, nilai kunci harus **`PGPPrivateKey3`** dan**`PGPPassphrase3`**.

g. Pilih Berikutnya.

h. Pada halaman Konfigurasi rahasia, masukkan nama dan deskripsi untuk rahasia Anda.

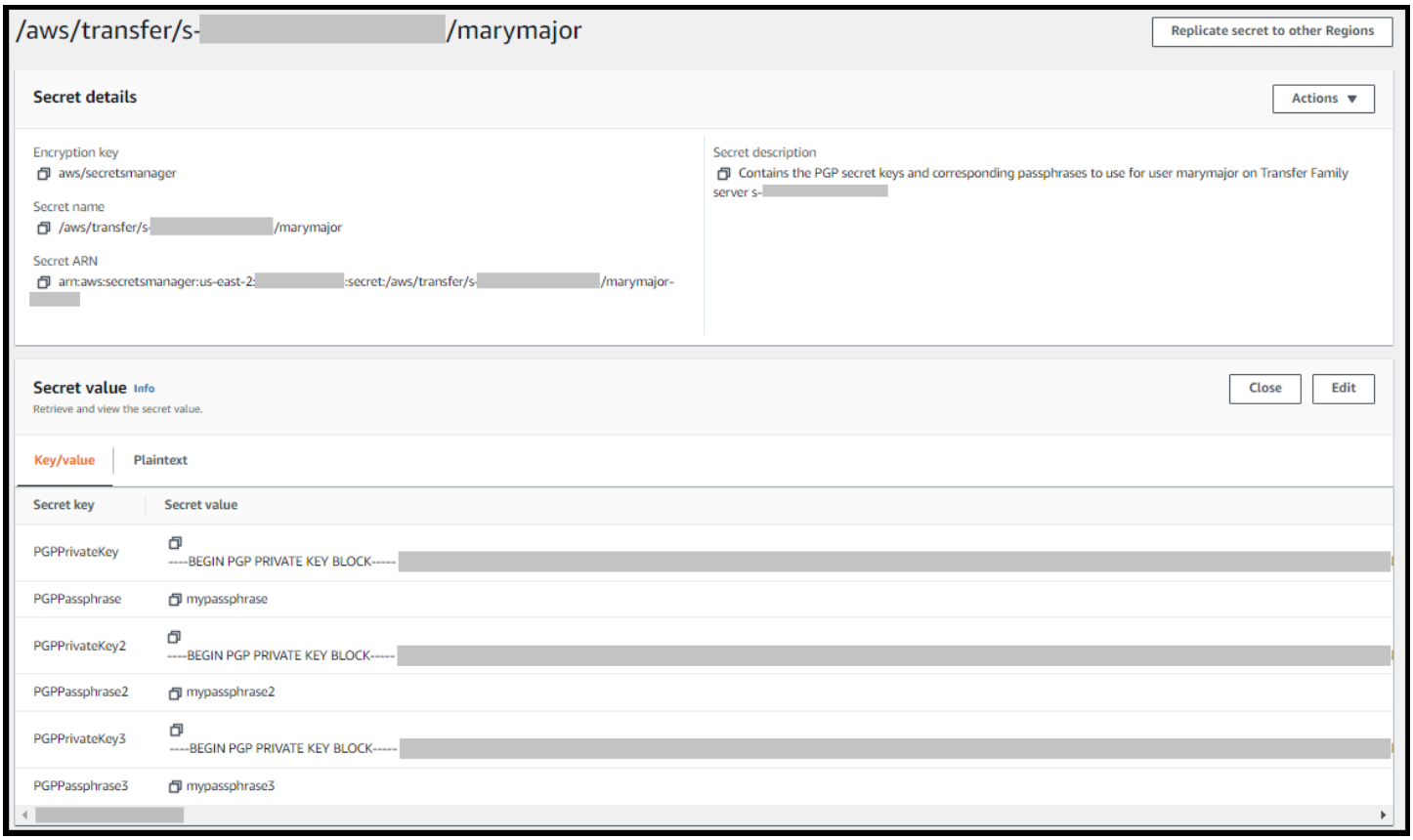
- Jika Anda membuat kunci default, yaitu kunci yang dapat digunakan oleh pengguna Transfer Family, masukkan**`aws/transfer/`*`server-id`*`/@pgp-default`**. Ganti *`server-id`* dengan ID server yang berisi alur kerja yang memiliki langkah dekripsi.
- Jika Anda membuat kunci untuk digunakan oleh pengguna Transfer Family tertentu, masukkan**`aws/transfer/`*`server-id`*`/`*`user-name`**. Ganti *`server-id`* dengan ID server yang berisi alur kerja yang memiliki langkah dekripsi, dan ganti *`user-name`*

dengan nama pengguna yang menjalankan alur kerja. *user-name*Ini disimpan di penyedia identitas yang digunakan server Transfer Family.

i. Pilih Berikutnya dan terima default pada halaman Konfigurasi rotasi. Lalu pilih Selanjutnya.

j. Pada halaman Review, pilih Store untuk membuat dan menyimpan rahasia.

Tangkapan layar berikut menunjukkan detail untuk pengguna **marymajor** untuk server Transfer Family tertentu. Contoh ini menunjukkan tiga kunci dan frasa sandi yang sesuai.

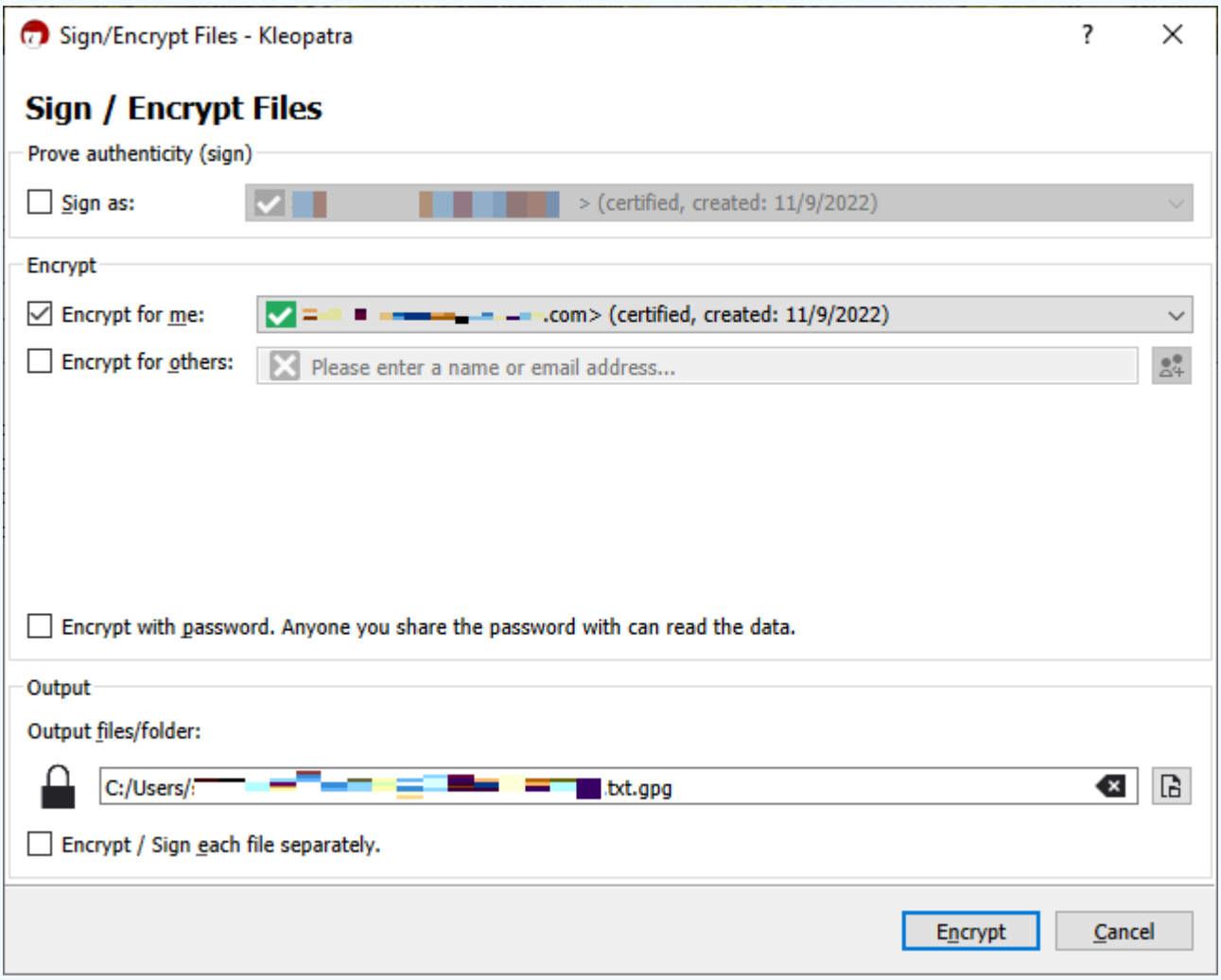


Klien PGP yang didukung

Klien berikut telah diuji dengan Transfer Family dan dapat digunakan untuk menghasilkan kunci PGP, dan untuk mengenkripsi file yang ingin Anda dekripsi dengan alur kerja.

- GPG4win + Kleopatra.

> **Note**
>
> Saat Anda memilih Sign/Encrypt Files, pastikan untuk menghapus pilihan untuk Sign as: saat ini kami tidak mendukung penandatanganan untuk file terenkripsi.



Jika Anda menandatangani file terenkripsi dan mencoba mengunggahnya ke server Transfer Family dengan alur kerja dekripsi, Anda menerima kesalahan berikut:

```
Encrypted file with signed message unsupported
```

- Versi GnuPG utama: 2.4, 2.3, 2.2, 2.0, dan 1.4.

Perhatikan bahwa klien PGP lain mungkin bekerja juga, tetapi hanya klien yang disebutkan di sini yang telah diuji dengan Transfer Family.

Identitas dan manajemen akses untuk AWS Transfer Family

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang

dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya. AWS Transfer Family IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- Audiens
- Mengautentikasi dengan identitas
- Mengelola akses menggunakan kebijakan
- Bagaimana AWS Transfer Family bekerja dengan IAM
- AWS Transfer Family contoh kebijakan berbasis identitas
- AWS Transfer Family contoh kebijakan berbasis tag
- Memecahkan masalah AWS Transfer Family identitas dan akses

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda, tergantung pada pekerjaan yang Anda lakukan. AWS Transfer Family

Pengguna layanan — Jika Anda menggunakan AWS Transfer Family layanan untuk melakukan pekerjaan Anda, maka administrator Anda memberi Anda kredensyal dan izin yang Anda butuhkan. Saat Anda menggunakan lebih banyak AWS Transfer Family fitur untuk melakukan pekerjaan Anda, Anda mungkin memerlukan izin tambahan. Memahami cara mengelola akses dapat membantu Anda meminta izin yang tepat dari administrator Anda. Jika Anda tidak dapat mengakses fitur di AWS Transfer Family, lihat Memecahkan masalah AWS Transfer Family identitas dan akses.

Administrator layanan — Jika Anda bertanggung jawab atas AWS Transfer Family sumber daya di perusahaan Anda, Anda mungkin memiliki akses penuh ke AWS Transfer Family. Tugas Anda adalah menentukan AWS Transfer Family fitur dan sumber daya mana yang harus diakses pengguna layanan Anda. Kemudian, Anda harus mengirimkan permintaan kepada administrator IAM Anda untuk mengubah izin pengguna layanan Anda. Tinjau informasi di halaman ini untuk memahami konsep Basic IAM. Untuk mempelajari lebih lanjut tentang bagaimana perusahaan Anda dapat menggunakan IAM AWS Transfer Family, lihatBagaimana AWS Transfer Family bekerja dengan IAM.

Administrator IAM – Jika Anda adalah administrator IAM, Anda mungkin ingin belajar dengan lebih detail tentang cara Anda menulis kebijakan untuk mengelola akses ke AWS Transfer Family. Untuk melihat contoh kebijakan AWS Transfer Family berbasis identitas yang dapat Anda gunakan di IAM, lihat. AWS Transfer Family contoh kebijakan berbasis identitas

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensyal identitas Anda. Anda harus diautentikasi (masuk ke AWS) sebagai Pengguna root akun AWS, sebagai pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk AWS sebagai identitas federasi dengan menggunakan kredensyal yang disediakan melalui sumber identitas. AWS IAM Identity Center Pengguna (IAM Identity Center), autentikasi masuk tunggal perusahaan Anda, dan kredensyal Google atau Facebook Anda adalah contoh identitas federasi. Saat Anda masuk sebagai identitas gabungan, administrator Anda sebelumnya menyiapkan federasi identitas menggunakan peran IAM. Ketika Anda mengakses AWS dengan menggunakan federasi, Anda secara tidak langsung mengambil peran.

Bergantung pada jenis pengguna Anda, Anda dapat masuk ke AWS Management Console atau portal AWS akses. Untuk informasi selengkapnya tentang masuk AWS, lihat Cara masuk ke Panduan AWS Sign-In Pengguna Anda Akun AWS.

Jika Anda mengakses AWS secara terprogram, AWS sediakan kit pengembangan perangkat lunak (SDK) dan antarmuka baris perintah (CLI) untuk menandatangani permintaan Anda secara kriptografis dengan menggunakan kredensyal Anda. Jika Anda tidak menggunakan AWS alat, Anda harus menandatangani permintaan sendiri. Untuk informasi selengkapnya tentang penggunaan metode yang disarankan untuk menandatangani permintaan sendiri, lihat Menandatangani permintaan AWS API di Panduan Pengguna IAM.

Apa pun metode autentikasi yang digunakan, Anda mungkin diminta untuk menyediakan informasi keamanan tambahan. Misalnya, AWS merekomendasikan agar Anda menggunakan otentikasi multi-faktor (MFA) untuk meningkatkan keamanan akun Anda. Untuk mempelajari selengkapnya, lihat Autentikasi multi-faktor dalam Panduan Pengguna AWS IAM Identity Center dan Menggunakan autentikasi multi-faktor (MFA) di AWS dalam Panduan Pengguna IAM.

Pengguna root akun AWS

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Identitas ini disebut pengguna Akun AWS root dan diakses dengan masuk dengan alamat email dan kata sandi yang Anda gunakan untuk membuat akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari Anda. Lindungi kredensial pengguna root Anda dan gunakan kredensial tersebut untuk melakukan tugas yang hanya dapat dilakukan pengguna root. Untuk daftar lengkap tugas

yang mengharuskan Anda masuk sebagai pengguna root, lihat Tugas yang memerlukan kredensial pengguna root dalam Panduan Pengguna IAM.

Identitas terfederasi

Sebagai praktik terbaik, mewajibkan pengguna manusia, termasuk pengguna yang memerlukan akses administrator, untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS dengan menggunakan kredensyal sementara.

Identitas federasi adalah pengguna dari direktori pengguna perusahaan Anda, penyedia identitas web, direktori Pusat Identitas AWS Directory Service, atau pengguna mana pun yang mengakses Layanan AWS dengan menggunakan kredensil yang disediakan melalui sumber identitas. Ketika identitas federasi mengakses Akun AWS, mereka mengambil peran, dan peran memberikan kredensi sementara.

Untuk manajemen akses terpusat, kami sarankan Anda menggunakan AWS IAM Identity Center. Anda dapat membuat pengguna dan grup di Pusat Identitas IAM, atau Anda dapat menghubungkan dan menyinkronkan ke sekumpulan pengguna dan grup di sumber identitas Anda sendiri untuk digunakan di semua aplikasi Akun AWS dan aplikasi Anda. Untuk informasi tentang Pusat Identitas IAM, lihat Apa itu Pusat Identitas IAM? dalam Panduan Pengguna AWS IAM Identity Center .

Pengguna dan grup IAM

Pengguna IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus untuk satu orang atau aplikasi. Jika memungkinkan, sebaiknya andalkan kredensial sementara daripada membuat pengguna IAM yang memiliki kredensial jangka panjang seperti kata sandi dan kunci akses. Namun, jika Anda memiliki kasus penggunaan khusus yang memerlukan kredensial jangka panjang dengan pengguna IAM, kami sarankan Anda merotasi kunci akses. Untuk informasi selengkapnya, lihat Rotasikan kunci akses secara rutin untuk kasus penggunaan yang memerlukan kredensial jangka panjang dalam Panduan Pengguna IAM.

Grup IAM adalah identitas yang menentukan kumpulan pengguna IAM. Anda tidak dapat masuk sebagai grup. Anda dapat menggunakan grup untuk menentukan izin untuk beberapa pengguna sekaligus. Grup membuat izin lebih mudah dikelola untuk sekelompok besar pengguna. Misalnya, Anda dapat memiliki grup yang bernama IAMAdmins dan memberikan izin kepada grup tersebut untuk mengelola sumber daya IAM.

Pengguna berbeda dari peran. Pengguna secara unik terkait dengan satu orang atau aplikasi, tetapi peran dimaksudkan untuk dapat digunakan oleh siapa pun yang membutuhkannya. Pengguna memiliki kredensial jangka panjang permanen, tetapi peran memberikan kredensial sementara. Untuk

mempelajari selengkapnya, silakan lihat Kapan harus membuat pengguna IAM (bukan peran) dalam Panduan Pengguna IAM.

Peran IAM

Peran IAM adalah identitas dalam diri Anda Akun AWS yang memiliki izin khusus. Peran ini mirip dengan pengguna IAM, tetapi tidak terkait dengan orang tertentu. Anda dapat mengambil peran IAM untuk sementara AWS Management Console dengan beralih peran. Anda dapat mengambil peran dengan memanggil operasi AWS CLI atau AWS API atau dengan menggunakan URL kustom. Untuk informasi selengkapnya tentang cara menggunakan peran, lihat Menggunakan peran IAM dalam Panduan Pengguna IAM.

Peran IAM dengan kredensial sementara berguna dalam situasi berikut:

- Akses pengguna gabungan – Untuk menetapkan izin ke sebuah identitas gabungan, Anda dapat membuat peran dan menentukan izin untuk peran tersebut. Saat identitas terfederasi mengautentikasi, identitas tersebut akan dikaitkan dengan peran dan diberi izin yang ditentukan oleh peran tersebut. Untuk informasi tentang peran-peran untuk federasi, lihat Membuat peran untuk Penyedia Identitas pihak ketiga dalam Panduan Pengguna IAM. Jika Anda menggunakan Pusat Identitas IAM, Anda perlu mengonfigurasi set izin. Untuk mengontrol apa yang dapat diakses identitas Anda setelah identitas tersebut diautentikasi, Pusat Identitas IAM mengorelasikan izin yang diatur ke peran dalam IAM. Untuk informasi tentang rangkaian izin, lihat Rangkaian izin dalam Panduan Pengguna AWS IAM Identity Center .
- Izin pengguna IAM sementara – Pengguna atau peran IAM dapat mengambil peran IAM guna mendapatkan berbagai izin secara sementara untuk tugas tertentu.
- Akses lintas akun – Anda dapat menggunakan peran IAM untuk mengizinkan seseorang (pengguna utama tepercaya) dengan akun berbeda untuk mengakses sumber daya yang ada di akun Anda. Peran adalah cara utama untuk memberikan akses lintas akun. Namun, dengan beberapa Layanan AWS, Anda dapat melampirkan kebijakan secara langsung ke sumber daya (alih-alih menggunakan peran sebagai proxy). Untuk mempelajari perbedaan antara kebijakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Bagaimana peran IAM berbeda dari kebijakan berbasis sumber daya dalam Panduan Pengguna IAM.
- Akses lintas layanan — Beberapa Layanan AWS menggunakan fitur lain Layanan AWS. Sebagai contoh, ketika Anda melakukan panggilan dalam suatu layanan, biasanya layanan tersebut menjalankan aplikasi di Amazon EC2 atau menyimpan objek di Amazon S3. Sebuah layanan mungkin melakukannya menggunakan izin prinsipal yang memanggil, menggunakan peran layanan, atau peran terkait layanan.

  - Sesi akses teruskan (FAS) — Saat Anda menggunakan pengguna atau peran IAM untuk melakukan tindakan AWS, Anda dianggap sebagai prinsipal. Saat Anda menggunakan beberapa layanan, Anda mungkin melakukan tindakan yang kemudian memulai tindakan lain di layanan yang berbeda. FAS menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Permintaan FAS hanya dibuat ketika layanan menerima permintaan yang memerlukan interaksi dengan orang lain Layanan AWS atau sumber daya untuk menyelesaikannya. Dalam hal ini, Anda harus memiliki izin untuk melakukan kedua tindakan tersebut. Untuk detail kebijakan saat membuat permintaan FAS, lihat Teruskan sesi akses.
  - Peran layanan – Peran layanan adalah peran IAM yang diambil oleh layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, memodifikasi, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat Membuat peran untuk mendelegasikan izin ke Layanan AWS dalam Panduan pengguna IAM.
  - Peran terkait layanan — Peran terkait layanan adalah jenis peran layanan yang ditautkan ke peran layanan. Layanan AWS Layanan dapat menggunakan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.
- Aplikasi yang berjalan di Amazon EC2 — Anda dapat menggunakan peran IAM untuk mengelola kredensyal sementara untuk aplikasi yang berjalan pada instans EC2 dan membuat atau permintaan API. AWS CLI AWS Cara ini lebih dianjurkan daripada menyimpan kunci akses dalam instans EC2. Untuk menetapkan AWS peran ke instans EC2 dan membuatnya tersedia untuk semua aplikasinya, Anda membuat profil instance yang dilampirkan ke instance. Profil instans berisi peran dan memungkinkan program yang berjalan di instans EC2 mendapatkan kredensial sementara. Untuk informasi selengkapnya, lihat Menggunakan peran IAM untuk memberikan izin ke aplikasi yang berjalan di instans Amazon EC2 dalam Panduan Pengguna IAM.

Untuk mempelajari apakah kita harus menggunakan peran IAM atau pengguna IAM, lihat Kapan harus membuat peran IAM (bukan pengguna) dalam Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan adalah objek AWS yang, ketika dikaitkan dengan identitas atau sumber daya, menentukan izinnya. AWS mengevaluasi kebijakan ini ketika prinsipal (pengguna, pengguna root, atau sesi peran) membuat permintaan. Izin dalam kebijakan dapat menentukan permintaan yang diizinkan atau ditolak. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk

informasi selengkapnya tentang struktur dan konten dokumen kebijakan JSON, lihat Gambaran umum kebijakan JSON dalam Panduan Pengguna IAM.

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Artinya, pengguna utama manakah yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Untuk memberikan izin kepada pengguna untuk melakukan tindakan pada sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM. Administrator kemudian dapat menambahkan kebijakan IAM ke peran, dan pengguna dapat menjalankan peran.

Kebijakan IAM mendefinisikan izin untuk suatu tindakan terlepas dari metode yang Anda gunakan untuk operasi. Sebagai contoh, anggap saja Anda memiliki kebijakan yang mengizinkan tindakan `iam:GetRole`. Pengguna dengan kebijakan tersebut bisa mendapatkan informasi peran dari AWS Management Console, API AWS CLI, atau AWS API.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan pengguna dan peran, di sumber daya mana, dan dengan ketentuan apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat Membuat kebijakan IAM dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat dikategorikan lebih lanjut sebagai kebijakan inline atau kebijakan terkelola. Kebijakan inline disematkan langsung ke satu pengguna, grup, atau peran. Kebijakan terkelola adalah kebijakan mandiri yang dapat Anda lampirkan ke beberapa pengguna, grup, dan peran dalam. Akun AWS Kebijakan AWS terkelola mencakup kebijakan terkelola dan kebijakan yang dikelola pelanggan. Untuk mempelajari cara memilih antara kebijakan terkelola atau kebijakan inline, lihat Memilih antara kebijakan terkelola dan kebijakan inline dalam Panduan Pengguna IAM.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya yang dilampiri kebijakan tersebut, kebijakan ini menentukan jenis tindakan yang

dapat dilakukan oleh pengguna utama tertentu di sumber daya tersebut dan apa ketentuannya. Anda harus menentukan pengguna utama dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM dalam kebijakan berbasis sumber daya.

Daftar kontrol akses (ACL)

Daftar kontrol akses (ACL) mengendalikan pengguna utama mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL sama dengan kebijakan berbasis sumber daya, meskipun tidak menggunakan format dokumen kebijakan JSON.

Amazon S3, AWS WAF, dan Amazon VPC adalah contoh layanan yang mendukung ACL. Untuk mempelajari ACL selengkapnya, silakan lihat Gambaran umum daftar kontrol akses (ACL) di Panduan Developer Layanan Penyimpanan Ringkas Amazon.

Tipe kebijakan lain

AWS mendukung jenis kebijakan tambahan yang kurang umum. Tipe-tipe kebijakan ini dapat mengatur izin maksimum yang diberikan kepada Anda berdasarkan tipe kebijakan yang lebih umum.

- Batasan izin – Batasan izin adalah fitur lanjutan di mana Anda menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM (pengguna atau peran IAM). Anda dapat menetapkan batasan izin untuk suatu entitas. Izin yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas milik entitas dan batasan izinnya. Kebijakan berbasis sumber daya yang menentukan pengguna atau peran dalam bidang `Principal` tidak dibatasi oleh batasan izin. Penolakan secara eksplisit terhadap salah satu kebijakan ini akan mengesampingkan izin tersebut. Untuk informasi selengkapnya tentang batasan izin, lihat Batasan izin untuk entitas IAM dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCP) — SCP adalah kebijakan JSON yang menentukan izin maksimum untuk organisasi atau unit organisasi (OU) di. AWS Organizations AWS Organizations adalah layanan untuk mengelompokkan dan mengelola secara terpusat beberapa Akun AWS yang dimiliki bisnis Anda. Jika Anda mengaktifkan semua fitur di sebuah organisasi, maka Anda dapat menerapkan kebijakan kontrol layanan (SCP) ke salah satu atau semua akun Anda. SCP membatasi izin untuk entitas di akun anggota, termasuk masing-masing. Pengguna root akun AWS Untuk informasi selengkapnya tentang Organisasi dan SCP, lihat Cara kerja SCP dalam Panduan Pengguna AWS Organizations .

- Kebijakan sesi – Kebijakan sesi adalah kebijakan lanjutan yang Anda teruskan sebagai parameter saat Anda membuat sesi sementara secara terprogram untuk peran atau pengguna gabungan. Izin sesi yang dihasilkan adalah perpotongan antara kebijakan berbasis identitas pengguna atau peran dan kebijakan sesi. Izin juga bisa datang dari kebijakan berbasis sumber daya. Penolakan eksplisit dalam salah satu kebijakan ini membatalkan izin. Untuk informasi selengkapnya, lihat Kebijakan sesi dalam Panduan Pengguna IAM.

Beberapa jenis kebijakan

Ketika beberapa jenis kebijakan berlaku untuk sebuah permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat Logika evaluasi kebijakan di Panduan Pengguna IAM.

Bagaimana AWS Transfer Family bekerja dengan IAM

Sebelum Anda menggunakan AWS Identity and Access Management (IAM) untuk mengelola akses AWS Transfer Family, Anda harus memahami fitur IAM apa yang tersedia untuk digunakan. AWS Transfer Family Untuk mendapatkan pandangan tingkat tinggi tentang bagaimana AWS Transfer Family dan AWS layanan lain bekerja dengan IAM, lihat AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM.

Topik

- AWS Transfer Family kebijakan berbasis identitas
- AWS Transfer Family Kebijakan berbasis sumber daya
- Otorisasi berdasarkan tanda AWS Transfer Family
- AWS Transfer Family Peran IAM

AWS Transfer Family kebijakan berbasis identitas

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan tindakan dan sumber daya yang diizinkan atau ditolak, serta kondisi di mana tindakan tersebut diperbolehkan atau ditolak. AWS Transfer Family mendukung tindakan tertentu, sumber daya, dan kunci syarat. Untuk mempelajari semua elemen yang Anda gunakan dalam kebijakan JSON, lihat referensi elemen kebijakan IAM JSON di Panduan Pengguna.AWS Identity and Access Management

Tindakan

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, pengguna utama mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Tindakan kebijakan biasanya memiliki nama yang sama dengan operasi AWS API terkait. Ada beberapa pengecualian, misalnya tindakan hanya izin yang tidak memiliki operasi API yang cocok. Ada juga beberapa operasi yang memerlukan beberapa tindakan dalam suatu kebijakan. Tindakan tambahan ini disebut tindakan dependen.

Menyertakan tindakan dalam suatu kebijakan untuk memberikan izin melakukan operasi terkait.

Tindakan kebijakan AWS Transfer Family menggunakan awalan berikut sebelum tindakan:`transfer:`. Misalnya, untuk memberikan izin kepada seseorang untuk membuat server, dengan operasi Transfer Family `CreateServer` API, Anda menyertakan `transfer:CreateServer` tindakan tersebut dalam kebijakan mereka. Pernyataan kebijakan harus menyertakan elemen `Action` atau `NotAction`. AWS Transfer Family menentukan serangkaian tindakan sendiri yang menjelaskan tugas yang dapat Anda lakukan dengan layanan ini.

Untuk menetapkan beberapa tindakan dalam satu pernyataan, pisahkan dengan koma seperti berikut.

```
"Action": [
      "transfer:action1",
      "transfer:action2"
```

Anda juga dapat menentukan beberapa tindakan menggunakan wildcard (*). Misalnya, untuk menentukan semua tindakan yang dimulai dengan kata `Describe`, sertakan tindakan berikut.

```
"Action": "transfer:Describe*"
```

Untuk melihat daftar AWS Transfer Family tindakan, lihat Tindakan yang ditentukan oleh AWS Transfer Family dalam Referensi Otorisasi Layanan.

Sumber daya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, pengguna utama mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek atau beberapa objek yang menjadi target penerapan tindakan. Pernyataan harus menyertakan elemen `Resource` atau `NotResource`. Praktik terbaiknya, tentukan sumber daya menggunakan Amazon Resource Name (ARN). Anda dapat melakukan ini untuk tindakan yang mendukung jenis sumber daya tertentu, yang dikenal sebagai izin tingkat sumber daya.

Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, misalnya operasi pencantuman, gunakan wildcard (*) untuk mengindikasikan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

```
"Resource": "*"
```

Sumber daya server Transfer Family memiliki ARN berikut.

```
arn:aws:transfer:${Region}:${Account}:server/${ServerId}
```

Misalnya, untuk menentukan server `s-01234567890abcdef` Transfer Family dalam pernyataan Anda, gunakan ARN berikut.

```
"Resource": "arn:aws:transfer:us-east-1:123456789012:server/s-01234567890abcdef"
```

Untuk informasi selengkapnya tentang format ARN, lihat Nama Sumber Daya Amazon (ARN) di Referensi Otorisasi Layanan, atau ARN IAM di Panduan Pengguna IAM.

Untuk menentukan semua instance milik akun tertentu, gunakan wildcard (*).

```
"Resource": "arn:aws:transfer:us-east-1:123456789012:server/*"
```

Beberapa AWS Transfer Family tindakan dilakukan pada beberapa sumber daya, seperti yang digunakan dalam kebijakan IAM. Dalam kasus tersebut, Anda harus menggunakan wildcard (*).

```
"Resource": "arn:aws:transfer:*:123456789012:server/*"
```

Dalam beberapa kasus, Anda perlu menentukan lebih dari satu jenis sumber daya, misalnya, jika Anda membuat kebijakan yang memungkinkan akses ke server dan pengguna Transfer Family. Untuk menentukan beberapa sumber daya dalam satu pernyataan, pisahkan ARN dengan koma.

```
"Resource": [
```

```
        "resource1",
        "resource2"
              ]
```

Untuk melihat daftar AWS Transfer Family sumber daya, lihat Jenis sumber daya yang ditentukan oleh AWS Transfer Family dalam Referensi Otorisasi Layanan.

Kunci syarat

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, pengguna utama mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Elemen `Condition` (atau blok `Condition`) memungkinkan Anda menentukan kondisi di mana suatu pernyataan akan diterapkan. Elemen `Condition` bersifat opsional. Anda dapat membuat ekspresi kondisional yang menggunakan operator kondisi, misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta.

Jika Anda menentukan beberapa elemen `Condition` dalam satu pernyataan, atau beberapa kunci dalam satu elemen `Condition`, AWS akan mengevaluasinya dengan menggunakan operasi `AND` logis. Jika Anda menentukan beberapa nilai untuk satu kunci kondisi, AWS mengevaluasi kondisi menggunakan `OR` operasi logis. Semua kondisi harus dipenuhi sebelum izin pernyataan diberikan.

Anda juga dapat menggunakan variabel placeholder saat menentukan kondisi. Misalnya, Anda dapat memberikan izin kepada pengguna IAM untuk mengakses sumber daya hanya jika izin tersebut mempunyai tag yang sesuai dengan nama pengguna IAM mereka. Untuk informasi selengkapnya, lihat Elemen kebijakan IAM: variabel dan tag di Panduan Pengguna IAM.

AWS mendukung kunci kondisi global dan kunci kondisi khusus layanan. Untuk melihat semua kunci kondisi AWS global, lihat kunci konteks kondisi AWS global di Panduan Pengguna IAM.

AWS Transfer Family mendefinisikan kumpulan kunci kondisinya sendiri dan juga mendukung penggunaan beberapa kunci kondisi global. Untuk melihat daftar kunci AWS Transfer Family kondisi, lihat Kunci kondisi untuk AWS Transfer Family dalam Referensi Otorisasi Layanan.

Contoh-contoh

Untuk melihat contoh kebijakan AWS Transfer Family berbasis identitas, lihat. AWS Transfer Family contoh kebijakan berbasis identitas

AWS Transfer Family Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang menentukan tindakan apa yang dapat dilakukan oleh prinsipal tertentu pada AWS Transfer Family sumber daya dan dalam kondisi apa. *Amazon S3 mendukung kebijakan izin berbasis sumber daya untuk bucket Amazon S3.* Kebijakan berbasis sumber daya mengizinkan Anda memberikan izin penggunaan ke akun lain berdasarkan penggunaan sumber daya. *Anda juga dapat menggunakan kebijakan berbasis sumber daya untuk mengizinkan AWS layanan mengakses bucket Amazon S3 Anda.*

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan seluruh akun atau entitas IAM di akun lain sebagai prinsipal di kebijakan berbasis sumber daya. Menambahkan pengguna utama lintas akun ke kebijakan berbasis sumber daya bagian dari membangun hubungan kepercayaan. Ketika prinsipal dan sumber daya berada di AWS akun yang berbeda, Anda juga harus memberikan izin entitas utama untuk mengakses sumber daya. Berikan izin dengan melampirkan kebijakan berbasis identitas ke entitas tersebut. Namun, jika kebijakan berbasis sumber daya memberikan akses ke prinsipal dalam akun yang sama, tidak diperlukan kebijakan berbasis identitas tambahan. Untuk informasi selengkapnya, lihat Perbedaan peran IAM dari kebijakan berbasis sumber daya di Panduan Pengguna.AWS Identity and Access Management

Layanan Amazon S3 hanya mendukung satu jenis kebijakan berbasis sumber daya yang disebut kebijakan bucket, yang dilampirkan ke bucket. Kebijakan ini menentukan entitas utama mana (akun, pengguna, peran, dan pengguna gabungan) yang dapat melakukan tindakan pada objek.

Contoh-contoh

Untuk melihat contoh kebijakan AWS Transfer Family berbasis sumber daya, lihat. AWS Transfer Family contoh kebijakan berbasis tag

Otorisasi berdasarkan tanda AWS Transfer Family

Anda dapat melampirkan tag ke AWS Transfer Family sumber daya atau meneruskan tag dalam permintaan AWS Transfer Family. Untuk mengendalikan akses berdasarkan tag, berikan informasi tentang tag di elemen kondisi dari kebijakan dengan menggunakan kunci kondisi `transfer:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, atau `aws:TagKeys`. Untuk informasi tentang cara menggunakan tag untuk mengontrol akses ke AWS Transfer Family sumber daya, lihatAWS Transfer Family contoh kebijakan berbasis tag.

AWS Transfer Family Peran IAM

Peran IAM adalah entitas dalam AWS akun Anda yang memiliki izin tertentu.

Menggunakan kredensyal sementara dengan AWS Transfer Family

Anda dapat menggunakan kredensial sementara untuk masuk dengan gabungan, menjalankan IAM role, atau menjalankan peran lintas akun. Anda memperoleh kredensyal keamanan sementara dengan memanggil operasi AWS STS API seperti AssumeRoleatau. GetFederationToken

AWS Transfer Family mendukung menggunakan kredensyal sementara.

AWS Transfer Family contoh kebijakan berbasis identitas

Secara default, pengguna dan peran IAM tidak memiliki izin untuk membuat atau memodifikasi sumber daya AWS Transfer Family . Mereka juga tidak dapat melakukan tugas menggunakan AWS Management Console, AWS CLI, atau AWS API. Administrator IAM harus membuat kebijakan IAM yang memberikan izin kepada pengguna dan peran untuk melakukan operasi API tertentu pada sumber daya yang diperlukan. Administrator kemudian harus melampirkan kebijakan tersebut ke pengguna IAM atau grup yang memerlukan izin tersebut.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM menggunakan contoh dokumen kebijakan JSON ini, lihat Membuat kebijakan pada tab JSON di Panduan Pengguna.AWS Identity and Access Management

Topik

- Praktik terbaik kebijakan
- Menggunakan konsol AWS Transfer Family
- Izinkan para pengguna untuk melihat izin mereka sendiri

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus AWS Transfer Family sumber daya di akun Anda. Tindakan ini dikenai biaya untuk Akun AWS Anda. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola

yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat kebijakan yang dikelola AWS atau kebijakan yang dikelola AWS untuk fungsi pekerjaan di Panduan Pengguna IAM.

- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukan ini dengan menentukan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, juga dikenal sebagai izin hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk menerapkan izin, lihat Kebijakan dan izin di IAM di Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti AWS CloudFormation. Untuk informasi selengkapnya, lihat Elemen kebijakan JSON IAM: Syarat di Panduan Pengguna IAM.
- Menggunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda guna memastikan izin yang aman dan berfungsi – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat validasi kebijakan Analizer Akses IAM di Panduan Pengguna IAM.
- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk mewajibkan MFA saat operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat Mengonfigurasi akses API yang dilindungi MFA di Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat Praktik terbaik keamanan di IAM di Panduan Pengguna IAM.

Menggunakan konsol AWS Transfer Family

Untuk mengakses AWS Transfer Family konsol, Anda harus memiliki set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang AWS Transfer Family sumber daya di AWS akun Anda. Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tersebut tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna IAM atau peran) dengan kebijakan tersebut. Untuk informasi selengkapnya, lihat Menambahkan izin ke pengguna di Panduan AWS Identity and Access Management Pengguna.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai alternatif, hanya izinkan akses ke tindakan yang cocok dengan operasi API yang sedang Anda coba lakukan.

Izinkan para pengguna untuk melihat izin mereka sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan para pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. AWS CLI AWS

```
{
    "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 contoh kebijakan berbasis tag

Berikut ini adalah contoh cara mengontrol akses ke AWS Transfer Family sumber daya berdasarkan tag.

Menggunakan tag untuk mengontrol akses ke AWS Transfer Family sumber daya

Ketentuan dalam kebijakan IAM adalah bagian dari sintaks yang Anda gunakan untuk menentukan izin ke sumber daya. AWS Transfer Family Anda dapat mengontrol akses ke AWS Transfer Family sumber daya (seperti pengguna, server, peran, dan entitas lain) berdasarkan tag pada sumber daya tersebut. Tag adalah pasangan nilai kunci. Untuk informasi selengkapnya tentang menandai sumber daya, lihat Menandai AWS sumber daya di. Referensi Umum AWS

Di AWS Transfer Family, sumber daya dapat memiliki tag, dan beberapa tindakan dapat menyertakan tag. Saat membuat kebijakan IAM, Anda dapat menggunakan kunci syarat tanda berikut:

- Pengguna mana yang dapat melakukan tindakan pada AWS Transfer Family sumber daya, berdasarkan tag yang dimiliki sumber daya.
- Tag apa yang dapat diteruskan dalam permintaan tindakan.
- Apakah kunci tag tertentu dapat digunakan dalam permintaan.

Dengan menggunakan kontrol akses berbasis tag, Anda dapat menerapkan kontrol yang lebih baik daripada di API level. Anda juga dapat menerapkan kontrol yang lebih dinamis daripada dengan menggunakan kontrol akses berbasis sumber daya. Anda dapat membuat kebijakan IAM yang mengizinkan atau menolak operasi berdasarkan tag yang disediakan dalam permintaan (tag permintaan). Anda juga dapat membuat kebijakan IAM berdasarkan tag pada sumber daya yang sedang dioperasikan (tag sumber daya). Secara umum, tag sumber daya adalah untuk tag yang

sudah ada di sumber daya, tag permintaan digunakan saat Anda menambahkan tag atau menghapus tag dari sumber daya.

Untuk sintaks dan semantik lengkap kunci kondisi tag, lihat Mengontrol akses ke AWS sumber daya menggunakan tag sumber daya di Panduan Pengguna IAM. Untuk detail tentang menentukan kebijakan IAM dengan API Gateway, lihat Mengontrol akses ke API dengan izin IAM di Panduan Pengembang API Gateway.

Contoh 1: Tolak tindakan berdasarkan tag sumber daya

Anda dapat menolak tindakan yang akan dilakukan pada sumber daya berdasarkan tag. Contoh kebijakan berikut menyangkal`TagResource`,`UntagResource`,`StartServer`,`StopServer`,`DescribeServer`, dan `DescribeUser` operasi jika sumber daya pengguna atau server ditandai dengan kunci `stage` dan nilainya. `prod`

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

Contoh 2: Izinkan tindakan berdasarkan tag sumber daya

Anda dapat mengizinkan tindakan dilakukan pada sumber daya berdasarkan tag. Contoh kebijakan berikut memungkinkan`TagResource`,`UntagResource`,`StartServer`,`StopServer`,`DescribeServer`, dan `DescribeUser` operasi jika sumber daya pengguna atau server ditandai dengan kunci `stage` dan nilainya`prod`.

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

Contoh 3: Tolak pembuatan pengguna atau server berdasarkan tag permintaan

Contoh kebijakan berikut berisi dua pernyataan. Pernyataan pertama menyangkal `CreateServer` operasi pada semua sumber daya jika kunci pusat biaya untuk tag tidak memiliki nilai.

Pernyataan kedua menyangkal `CreateServer` operasi jika kunci pusat biaya untuk tag berisi nilai lain selain 1, 2 atau 3.

> **Note**
>
> Kebijakan ini memungkinkan pembuatan atau penghapusan sumber daya yang berisi kunci yang dipanggil `costcenter` dan nilai1,2, atau3.

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

Memecahkan masalah AWS Transfer Family identitas dan akses

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan AWS Transfer Family dan IAM.

Topik

- Saya tidak berwenang untuk melakukan tindakan di AWS Transfer Family
- Saya tidak berwenang untuk melakukan iam: PassRole
- Saya ingin mengizinkan orang di luar AWS akun saya untuk mengakses AWS Transfer Family sumber daya saya

Saya tidak berwenang untuk melakukan tindakan di AWS Transfer Family

Jika AWS Management Console memberitahu Anda bahwa Anda tidak berwenang untuk melakukan tindakan, maka Anda harus menghubungi administrator Anda untuk bantuan. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Contoh kesalahan berikut terjadi ketika pengguna IAM `mateojackson` mencoba menggunakan konsol untuk melihat detail tentang suatu *`widget`*, tetapi tidak memiliki izin `transfer:`*`GetWidget`*.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 transfer:GetWidget on resource: my-example-widget
```

Dalam hal ini, Mateo meminta administratornya untuk memperbarui kebijakannya agar dia dapat mengakses *`my-example-widget`* menggunakan `transfer;:`*`GetWidget`* tindakan.

Saya tidak berwenang untuk melakukan iam: PassRole

Jika Anda menerima kesalahan yang tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran AWS Transfer Family.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol tersebut untuk melakukan tindakan di AWS Transfer Family. Namun, tindakan tersebut

memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
 iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Kebijakan contoh berikut berisi izin untuk meneruskan peran AWS Transfer Family.

```
{
   "Version": "2012-10-17",
   "Statement": [
      { "Action": "iam:PassRole",
        "Resource": "arn:aws::iam::123456789012:role/*",
        "Effect": "Allow"
      }
   ]
}
```

Saya ingin mengizinkan orang di luar AWS akun saya untuk mengakses AWS Transfer Family sumber daya saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau pengguna di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACL), Anda dapat menggunakan kebijakan tersebut untuk memberi pengguna akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa hal berikut:

- Untuk mempelajari apakah AWS Transfer Family mendukung fitur-fitur ini, lihatBagaimana AWS Transfer Family bekerja dengan IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat Menyediakan akses ke pengguna IAM di pengguna lain Akun AWS yang Anda miliki di Panduan Pengguna IAM.

- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat Memberikan akses kepada pengguna eksternal yang sah (federasi identitas) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara penggunaan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat Perbedaan antara peran IAM dan kebijakan berbasis sumber daya di Panduan Pengguna IAM.

Validasi kepatuhan untuk AWS Transfer Family

Auditor pihak ketiga menilai keamanan dan kepatuhan AWS Transfer Family sebagai bagian dari beberapa program AWS kepatuhan. Hal ini mencakup SOC, PCI, HIPAA, dan lainnya. Untuk daftar lengkapnya, lihat AWS Layanan dalam Lingkup menurut Program Kepatuhan.

Untuk daftar AWS layanan dalam lingkup program kepatuhan tertentu, lihat AWS layanan dalam lingkup oleh program kepatuhan. Untuk informasi umum, lihat Program kepatuhan AWS.

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat Mengunduh laporan di AWS Artifact.

Tanggung jawab kepatuhan Anda saat menggunakan AWS Transfer Family ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- Panduan memulai cepat keamanan dan kepatuhan — Panduan penerapan ini membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar yang berfokus pada keamanan dan kepatuhan. AWS
- Arsitektur untuk whitepaper keamanan dan kepatuhan HIPAA - Whitepaper ini menjelaskan bagaimana perusahaan dapat menggunakan untuk membuat aplikasi yang sesuai dengan HIPAA. AWS
- Sumber daya kepatuhan AWS – Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- AWS Config AWS Layanan ini menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.

- AWS Security Hub— AWS Layanan ini memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS yang membantu Anda memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik.

Ketahanan di AWS Transfer Family

Infrastruktur AWS global dibangun di sekitar AWS Wilayah dan Zona Ketersediaan. AWS Wilayah menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang dan mengoperasikan aplikasi dan basis data yang melakukan secara otomatis pinda saat gagal/failover di antara zona-zona tanpa terputus. Zona Ketersediaan lebih sangat tersedia, lebih toleran kesalahan, dan lebih dapat diskalakan daripadai infrastruktur pusat data tunggal atau multi tradisional.

AWS Transfer Family mendukung hingga 3 Availability Zones dan didukung oleh penskalaan otomatis, armada redundan untuk permintaan koneksi dan transfer Anda.

Perhatikan hal berikut:

- Untuk titik akhir publik:
  - Ketersediaan Redundansi tingkat zona dibangun ke dalam layanan
  - Ada armada redundan untuk setiap AZ.
  - Redundansi ini disediakan secara otomatis
- Untuk titik akhir di Virtual Private Cloud (VPC), lihat. Buat server di cloud pribadi virtual

Lihat juga

- Untuk informasi selengkapnya tentang Wilayah AWS dan Availability Zone, lihat infrastruktur AWS global.
- Untuk contoh tentang cara membangun redundansi yang lebih tinggi dan meminimalkan latensi jaringan dengan menggunakan perutean berbasis Latensi, lihat posting blog Minimalkan latensi jaringan dengan server Anda. AWS Transfer Family

Keamanan infrastruktur di AWS Transfer Family

Sebagai layanan terkelola, AWS Transfer Family dilindungi oleh keamanan jaringan AWS global. Untuk informasi tentang layanan AWS keamanan dan cara AWS melindungi infrastruktur, lihat Keamanan AWS Cloud. Untuk mendesain AWS lingkungan Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat Perlindungan Infrastruktur dalam Kerangka Kerja yang AWS Diarsiteksikan dengan Baik Pilar Keamanan.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses AWS Transfer Family melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Pengangkutan (TLS). Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Sandi cocok dengan sistem kerahasiaan maju sempurna (perfect forward secrecy, PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

Selain itu, permintaan harus ditandatangani dengan menggunakan ID kunci akses dan kunci akses rahasia yang terkait dengan pengguna utama IAM. Atau Anda dapat menggunakan AWS Security Token Service (AWS STS) untuk menghasilkan kredensial keamanan sementara untuk menandatangani permintaan.

Tambahkan firewall aplikasi web

AWS WAF adalah firewall aplikasi web yang membantu melindungi aplikasi web dan API dari serangan. Anda dapat menggunakannya untuk mengonfigurasi seperangkat aturan yang dikenal sebagai daftar kontrol akses web (web ACL) yang mengizinkan, memblokir, atau menghitung permintaan web berdasarkan aturan dan kondisi keamanan web yang dapat disesuaikan yang Anda tentukan. Untuk informasi selengkapnya, lihat Menggunakan AWS WAF untuk melindungi API Anda.

Untuk menambahkan AWS WAF

1. Buka konsol API Gateway di https://console.aws.amazon.com/apigateway/.
2. Di panel navigasi API, lalu pilih templat penyedia identitas kustom Anda.
3. Memilih Tahapan.
4. Di panel Tahapan, pilih nama panggung.
5. Di panel Editor Panggung, pilih Pengaturan tab.

6. Lakukan salah satu hal berikut:

   - Di bawah Web Application Firewall (WAF), untuk Web ACL, pilih ACL web yang ingin Anda kaitkan dengan tahap ini.
   - Jika ACL web yang Anda butuhkan tidak ada, Anda harus membuatnya dengan melakukan hal berikut:

     1. Pilih Buat Web ACL.
     2. Di beranda layanan AWS WAF, pilih Buat web ACL.
     3. Dalam detail Web ACL, untuk Nama, ketikkan nama ACL web.
     4. Di Aturan, pilih Tambahkan aturan, lalu pilih Tambahkan aturan dan grup aturan saya sendiri.
     5. Untuk tipe Rule, pilih IP set untuk mengidentifikasi daftar alamat IP tertentu.
     6. Untuk Aturan, masukkan nama aturan.
     7. Untuk set IP, pilih set IP yang ada. Untuk membuat kumpulan IP, lihat Membuat set IP.
     8. Untuk alamat IP yang akan digunakan sebagai alamat asal, pilih alamat IP di header.
     9. Untuk nama bidang Header, masukkan`SourceIP`.
     10. Untuk Posisi di dalam header, pilih Alamat IP Pertama.
     11. Untuk Fallback untuk alamat IP yang hilang, pilih Cocokkan atau Tidak Cocokkan tergantung pada bagaimana Anda ingin menangani alamat IP yang tidak valid (atau hilang) di header.
     12. Untuk Tindakan, pilih tindakan set IP.
     13. Untuk tindakan ACL web default untuk permintaan yang tidak cocok dengan aturan apa pun, pilih Izinkan atau Blokir, lalu klik Berikutnya.
     14. Untuk langkah 4 dan 5, pilih Berikutnya.
     15. Di Tinjau dan buat, tinjau pilihan Anda, lalu pilih Buat ACL web.
7. Pilih Simpan Perubahan.
8. Pilih Sumber daya.
9. Untuk Tindakan, pilih Deploy API.

Untuk informasi tentang seberapa aman AWS Transfer Family dengan firewall aplikasi AWS web, lihat Mengamankan AWS Transfer Family dengan firewall AWS aplikasi dan Amazon API Gateway di blog AWS penyimpanan.

Pencegahan confused deputy lintas layanan

Masalah deputi yang bingung adalah masalah keamanan di mana entitas yang tidak memiliki izin untuk melakukan tindakan dapat memaksa entitas yang lebih istimewa untuk melakukan tindakan. Pada tahun AWS, peniruan lintas layanan dapat mengakibatkan masalah wakil yang membingungkan. Peniruan identitas lintas layanan dapat terjadi ketika satu layanan (layanan yang dipanggil) memanggil layanan lain (layanan yang dipanggil). Layanan panggilan dapat dimanipulasi untuk menggunakan izinnya untuk bertindak atas sumber daya pelanggan lain dengan cara yang seharusnya tidak memiliki izin untuk mengakses. Untuk mencegah hal ini, AWS menyediakan alat yang membantu Anda melindungi data untuk semua layanan dengan pengguna utama layanan yang telah diberi akses ke sumber daya di akun Anda. Untuk penjelasan rinci tentang masalah ini, lihat masalah wakil yang membingungkan di Panduan Pengguna IAM.

Sebaiknya gunakan kunci konteks kondisi `aws:SourceAccount`global `aws:SourceArn`dan global dalam kebijakan sumber daya untuk membatasi izin yang dimiliki AWS Transfer Family untuk sumber daya. Jika Anda menggunakan kedua kunci konteks kondisi global, `aws:SourceAccount` nilai dan akun dalam `aws:SourceArn` nilai harus menggunakan ID akun yang sama saat digunakan dalam pernyataan kebijakan yang sama.

Cara paling efektif untuk melindungi dari masalah wakil yang membingungkan adalah dengan menggunakan Nama Sumber Daya Amazon (ARN) yang tepat dari sumber daya yang ingin Anda izinkan. Jika Anda menentukan beberapa sumber daya, gunakan kunci kondisi konteks `aws:SourceArn` global dengan karakter wildcard (`*`) untuk bagian ARN yang tidak diketahui. Misalnya, `arn:aws:transfer::`*`region`*`::`*`account-id`*`:server/*`.

AWS Transfer Family menggunakan jenis peran berikut:

- Peran pengguna — Memungkinkan pengguna yang dikelola layanan mengakses sumber daya Transfer Family yang diperlukan. AWS Transfer Family mengambil peran ini dalam konteks ARN pengguna Transfer Family.
- Peran akses - Menyediakan akses hanya ke file Amazon S3 yang sedang ditransfer. Untuk transfer AS2 masuk, peran akses menggunakan Amazon Resource Name (ARN) untuk perjanjian. Untuk transfer AS2 keluar, peran akses menggunakan ARN untuk konektor.
- Peran pemanggilan — Untuk digunakan dengan Amazon API Gateway sebagai penyedia identitas kustom server. Transfer Family mengasumsikan peran ini dalam konteks ARN server Transfer Family.
- Peran logging - Digunakan untuk log entri ke Amazon CloudWatch. Transfer Family menggunakan peran ini untuk mencatat detail keberhasilan dan kegagalan bersama dengan informasi tentang

transfer file. Transfer Family mengasumsikan peran ini dalam konteks ARN server Transfer Family. Untuk transfer AS2 keluar, peran logging menggunakan konektor ARN.

- Peran eksekusi — Memungkinkan pengguna Transfer Family memanggil dan meluncurkan alur kerja. Transfer Family mengasumsikan peran ini dalam konteks alur kerja Transfer Family ARN.

Untuk informasi selengkapnya, lihat Kebijakan dan izin di IAM dalam Panduan Pengguna IAM.

> **Note**
>
> Dalam contoh-contoh berikut, ganti setiap *placeholder input pengguna* dengan informasi Anda sendiri.

> **Note**
>
> Dalam contoh kami, kami menggunakan keduanya `ArnLike` dan`ArnEquals`. Mereka identik secara fungsional, dan oleh karena itu Anda dapat menggunakan keduanya ketika Anda membuat kebijakan Anda. Dokumentasi Transfer Family digunakan `ArnLike` ketika kondisi berisi karakter wildcard, dan `ArnEquals` untuk menunjukkan kondisi kecocokan yang tepat.

AWS Transfer Family peran pengguna lintas layanan pencegahan wakil yang membingungkan

Contoh kebijakan berikut memungkinkan setiap pengguna dari server mana pun di akun untuk mengambil peran.

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

Contoh kebijakan berikut memungkinkan setiap pengguna dari server tertentu untuk mengambil peran.

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

Contoh kebijakan berikut memungkinkan pengguna tertentu dari server tertentu untuk mengambil peran.

```
{
    "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 Alur kerja Transfer Family peran lintas layanan membingungkan pencegahan wakil

Contoh kebijakan berikut memungkinkan alur kerja apa pun di akun untuk mengambil peran.

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

```
}
```

Contoh kebijakan berikut memungkinkan alur kerja tertentu untuk mengambil peran.

```
{
    "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 logging dan peran doa lintas layanan membingungkan deputi pencegahan

> **Note**
>
> Contoh berikut dapat digunakan dalam peran logging dan pemanggilan.
> Dalam contoh ini, Anda dapat menghapus detail ARN untuk alur kerja jika server Anda tidak memiliki alur kerja yang melekat padanya.

Contoh kebijakan logging/pemanggilan berikut memungkinkan server apa pun (dan alur kerja) di akun untuk mengambil peran.

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

Contoh kebijakan logging/pemanggilan berikut memungkinkan server tertentu (dan alur kerja) untuk mengambil peran.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "AllowSpecificServerWithWorkflowAttached",
            "Effect": "Allow",
            "Principal": {
                "Service": "transfer.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
            "Condition": {
                "StringEquals": {
                    "aws:SourceAccount": "account-id"
                },
                "ArnEquals": {
                   "aws:SourceArn": [
                      "arn:aws:transfer:region:account-id:server/server-id",
                      "arn:aws:transfer:region:account-id:workflow/workflow-id"
```

```
                    ]
                }
            }
        }
    ]
}
```

AWS kebijakan terkelola untuk AWS Transfer Family

Untuk menambahkan izin ke pengguna, grup, dan peran, lebih mudah menggunakan kebijakan AWS terkelola daripada menulis kebijakan sendiri. Dibutuhkan waktu dan keahlian untuk membuat kebijakan terkelola pelanggan AWS Identity and Access Management (IAM) yang hanya memberi tim Anda izin yang mereka butuhkan. Untuk memulai dengan cepat, Anda dapat menggunakan kebijakan AWS terkelola kami. Kebijakan ini mencakup kasus penggunaan umum dan tersedia di Akun AWS Anda. Untuk informasi lebih lanjut tentang kebijakan yang dikelola AWS , lihat kebijakan yang dikelola AWS di Panduan Pengguna IAM. Untuk daftar detail semua kebijakan AWS terkelola, lihat panduan referensi kebijakan AWS terkelola.

AWS layanan memelihara dan memperbarui kebijakan AWS terkelola. Anda tidak dapat mengubah izin dalam kebijakan AWS terkelola. Layanan terkadang menambahkan izin tambahan ke kebijakan yang dikelola AWS untuk mendukung fitur-fitur baru. Jenis pembaruan ini akan memengaruhi semua identitas (pengguna, grup, dan peran) di mana kebijakan tersebut dilampirkan. Layanan kemungkinan besar akan memperbarui kebijakan yang dikelola AWS saat ada fitur baru yang diluncurkan atau saat ada operasi baru yang tersedia. Layanan tidak menghapus izin dari kebijakan AWS terkelola, sehingga pembaruan kebijakan tidak akan merusak izin yang ada.

Selain itu, AWS mendukung kebijakan terkelola untuk fungsi pekerjaan yang mencakup beberapa layanan. Misalnya, kebijakan `ReadOnlyAccess` AWS terkelola menyediakan akses hanya-baca ke semua AWS layanan dan sumber daya. Saat layanan meluncurkan fitur baru, AWS tambahkan izin hanya-baca untuk operasi dan sumber daya baru. Untuk melihat daftar dan deskripsi dari kebijakan fungsi tugas, lihat kebijakan yang dikelola AWS untuk fungsi tugas di Panduan Pengguna IAM.

AWS kebijakan terkelola: AWSTransferConsoleFullAccess

`AWSTransferConsoleFullAccess`Kebijakan ini menyediakan akses penuh ke Transfer Family melalui Konsol AWS Manajemen.

Detail izin

Kebijakan ini mencakup izin berikut.

- `acm:ListCertificates`— Memberikan izin untuk mengambil daftar sertifikat Amazon Resource Names (ARN) dan nama domain untuk setiap ARN.
- `ec2:DescribeAddresses`— Memberikan izin untuk menjelaskan satu atau lebih alamat IP Elastis.
- `ec2:DescribeAvailabilityZones`— Memberikan izin untuk menjelaskan satu atau beberapa Availability Zone yang tersedia untuk Anda.
- `ec2:DescribeNetworkInterfaces`Memberikan izin untuk mendeskripsikan satu atau lebih antarmuka jaringan elastis.
- `ec2:DescribeSecurityGroups`Memberikan izin untuk mendeskripsikan satu atau lebih kelompok keamanan.
- `ec2:DescribeSubnets`Memberikan izin untuk mendeskripsikan satu atau lebih subnet.
- `ec2:DescribeVpcs`Memberikan izin untuk mendeskripsikan satu atau lebih virtual private cloud (VPC)
- `ec2:DescribeVpcEndpoints`— Memberikan izin untuk menjelaskan satu atau lebih titik akhir VPC.
- `health:DescribeEventAggregates`— Mengembalikan jumlah acara dari setiap jenis acara (masalah, perubahan terjadwal, dan pemberitahuan akun).
- `iam:GetPolicyVersion`Memberikan izin untuk mengambil informasi tentang versi kebijakan terkelola yang ditentukan, termasuk dokumen kebijakan.
- `iam:ListPolicies`— Memberikan izin untuk membuat daftar semua kebijakan yang dikelola.
- `iam:ListRoles`— Memberikan izin untuk membuat daftar peran IAM yang memiliki awalan jalur yang ditentukan.
- `iam:PassRole`— Memberikan izin untuk lulus peran IAM ke Transfer Family. Untuk detail selengkapnya, lihat Memberikan izin pengguna untuk meneruskan peran ke peran. Layanan AWS
- `route53:ListHostedZones`— Memberikan izin untuk mendapatkan daftar zona host publik dan pribadi yang terkait dengan saat ini Akun AWS.
- `s3:ListAllMyBuckets`— Memberikan izin untuk membuat daftar semua ember yang dimiliki oleh pengirim permintaan yang diautentikasi.
- `transfer:*`— Memberikan akses ke sumber daya Transfer Family. Tanda bintang (*) memberikan akses ke semua sumber daya.

```
{
```

```
    "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 kebijakan terkelola: AWSTransferFullAccess

`AWSTransferFullAccess`Kebijakan ini menyediakan akses penuh ke layanan Transfer Family.

Detail izin

Kebijakan ini mencakup izin berikut.

- `transfer:*`— Memberikan izin untuk mengakses sumber daya Transfer Family. Tanda bintang (*) memberikan akses ke semua sumber daya.
- `iam:PassRole`— Memberikan izin untuk lulus peran IAM ke Transfer Family. Untuk detail selengkapnya, lihat Memberikan izin pengguna untuk meneruskan peran ke peran. Layanan AWS
- `ec2:DescribeAddresses`— Memberikan izin untuk menjelaskan satu atau lebih alamat IP Elastis.
- `ec2:DescribeNetworkInterfaces`Memberikan izin untuk mendeskripsikan satu atau lebih antarmuka jaringan.
- `ec2:DescribeVpcEndpoints`— Memberikan izin untuk menjelaskan satu atau lebih titik akhir VPC.

```
{
    "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 kebijakan terkelola: AWSTransferLoggingAccess

`AWSTransferLoggingAccess`Kebijakan ini memberikan akses penuh kepada AWS Transfer Family untuk membuat aliran log dan grup serta memasukkan peristiwa log ke akun Anda.

Detail izin

Kebijakan ini mencakup izin berikut untuk Amazon CloudWatch Logs.

- `CreateLogStream`— Memberikan izin bagi kepala sekolah untuk membuat aliran log.
- `DescribeLogStreams`— Memberikan izin bagi kepala sekolah untuk membuat daftar aliran log untuk grup log.
- `CreateLogGroup`— Memberikan izin bagi kepala sekolah untuk membuat grup log.
- `PutLogEvents`— Memberikan izin bagi kepala sekolah untuk mengunggah sekumpulan peristiwa log ke aliran log.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "logs:CreateLogStream",
                "logs:DescribeLogStreams",
                "logs:CreateLogGroup",
                "logs:PutLogEvents"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS kebijakan terkelola: AWSTransferReadOnlyAccess

`AWSTransferReadOnlyAccess`Kebijakan ini menyediakan akses hanya-baca ke layanan Transfer Family.

Detail izin

Kebijakan ini mencakup izin berikut untuk Transfer Family.

- `DescribeUser`— Memberikan izin bagi kepala sekolah untuk melihat deskripsi bagi pengguna.
- `DescribeServer`— Memberikan izin bagi kepala sekolah untuk melihat deskripsi untuk server.
- `ListUsers`— Memberikan izin bagi kepala sekolah untuk mencantumkan pengguna untuk server.
- `ListServers`— Memberikan izin bagi kepala sekolah untuk membuat daftar server untuk akun tersebut.
- `TestIdentityProvider`— Memberikan izin kepada kepala sekolah untuk menguji apakah penyedia identitas yang dikonfigurasi sudah diatur dengan benar.
- `ListTagsForResource`— Memberikan izin bagi kepala sekolah untuk membuat daftar tag untuk sumber daya.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "transfer:DescribeUser",
                "transfer:DescribeServer",
                "transfer:ListUsers",
                "transfer:ListServers",
                "transfer:TestIdentityProvider",
                "transfer:ListTagsForResource"
            ],
            "Resource": "*"
        }
    ]
}
```

AWS Transfer Family memperbarui kebijakan AWS terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola untuk AWS Transfer Family sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan ke umpan RSS pada halaman Riwayat dokumen untuk AWS Transfer Family.

Perubahan	Deskripsi	Tanggal
Pembaruan dokumentasi	Menambahkan bagian untuk setiap kebijakan yang dikelola Transfer Family.	27 Januari 2022
AWSTransferReadOnlyAccess – Pembaruan ke kebijakan yang ada	AWS Transfer Family menambahkan izin baru agar kebijakan dapat dibaca AWS Managed Microsoft AD.	30 September 2021
AWS Transfer Family mulai melacak perubahan	AWS Transfer Family mulai melacak perubahan untuk kebijakan yang AWS dikelola.	15 Juni 2021

Pemecahan masalah AWS Transfer Family

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengannya AWS Transfer Family.

Untuk masalah dengan IAM di Transfer Family, lihatMemecahkan masalah AWS Transfer Family identitas dan akses.

Topik

- Memecahkan masalah pengguna yang dikelola layanan
- Memecahkan masalah Amazon API Gateway
- Memecahkan masalah kebijakan untuk bucket Amazon S3 terenkripsi
- Memecahkan masalah otentikasi
- Memecahkan masalah alur kerja terkelola
- Memecahkan masalah dekripsi alur kerja
- Memecahkan masalah Amazon EFS
- Memecahkan masalah pengujian penyedia identitas Anda
- Memecahkan masalah menambahkan kunci host tepercaya untuk konektor SFTP Anda
- Memecahkan masalah pengunggahan file
- Memecahkan masalah pengecualian ResourceNotFound
- Memecahkan masalah konektor SFTP
- Memecahkan masalah AS2

Memecahkan masalah pengguna yang dikelola layanan

Bagian ini menjelaskan kemungkinan solusi untuk masalah berikut.

Topik

- Memecahkan masalah pengguna yang dikelola layanan Amazon EFS
- Memecahkan masalah badan kunci publik terlalu lama
- Pemecahan masalah gagal menambahkan kunci publik SSH

Memecahkan masalah pengguna yang dikelola layanan Amazon EFS

Deskripsi

Anda menjalankan `sftp` perintah dan prompt tidak muncul, dan sebagai gantinya Anda melihat pesan berikut:

```
Couldn't canonicalize: Permission denied
           Need cwd
```

Menyebabkan

Peran pengguna AWS Identity and Access Management (IAM) Anda tidak memiliki izin untuk mengakses Amazon Elastic File System (Amazon EFS).

Solusi

Tingkatkan izin kebijakan untuk peran pengguna Anda. Anda dapat menambahkan kebijakan AWS terkelola, seperti`AmazonElasticFileSystemClientFullAccess`.

Memecahkan masalah badan kunci publik terlalu lama

Deskripsi

Saat Anda mencoba membuat pengguna yang dikelola layanan, Anda menerima kesalahan berikut:

```
Failed to create user (1 validation error detected:
'sshPublicKeyBody' failed to satisfy constraint: Member must have length less than or
 equal to 2048)
```

Menyebabkan

Anda mungkin memasukkan kunci PGP untuk badan kunci publik, dan AWS Transfer Family tidak mendukung kunci PGP untuk pengguna yang dikelola layanan.

Solusi

Jika kunci PGP berbasis RSA, Anda dapat mengonversinya ke format PEM. Misalnya, Ubuntu menyediakan alat konversi di sini: https://manpages.ubuntu.com/manpages/xenial/man1/openpgp2ssh.1.html

Pemecahan masalah gagal menambahkan kunci publik SSH

Deskripsi

Saat Anda mencoba menambahkan kunci publik untuk pengguna yang dikelola layanan, Anda menerima kesalahan berikut:

```
Failed to add SSH public key (Unsupported or invalid SSH public key format)
```

Menyebabkan

Anda mungkin mencoba mengimpor kunci publik berformat SSH2, dan AWS Transfer Family tidak mendukung kunci publik berformat SSH2 untuk pengguna yang dikelola layanan.

Solusi

Anda perlu mengonversi kunci ke dalam format OpenSSH. Proses ini dijelaskan di Mengkonversi kunci publik SSH2 ke format PEM.

Memecahkan masalah Amazon API Gateway

Bagian ini menjelaskan kemungkinan solusi untuk masalah API Gateway berikut.

Topik

- Terlalu banyak kegagalan otentikasi
- Koneksi ditutup

Terlalu banyak kegagalan otentikasi

Deskripsi

Ketika Anda mencoba untuk terhubung ke server Anda menggunakan Secure Shell (SSH) File Transfer Protocol (SFTP), Anda mendapatkan kesalahan berikut:

```
Received disconnect from 3.15.127.197 port 22:2: Too many authentication failures
Authentication failed.
Couldn't read packet: Connection reset by peer
```

Menyebabkan

Anda mungkin telah memasukkan kata sandi yang salah untuk pengguna Anda. Coba lagi untuk memasukkan kata sandi yang benar.

Jika kata sandi sudah benar, masalah mungkin disebabkan oleh peran Amazon Resource Name (ARN) yang tidak valid. Untuk mengonfirmasi bahwa ini masalahnya, uji penyedia identitas untuk server Anda. Jika Anda melihat respons yang mirip dengan berikut ini, ARN peran hanya merupakan placeholder, seperti yang ditunjukkan oleh nilai ID peran dari semua nol:

```
{
    "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"
}
```

Solusi

Ganti peran placeholder ARN dengan peran aktual yang memiliki izin untuk mengakses server.

Untuk memperbarui peran

1. Buka AWS CloudFormation konsol di https://console.aws.amazon.com/cloudformation.
2. Di panel navigasi sebelah kiri, pilih Tumpukan.
3. Dalam daftar Stacks, pilih tumpukan Anda, lalu pilih tab Parameter.
4. Pilih Perbarui. Pada halaman Update stack, pilih Use current template, lalu pilih Next.
5. Ganti UserRoleArndengan ARN peran yang memiliki izin yang cukup untuk mengakses server Transfer Family Anda.

> **Note**
>
> Untuk memberikan izin yang diperlukan, Anda dapat menambahkan `AmazonAPIGatewayAdministrator` dan kebijakan `AmazonS3FullAccess` terkelola ke peran Anda.

6. Pilih Berikutnya, lalu pilih Berikutnya lagi. Pada halaman ***tumpukan*** Tinjauan, pilih Saya mengakui yang AWS CloudFormation mungkin membuat sumber daya IAM, lalu pilih Perbarui tumpukan.

Koneksi ditutup

Deskripsi

Ketika Anda mencoba untuk terhubung ke server Anda menggunakan Secure Shell (SSH) File Transfer Protocol (SFTP), Anda mendapatkan kesalahan berikut:

```
Connection closed
```

Menyebabkan

Salah satu kemungkinan penyebab masalah ini adalah bahwa peran CloudWatch pencatatan Amazon Anda tidak memiliki hubungan kepercayaan dengan Transfer Family.

Solusi

Pastikan bahwa peran logging untuk server memiliki hubungan kepercayaan dengan Transfer Family. Untuk informasi selengkapnya, lihat Untuk membangun hubungan kepercayaan.

Memecahkan masalah kebijakan untuk bucket Amazon S3 terenkripsi

Deskripsi

Anda memiliki bucket Amazon S3 terenkripsi yang Anda gunakan sebagai penyimpanan untuk server Transfer Family Anda. Jika Anda mencoba mengunggah file ke server, Anda menerima kesalahan`Couldn't close file: Permission denied`.

Dan jika Anda melihat log server, Anda melihat kesalahan berikut:

```
ERROR Message="Access denied" Operation=CLOSE Path=/bucket/user/test.txt BytesIn=13
ERROR Message="Access denied"
```

Menyebabkan

Kebijakan untuk pengguna IAM Anda tidak memiliki izin untuk mengakses bucket terenkripsi.

Solusi

Anda harus menentukan izin tambahan dalam kebijakan Anda untuk memberikan izin required AWS Key Management Service (AWS KMS). Untuk detailnya, lihat Enkripsi data di Amazon S3.

Memecahkan masalah otentikasi

Bagian ini menjelaskan kemungkinan solusi untuk masalah otentikasi berikut.

Topik

- Kegagalan otentikasi—SSH/SFTP
- Masalah alam tidak cocok AD terkelola
- Masalah otentikasi lain-lain

Kegagalan otentikasi—SSH/SFTP

Deskripsi

Ketika Anda mencoba untuk terhubung ke server Anda menggunakan Secure Shell (SSH) File Transfer Protocol (SFTP), Anda menerima pesan yang mirip dengan berikut ini:

```
Received disconnect from 3.130.115.105 port 22:2: Too many authentication failures
  Authentication failed.
```

> **Note**
>
> Jika Anda menggunakan API Gateway dan menerima kesalahan ini, lihatTerlalu banyak kegagalan otentikasi.

Menyebabkan

Anda belum menambahkan key pair RSA untuk pengguna Anda, jadi Anda harus mengautentikasi menggunakan kata sandi sebagai gantinya.

Solusi

Saat Anda menjalankan `sftp` perintah, tentukan `-o PubkeyAuthentication=no` opsi. Opsi ini memaksa sistem untuk meminta kata sandi Anda. Sebagai contoh:

```
sftp -o PubkeyAuthentication=no sftp-user@server-id.server.transfer.region-id.amazonaws.com
```

Masalah alam tidak cocok AD terkelola

Deskripsi

Ranah pengguna dan ranah grup mereka harus cocok. Mereka berdua harus berada di ranah default, atau keduanya harus berada di ranah tepercaya.

Menyebabkan

Jika pengguna dan grup mereka tidak cocok, pengguna tidak dapat diautentikasi oleh Transfer Family. Jika Anda menguji penyedia identitas untuk pengguna, Anda menerima kesalahan Tidak ada akses terkait yang ditemukan untuk grup pengguna.

Solusi

Referensikan grup di ranah pengguna yang cocok dengan ranah grup (baik default atau tepercaya).

Masalah otentikasi lain-lain

Deskripsi

Anda menerima kesalahan otentikasi dan tidak ada pemecahan masalah lainnya yang berfungsi

Menyebabkan

Anda mungkin telah menentukan target untuk direktori logis yang berisi garis miring (/).

Solusi

Perbarui target direktori logis Anda, untuk memastikannya dimulai dengan garis miring, dan tidak berisi garis miring. Misalnya, dapat `/DOC-EXAMPLE-BUCKET/images` diterima, tetapi `DOC-EXAMPLE-BUCKET/images` dan `/DOC-EXAMPLE-BUCKET/images/` tidak.

Memecahkan masalah alur kerja terkelola

Bagian ini menjelaskan kemungkinan solusi untuk masalah alur kerja berikut.

Topik

- Memecahkan masalah kesalahan terkait alur kerja menggunakan Amazon CloudWatch
- Memecahkan masalah kesalahan penyalinan alur kerja

Memecahkan masalah kesalahan terkait alur kerja menggunakan Amazon CloudWatch

Deskripsi

Jika Anda mengalami masalah dengan alur kerja Anda, Anda dapat menggunakan Amazon CloudWatch untuk menyelidiki penyebabnya.

Menyebabkan

Mungkin ada beberapa penyebab. Gunakan Amazon CloudWatch Logs untuk menyelidiki.

Solusi

Transfer Family memancarkan status eksekusi alur kerja ke dalam CloudWatch Log. Jenis kesalahan alur kerja berikut dapat muncul di CloudWatch Log:

- `"type": "StepErrored"`
- `"type": "ExecutionErrored"`
- `"type": "ExecutionThrottled"`
- `"Service failure on starting workflow"`

Anda dapat memfilter log eksekusi alur kerja Anda menggunakan sintaks filter dan pola yang berbeda. Misalnya, Anda dapat membuat filter log di CloudWatch log Anda untuk menangkap log eksekusi alur kerja yang berisi ExecutionErroredpesan. Untuk detailnya, lihat Pemrosesan data log secara real-time dengan langganan serta Filter dan sintaks pola di Panduan Pengguna Amazon CloudWatch Logs.

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"}
```

Di sini, `StepErrored` menunjukkan bahwa langkah dalam alur kerja telah menghasilkan kesalahan. Dalam satu alur kerja, Anda dapat memiliki beberapa langkah yang dikonfigurasi. Kesalahan ini memberi tahu Anda di langkah mana kesalahan terjadi dan memberikan pesan kesalahan. Dalam contoh khusus ini, langkah dikonfigurasi untuk menandai file; Namun, menandai file dalam sistem file Amazon EFS tidak didukung, sehingga langkah tersebut menghasilkan kesalahan.

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"}}
```

Ketika alur kerja tidak dapat menjalankan langkah apa pun, itu menghasilkan `ExecutionErrored` pesan. Misalnya, jika Anda telah mengonfigurasi satu langkah dalam alur kerja tertentu, dan jika langkah tersebut tidak dapat dijalankan, alur kerja keseluruhan gagal.

EksekusiTerhambat

Eksekusi dibatasi jika alur kerja dipicu pada tingkat yang lebih cepat daripada yang dapat didukung sistem. Pesan log ini menunjukkan bahwa Anda harus memperlambat laju eksekusi untuk alur kerja. Jika Anda tidak dapat menurunkan tingkat eksekusi alur kerja Anda, hubungi AWS Support di Kontak. AWS

Kegagalan layanan saat memulai alur kerja

Setiap kali Anda menghapus alur kerja dari server dan menggantinya dengan yang baru, atau memperbarui konfigurasi server (yang memengaruhi peran eksekusi alur kerja), Anda harus menunggu sekitar 10 menit sebelum menjalankan alur kerja baru. Server Transfer Family menyimpan cache detail alur kerja, dan dibutuhkan waktu 10 menit bagi server untuk menyegarkan cache-nya.

Selain itu, Anda harus keluar dari sesi SFTP aktif apa pun, dan kemudian masuk kembali setelah masa tunggu 10 menit untuk melihat perubahannya.

Memecahkan masalah kesalahan penyalinan alur kerja

Deskripsi

Jika Anda menjalankan alur kerja yang berisi langkah untuk menyalin file yang diunggah, Anda dapat mengalami kesalahan berikut:

```
{
  "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"
  }
}
```

Menyebabkan

File sumber ada di bucket Amazon S3 yang berbeda Wilayah AWS dari bucket tujuan.

Solusi

Jika Anda menjalankan alur kerja yang menyertakan langkah penyalinan, pastikan bucket sumber dan tujuan berada di tempat yang sama. Wilayah AWS

Memecahkan masalah dekripsi alur kerja

Bagian ini menjelaskan kemungkinan solusi untuk masalah berikut dengan alur kerja terenkripsi.

Topik

- Memecahkan masalah kesalahan untuk file enkripsi yang ditandatangani
- Memecahkan masalah kesalahan untuk algoritma FIPS

Memecahkan masalah kesalahan untuk file enkripsi yang ditandatangani

Deskripsi

Alur kerja dekripsi Anda gagal dan Anda menerima kesalahan berikut:

```
"Encrypted file with signed message unsupported"
```

Menyebabkan

Transfer Family saat ini tidak mendukung penandatanganan untuk file terenkripsi.

Solusi

Di klien PGP Anda, jika ada opsi untuk menandatangani file terenkripsi, pastikan untuk menghapus pilihan, karena Transfer Family saat ini tidak mendukung penandatanganan untuk file terenkripsi.

Memecahkan masalah kesalahan untuk algoritma FIPS

Deskripsi

Alur kerja dekripsi Anda gagal, dan pesan log menyerupai berikut ini:

```
{
   "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"
   }
}
```

Menyebabkan

Server Transfer Family Anda mengaktifkan mode FIPS dan langkah alur kerja Dekripsi terkait. Saat mengenkripsi file sebelum mengunggah ke server Transfer Family Anda, klien enkripsi mungkin menghasilkan file terenkripsi yang menggunakan algoritme enkripsi simetris yang disetujui non-FIPS. Dalam skenario seperti itu, alur kerja tidak dapat mendekripsi file. Dalam contoh berikut, GnuPG versi 2.4.0 menggunakan OCB (mode sandi blok non-FIPS) untuk mengenkripsi file: ini menyebabkan alur kerja gagal.

Solusi

Anda harus mengedit kunci GPG yang Anda gunakan untuk mengenkripsi file Anda, dan kemudian mengenkripsi ulang mereka. Prosedur berikut menjelaskan langkah-langkah yang harus Anda ambil.

Untuk mengedit kunci PGP

1. Identifikasi kunci yang harus Anda edit dengan menjalankan `gpg --list-keys`

   Ini mengembalikan daftar kunci. Setiap kunci memiliki detail yang mirip dengan yang berikut ini:

   ```
   pub   ed25519 2022-07-07 [SC]
         wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
   uid           [ultimate] Mary Major <marymajor@example.com>
   sub   cv25519 2022-07-07 [E]
   ```

2. Identifikasi kunci yang ingin Anda edit. Pada contoh yang ditunjukkan pada langkah sebelumnya, ID adalah`wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`.
3. Jalankan `gpg --edit-key wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`.

   Sistem merespons dengan detail tentang program GnuPG dan kunci yang ditentukan.
4. Pada `gpg>` prompt, masukkan`showpref`. Rincian berikut dikembalikan:

   ```
   [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
   ```

   Perhatikan bahwa algoritma pilihan yang disimpan pada kunci terdaftar.
5. Kami ingin mengedit kunci untuk mempertahankan semua algoritma kecuali untuk OCB. Jalankan `setpref` perintah, tentukan semua algoritma untuk mempertahankan:

```
gpg> setpref AES256, AES192,AES,3DES,SHA512, SHA384, SHA256, SHA224, SHA1,ZLIB,
 BZIP2, ZIP, Uncompressed
```

Ini mengembalikan rincian berikut:

```
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. Masukkan y untuk memperbarui, lalu masukkan kata sandi Anda saat diminta untuk mengonfirmasi perubahan.
7. Simpan perubahan.

```
gpg> save
```

Sebelum menjalankan kembali alur kerja dekripsi Anda, Anda harus mengenkripsi ulang file Anda, menggunakan kunci yang diedit.

Memecahkan masalah Amazon EFS

Bagian ini menjelaskan kemungkinan solusi untuk masalah Amazon EFS berikut.

Topik

- Memecahkan masalah profil POSIX yang hilang
- Memecahkan masalah direktori logis dengan Amazon EFS

Memecahkan masalah profil POSIX yang hilang

Deskripsi

Jika Anda menggunakan penyimpanan Amazon EFS untuk server Anda dan Anda menggunakan penyedia identitas khusus, Anda harus menyediakan AWS Lambda fungsi Anda dengan profil POSIX.

Menyebabkan

Salah satu kemungkinan penyebabnya adalah template yang kami sediakan untuk membuat metode Amazon API Gateway yang AWS Lambda didukung saat ini tidak berisi informasi POSIX.

Jika Anda memang memberikan informasi POSIX, format yang Anda gunakan untuk memberikan informasi POSIX mungkin tidak diuraikan dengan benar oleh Transfer Family.

Solusi

Pastikan bahwa Anda menyediakan elemen JSON untuk Transfer Family untuk `PosixProfile` parameter.

Misalnya, jika Anda menggunakan Python, Anda dapat menambahkan baris berikut di mana Anda mengurai parameter: `PosixProfile`

```
if PosixProfile:
        response_data["PosixProfile"] = json.loads(PosixProfile)
```

Atau, di JavaScript, Anda bisa menambahkan baris berikut, di mana *`uid-value`* dan *`gid-value`* bilangan bulat, 0 atau lebih besar, yang mewakili User ID (UID) dan Group ID (GID) masing-masing:

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

Contoh kode ini mengirim `PosixProfile` parameter ke Transfer Family sebagai objek JSON, bukan sebagai string.

Juga, di dalam AWS Secrets Manager, Anda harus menyimpan `PosixProfile` parameter sebagai berikut. Ganti *`your-uid`* dan *`your-gid`* dengan nilai aktual Anda untuk GID dan UID.

```
{"Uid": your-uid, "Gid": your-gid, "SecondaryGids": []}
```

Memecahkan masalah direktori logis dengan Amazon EFS

Deskripsi

Jika direktori home pengguna tidak ada, dan mereka menjalankan `ls` perintah, sistem merespons sebagai berikut:

```
sftp> ls
```

```
remote readdir ("/"): No such file or directory
```

Menyebabkan

Jika server Transfer Family Anda menggunakan Amazon EFS, direktori home untuk pengguna harus dibuat dengan akses baca dan tulis sebelum pengguna dapat bekerja di direktori home logisnya. Pengguna tidak dapat membuat direktori ini sendiri, karena mereka akan kekurangan izin untuk `mkdir` direktori home logis mereka.

Solusi

Seorang pengguna dengan akses administratif ke direktori induk perlu membuat direktori home logis pengguna.

Memecahkan masalah pengujian penyedia identitas Anda

Deskripsi

Jika Anda menguji penyedia identitas menggunakan konsol atau panggilan `TestIdentityProvider` API, `Response` bidang tersebut kosong. Sebagai contoh:

```
{
    "Response": "{}",
    "StatusCode": 200,
    "Message": ""
}
```

Menyebabkan

Penyebab yang paling mungkin adalah otentikasi gagal karena nama pengguna atau kata sandi yang salah.

Solusi

Pastikan Anda menggunakan kredensyal yang benar untuk pengguna Anda, dan buat pembaruan pada nama pengguna atau kata sandi, jika perlu.

Memecahkan masalah menambahkan kunci host tepercaya untuk konektor SFTP Anda

Deskripsi

Saat Anda membuat atau mengedit konektor SFTP, dan Anda menambahkan kunci host tepercaya, Anda menerima kesalahan berikut: `Failed to edit connector details (Invalid host key format.)`

Menyebabkan

Jika Anda menempelkan kunci publik yang benar, masalahnya mungkin Anda memasukkan `comment` bagian kunci tersebut. AWS Transfer Family saat ini tidak menerima bagian komentar dari kunci.

Solusi

Hapus bagian komentar dari kunci, saat Anda menempelkannya ke bidang teks. Misalnya, anggap kunci Anda terlihat mirip dengan yang berikut ini:

```
ssh-rsa AAAA...== marymajor@dev-dsk-marymajor-1d-c1234567.us-east-1.amazon.com
```

Hapus teks yang mengikuti == karakter dan hanya tempel di bagian tombol hingga dan termasuk==.

```
ssh-rsa AAAA...==
```

Memecahkan masalah pengunggahan file

Bagian ini menjelaskan kemungkinan solusi untuk masalah unggahan file berikut.

Topik

- Memecahkan masalah kesalahan unggahan file Amazon S3
- Memecahkan masalah nama file yang tidak dapat dibaca

Memecahkan masalah kesalahan unggahan file Amazon S3

Deskripsi

Saat Anda mencoba mengunggah file ke penyimpanan Amazon S3 menggunakan Transfer Family, Anda menerima pesan galat berikut AWS : Transfer tidak mendukung penulisan akses acak ke objek S3.

Menyebabkan

Saat Anda menggunakan Amazon S3 untuk penyimpanan server Anda, Transfer Family tidak mendukung beberapa koneksi untuk satu transfer.

Solusi

Jika server Transfer Family Anda menggunakan Amazon S3 untuk penyimpanannya, nonaktifkan opsi apa pun untuk perangkat lunak klien Anda yang menyebutkan menggunakan beberapa koneksi untuk satu transfer.

Memecahkan masalah nama file yang tidak dapat dibaca

Deskripsi

Anda melihat nama file rusak di beberapa file yang Anda unggah. Pengguna terkadang mengalami masalah dengan transfer FTP dan SFTP yang mengacaukan karakter tertentu dalam nama file, seperti umlauts, huruf beraksen, atau skrip tertentu, seperti bahasa Mandarin atau Arab.

Menyebabkan

Meskipun protokol FTP dan SFTP dapat memungkinkan pengkodean karakter nama file dinegosiasikan oleh klien, Amazon S3 dan Amazon EFS tidak. Sebaliknya, mereka memerlukan pengkodean karakter UTF-8. Akibatnya, karakter tertentu tidak dirender dengan benar.

Solusi

Untuk mengatasi masalah ini, tinjau aplikasi klien Anda untuk pengkodean karakter nama file dan pastikan itu diatur ke UTF-8.

Memecahkan masalah pengecualian **`ResourceNotFound`**

Deskripsi

Anda menerima kesalahan di mana sumber daya tidak dapat ditemukan. Misalnya, jika Anda menjalankan`UpdateServer`, Anda mungkin mendapatkan kesalahan berikut:

```
An error occurred (ResourceNotFoundException) when calling the UpdateServer operation:
 Unknown server
```

Menyebabkan

Ada beberapa alasan untuk menerima ResourceNotFoundExceptionpesan. Dalam kebanyakan kasus, sumber daya yang Anda tentukan dalam perintah API Anda tidak ada. Jika Anda memang menentukan sumber daya yang ada, maka penyebab yang paling mungkin adalah bahwa wilayah default Anda berbeda dari wilayah untuk sumber daya Anda. Misalnya, jika wilayah default Anda adalah us-east-1, dan server Transfer Family Anda berada di us-east-2, Anda akan menerima pengecualian sumber daya Tidak Dikenal.

Untuk detail tentang menyetel wilayah default, lihat Konfigurasi cepat dengan `aws configure`.

Solusi

Tambahkan parameter region ke perintah API Anda untuk secara eksplisit menentukan tempat menemukan sumber daya tertentu.

```
aws transfer -describe-server --server-id server-id --region us-east-2
```

Memecahkan masalah konektor SFTP

Bagian ini menjelaskan kemungkinan solusi untuk masalah konektor SFTP berikut.

Topik

- Negosiasi kunci gagal
- Masalah konektor SFTP lain-lain

Negosiasi kunci gagal

Deskripsi

Anda menerima kesalahan di mana negosiasi pertukaran kunci gagal. Sebagai contoh:

```
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]
```

Menyebabkan

Kesalahan ini karena tidak ada tumpang tindih antara algoritma kunci host yang didukung oleh server dan yang didukung oleh konektor.

Solusi

Pastikan server jarak jauh mendukung setidaknya satu dari algoritma kunci host Klien yang tercantum dalam pesan kesalahan. Untuk daftar algoritma yang didukung, lihatKebijakan keamanan untuk konektor AWS Transfer Family SFTP.

Masalah konektor SFTP lain-lain

Deskripsi

Anda menerima kesalahan setelah menjalankan`StartFileTransfer`, tetapi tidak tahu penyebab masalahnya, dan hanya ID konektor yang dikembalikan setelah panggilan API.

Menyebabkan

Kesalahan ini dapat memiliki beberapa penyebab. Untuk memecahkan masalah, kami sarankan Anda menguji konektor Anda dan mencari log Anda CloudWatch .

Solusi

- Uji konektor Anda: LihatUji konektor SFTP. Jika tes gagal, sistem memberikan pesan kesalahan berdasarkan alasan pengujian gagal. Bagian itu menjelaskan cara menguji konektor Anda baik dari konsol atau dengan menggunakan perintah TestConnectionAPI.
- Lihat CloudWatch log untuk konektor Anda: LihatContoh entri log untuk konektor SFTP. Topik ini memberikan contoh untuk entri log konektor SFTP, dan konvensi penamaan untuk membantu Anda menemukan log yang sesuai.

Memecahkan masalah AS2

Pesan kesalahan dan tips pemecahan masalah untuk server yang mendukung Pernyataan Penerapan 2 (AS2) dijelaskan di sini:. Kode kesalahan AS2

Referensi API

Bagian berikut mendokumentasikan panggilan layanan AWS Transfer Family API, tipe data, parameter, dan kesalahan.

Topik

- Selamat datang di AWS Transfer Family API
- Tindakan
- Tipe Data
- Membuat permintaan API
- Parameter Umum
- Kesalahan Umum

Selamat datang di AWS Transfer Family API

AWS Transfer Family adalah layanan transfer aman yang dapat Anda gunakan untuk mentransfer file masuk dan keluar dari penyimpanan Amazon Simple Storage Service (Amazon S3) melalui protokol berikut:

- Protokol Transfer File Secure Shell (SSH) (SFTP)
- Protokol Transfer File Aman (FTPS)
- Protokol Transfer File (FTP)
- Pernyataan Penerapan 2 (AS2)

Protokol transfer file digunakan dalam alur kerja pertukaran data di berbagai industri seperti layanan keuangan, perawatan kesehatan, periklanan, dan ritel, antara lain. AWS Transfer Family menyederhanakan migrasi alur kerja transfer file ke. AWS

Untuk menggunakan AWS Transfer Family layanan ini, Anda membuat instance server di AWS Wilayah pilihan Anda. Anda dapat membuat server, daftar server yang tersedia, dan memperbarui dan menghapus server. Server adalah entitas yang meminta operasi file dari AWS Transfer Family. Server memiliki sejumlah properti penting. Server adalah contoh bernama seperti yang diidentifikasi oleh `ServerId` pengidentifikasi yang ditetapkan sistem. Anda dapat secara opsional

menetapkan nama host, atau bahkan nama host khusus ke server. Tagihan layanan untuk setiap server instantiated (bahkan yang`OFFLINE`), dan untuk jumlah data yang ditransfer.

Pengguna harus diketahui oleh server yang meminta operasi file. Seorang pengguna yang diidentifikasi oleh nama pengguna mereka ditugaskan ke server. Nama pengguna digunakan untuk mengautentikasi permintaan. Server hanya dapat memiliki satu metode otentikasi:`AWS_DIRECTORY_SERVICE`,, `SERVICE_MANAGED`AWS_LAMBDA, atau`API_GATEWAY`.

Anda dapat menggunakan salah satu jenis penyedia identitas berikut untuk mengautentikasi pengguna:

- Untuk`SERVICE_MANAGED`, kunci publik SSH disimpan dengan properti pengguna di server. Seorang pengguna dapat memiliki satu atau lebih kunci publik SSH pada file untuk metode `SERVICE_MANAGED` otentikasi. Ketika klien meminta operasi file untuk `SERVICE_MANAGED` metode, klien menyediakan nama pengguna dan kunci pribadi SSH, yang diautentikasi, dan akses disediakan.
- Anda dapat mengelola otentikasi dan akses pengguna dengan grup Microsoft Active Directory Anda dengan memilih metode `AWS_DIRECTORY_SERVICE` otentikasi.
- Anda dapat terhubung ke penyedia identitas kustom dengan menggunakan AWS Lambda. Pilih metode `AWS_LAMBDA` otentikasi.
- Anda juga dapat mengautentikasi permintaan pengguna menggunakan metode otentikasi khusus yang menyediakan otentikasi dan akses pengguna. Metode ini bergantung pada Amazon API Gateway untuk menggunakan panggilan API Anda dari penyedia identitas Anda untuk memvalidasi permintaan pengguna. Metode ini disebut sebagai `API_GATEWAY` dalam panggilan API, dan sebagai Kustom di konsol. Anda dapat menggunakan metode kustom ini untuk mengautentikasi pengguna terhadap layanan direktori, pasangan nama/kata sandi database, atau mekanisme lainnya.

Pengguna diberi kebijakan dengan hubungan kepercayaan antara mereka dan bucket Amazon S3. Mereka mungkin dapat mengakses semua atau sebagian ember. Agar server bertindak atas nama pengguna, server harus mewarisi hubungan kepercayaan dari pengguna. Peran AWS Identity and Access Management (IAM) dibuat yang berisi hubungan kepercayaan, dan peran itu diberi `AssumeRole` tindakan. Server kemudian dapat melakukan operasi file seolah-olah itu adalah pengguna.

Pengguna yang memiliki set properti home direktori akan memiliki direktori (atau folder) yang bertindak sebagai target dan sumber operasi file. Ketika tidak ada home direktori yang disetel, `root` direktori bucket menjadi direktori pendaratan.

Server, pengguna, dan peran semuanya diidentifikasi oleh Amazon Resource Name (ARN) mereka. Anda dapat menetapkan tag, yang merupakan pasangan nilai kunci, ke entitas dengan ARN. Tag adalah metadata yang dapat digunakan untuk mengelompokkan atau mencari entitas ini. Salah satu contoh di mana tag berguna adalah untuk tujuan akuntansi.

Konvensi berikut diamati dalam format AWS Transfer Family ID:

- `ServerId`nilai mengambil bentuk`s-01234567890abcdef`.
- `SshPublicKeyId`nilai mengambil bentuk`key-01234567890abcdef`.

Format Amazon Resource Name (ARN) mengambil bentuk berikut:

- Untuk server, ARN mengambil formulir`arn:aws:transfer:`*`region`*`:`*`account-id`*`:server/`*`server-id`*.

  Contoh dari ARN server adalah: `arn:aws:transfer:us-east-1:123456789012:server/s-01234567890abcdef`.
- Untuk pengguna, ARN mengambil formulir`arn:aws:transfer:`*`region`*`:`*`account-id`*`:user/`*`server-id`*`/`*`username`*.

  Contohnya adalah `arn:aws:transfer:us-east-1:123456789012:user/s-01234567890abcdef/user1`.

Entri DNS (endpoint) yang digunakan adalah sebagai berikut:

- Titik akhir API mengambil formulir`transfer.`*`region`*`.amazonaws.com`.
- Titik akhir server mengambil formulir`server.transfer.`*`region`*`.amazonaws.com`.

Untuk daftar titik akhir Transfer Family menurut AWS Wilayah, lihat AWS Transfer Family titik akhir dan kuota di. Referensi Umum AWS

Referensi antarmuka API ini AWS Transfer Family berisi dokumentasi untuk antarmuka pemrograman yang dapat Anda gunakan untuk mengelola AWS Transfer Family. Struktur referensi adalah sebagai berikut:

- Untuk daftar tindakan API menurut abjad, lihat. Actions
- Untuk daftar alfabet tipe data, lihat. Data Types
- Untuk daftar parameter kueri umum, lihat Parameter Umum.
- Untuk deskripsi kode kesalahan, lihat Kesalahan Umum.

> **Tip**
>
> Daripada benar-benar menjalankan perintah, Anda dapat menggunakan `--generate-cli-skeleton` parameter dengan panggilan API apa pun untuk menghasilkan dan menampilkan templat parameter. Anda kemudian dapat menggunakan template yang dihasilkan untuk menyesuaikan dan digunakan sebagai input pada perintah selanjutnya. Untuk detailnya, lihat Menghasilkan dan menggunakan file kerangka parameter.

Tindakan

Tindakan berikut didukung:

- CreateAccess
- CreateAgreement
- CreateConnector
- CreateProfile
- CreateServer
- CreateUser
- CreateWorkflow
- DeleteAccess
- DeleteAgreement
- DeleteCertificate
- DeleteConnector
- DeleteHostKey
- DeleteProfile
- DeleteServer
- DeleteSshPublicKey

- DeleteUser
- DeleteWorkflow
- DescribeAccess
- DescribeAgreement
- DescribeCertificate
- DescribeConnector
- DescribeExecution
- DescribeHostKey
- DescribeProfile
- DescribeSecurityPolicy
- DescribeServer
- DescribeUser
- DescribeWorkflow
- ImportCertificate
- ImportHostKey
- ImportSshPublicKey
- ListAccesses
- ListAgreements
- ListCertificates
- ListConnectors
- ListExecutions
- ListHostKeys
- ListProfiles
- ListSecurityPolicies
- ListServers
- ListTagsForResource
- ListUsers
- ListWorkflows
- SendWorkflowStepState
- StartFileTransfer

- StartServer
- StopServer
- TagResource
- TestConnection
- TestIdentityProvider
- UntagResource
- UpdateAccess
- UpdateAgreement
- UpdateCertificate
- UpdateConnector
- UpdateHostKey
- UpdateProfile
- UpdateServer
- UpdateUser

CreateAccess

Digunakan oleh administrator untuk memilih grup mana dalam direktori yang harus memiliki akses untuk mengunggah dan mengunduh file melalui protokol yang diaktifkan menggunakan. AWS Transfer Family Misalnya, Microsoft Active Directory mungkin berisi 50.000 pengguna, tetapi hanya sebagian kecil yang mungkin memerlukan kemampuan untuk mentransfer file ke server. Administrator dapat menggunakan `CreateAccess` untuk membatasi akses ke set pengguna yang benar yang membutuhkan kemampuan ini.

Sintaksis Permintaan

```
{
   "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"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

ExternalId

Pengidentifikasi unik yang diperlukan untuk mengidentifikasi grup tertentu dalam direktori Anda. Pengguna grup yang Anda asosiasikan memiliki akses ke sumber daya Amazon S3 atau Amazon

EFS Anda melalui protokol yang diaktifkan. AWS Transfer Family Jika Anda tahu nama grup, Anda dapat melihat nilai SID dengan menjalankan perintah berikut menggunakan Windows PowerShell.

```
Get-ADGroup -Filter {samAccountName -like "YourGroupName*"} -Properties
* | Select SamAccountName,ObjectSid
```

Dalam perintah itu, ganti YourGroupNamedengan nama grup Active Directory Anda.

Ekspresi reguler yang digunakan untuk memvalidasi parameter ini adalah string karakter yang terdiri dari huruf besar dan huruf kecil karakter alfanumerik tanpa spasi. Anda juga dapat menyertakan garis bawah atau salah satu karakter berikut: =, . @: /-

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `S-1-[\d-]+`

Diperlukan: Ya

HomeDirectory

Direktori arahan (folder) untuk pengguna ketika mereka masuk ke server menggunakan klien.

Contoh `HomeDirectory` adalah `/bucket_name/home/mydirectory`.

> **Note**
>
> Parameter `HomeDirectory` hanya digunakan jika `HomeDirectoryType` diatur ke `PATH`.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 1024.

Pola: `(|/.*)`

Diperlukan: Tidak

HomeDirectoryMappings

Pemetaan direktori logis yang menentukan jalur dan kunci Amazon S3 atau Amazon EFS apa yang harus terlihat oleh pengguna Anda dan bagaimana Anda ingin membuatnya terlihat. Anda

harus menentukan `Entry` dan `Target` memasangkan, di mana `Entry` menunjukkan bagaimana jalur dibuat terlihat dan `Target` merupakan jalur Amazon S3 atau Amazon EFS yang sebenarnya. Jika Anda hanya menentukan target, itu ditampilkan apa adanya. Anda juga harus memastikan bahwa peran AWS Identity and Access Management (IAM) Anda menyediakan akses ke jalur masuk`Target`. Nilai ini dapat diatur hanya ketika `HomeDirectoryType` diatur ke LOGICAL.

Berikut ini adalah contoh `Entry` dan `Target` pasangkan.

```
[ { "Entry": "/directory1", "Target": "/bucket_name/home/
mydirectory" } ]
```

Dalam kebanyakan kasus, Anda dapat menggunakan nilai ini alih-alih kebijakan sesi untuk mengunci pengguna Anda ke direktori home yang ditunjuk (`chroot`“”). Untuk melakukan ini, Anda dapat mengatur `Entry` ke `/` dan mengatur `Target` ke nilai `HomeDirectory` parameter.

Berikut ini adalah contoh `Entry` dan `Target` pair untuk`chroot`.

```
[ { "Entry": "/", "Target": "/bucket_name/home/mydirectory" } ]
```

Tipe: Array objek HomeDirectoryMapEntry

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50000 item.

Diperlukan: Tidak

HomeDirectoryType

Jenis direktori pendaratan (folder) yang Anda inginkan direktori home pengguna Anda ketika mereka masuk ke server. Jika Anda mengaturnya`PATH`, pengguna akan melihat bucket Amazon S3 absolut atau jalur Amazon EFS seperti pada klien protokol transfer file mereka. Jika Anda menyetelnya`LOGICAL`, Anda harus menyediakan pemetaan `HomeDirectoryMappings` untuk bagaimana Anda ingin membuat jalur Amazon S3 atau Amazon EFS terlihat oleh pengguna Anda.

> **Note**
>
> Jika `HomeDirectoryType` ya`LOGICAL`, Anda harus memberikan pemetaan, menggunakan parameter. `HomeDirectoryMappings` Jika, di sisi lain, `HomeDirectoryType` adalah`PATH`, Anda memberikan jalur absolut menggunakan `HomeDirectory` parameter. Anda tidak dapat memiliki keduanya `HomeDirectory` dan `HomeDirectoryMappings` di template Anda.

Jenis: String

Nilai yang Valid: `PATH | LOGICAL`

Diperlukan: Tidak

Policy

Kebijakan sesi untuk pengguna Anda sehingga Anda dapat menggunakan peran yang sama AWS Identity and Access Management (IAM) di beberapa pengguna. Kebijakan ini mencakup akses pengguna ke sebagian bucket Amazon S3 mereka. Variabel yang dapat Anda gunakan dalam kebijakan ini meliputi `${Transfer:UserName}`, `${Transfer:HomeDirectory}`, dan `${Transfer:HomeBucket}`.

> **Note**
>
> Kebijakan ini hanya berlaku jika domainnya `ServerId` adalah Amazon S3. Amazon EFS tidak menggunakan kebijakan sesi.
> Untuk kebijakan sesi, AWS Transfer Family menyimpan kebijakan sebagai gumpalan JSON, bukan Nama Sumber Daya Amazon (ARN) kebijakan tersebut. Anda menyimpan kebijakan sebagai blob JSON dan meneruskan dalam argumen `Policy`.
> Untuk contoh kebijakan sesi, lihat Contoh kebijakan sesi.
> Untuk informasi selengkapnya, lihat AssumeRoledi Referensi AWS Security Token Service API.

Jenis: String

Batasan Panjang: Panjang minimum sebesar 0. Panjang maksimum 2048.

Diperlukan: Tidak

PosixProfile

Identitas POSIX lengkap, termasuk ID pengguna (`Uid`), ID grup (`Gid`), dan setiap grup sekunder ID (`SecondaryGids`), yang mengendalikan akses pengguna Anda ke sistem file Amazon EFS Anda. POSIX izin yang ditetapkan pada file dan direktori dalam sistem file Anda menentukan tingkat akses yang pengguna Anda dapatkan ketika mentransfer file ke dalam dan keluar dari sistem file Amazon EFS Anda.

Tipe: Objek PosixProfile

Diperlukan: Tidak

Role

Nama Sumber Daya Amazon (ARN) dari peran AWS Identity and Access Management (IAM) yang mengontrol akses pengguna ke bucket Amazon S3 atau sistem file Amazon EFS. Kebijakan yang dilampirkan pada peran ini menentukan tingkat akses yang ingin Anda berikan kepada pengguna saat mentransfer file masuk dan keluar dari bucket Amazon S3 atau sistem file Amazon EFS Anda. IAM role juga harus berisi hubungan kepercayaan yang mengizinkan server untuk mengakses sumber daya Anda saat melayani permintaan transfer pengguna.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 2048.

Pola: `arn:.*role/\S+`

Diperlukan: Ya

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk instans server. Ini adalah server tertentu tempat Anda menambahkan pengguna.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

Sintaksis Respons

```
{
   "ExternalId": "string",
   "ServerId": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

ExternalId

Pengenal eksternal grup yang penggunanya memiliki akses ke sumber daya Amazon S3 atau Amazon EFS Anda melalui protokol yang diaktifkan yang digunakan. AWS Transfer Family

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `S-1-[\d-]+`

ServerId

Pengidentifikasi server tempat pengguna dilampirkan.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceExistsException

Sumber daya yang diminta tidak ada, atau ada di wilayah selain yang ditentukan untuk perintah.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

CreateAgreement

Membuat perjanjian. Perjanjian adalah perjanjian mitra dagang bilateral, atau kemitraan, antara AWS Transfer Family server dan proses AS2. Perjanjian mendefinisikan hubungan transfer file dan pesan antara server dan proses AS2. Untuk menentukan perjanjian, Transfer Family menggabungkan server, profil lokal, profil mitra, sertifikat, dan atribut lainnya.

Mitra diidentifikasi dengan`PartnerProfileId`, dan proses AS2 diidentifikasi dengan. `LocalProfileId`

Sintaksis Permintaan

```
{
   "AccessRole": "string",
   "BaseDirectory": "string",
   "Description": "string",
   "LocalProfileId": "string",
   "PartnerProfileId": "string",
   "ServerId": "string",
   "Status": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

AccessRole

Konektor digunakan untuk mengirim file menggunakan protokol AS2 atau SFTP. Untuk peran akses, berikan Nama Sumber Daya Amazon (ARN) AWS Identity and Access Management peran yang akan digunakan.

Untuk konektor AS2

Dengan AS2, Anda dapat mengirim file dengan memanggil `StartFileTransfer` dan menentukan jalur file dalam parameter permintaan,. `SendFilePaths` Kami menggunakan direktori induk file (misalnya, untuk, direktori induk`/bucket/dir/`) untuk `--send-file-paths /bucket/dir/file.txt` sementara menyimpan file pesan AS2 yang diproses, menyimpan MDN ketika kami menerimanya dari mitra, dan menulis file JSON akhir yang berisi metadata transmisi yang relevan. Jadi, `AccessRole` kebutuhan untuk menyediakan akses baca dan tulis ke direktori induk dari lokasi file yang digunakan dalam `StartFileTransfer` permintaan. Selain itu, Anda perlu menyediakan akses baca dan tulis ke direktori induk dari file yang ingin Anda kirim`StartFileTransfer`.

Jika Anda menggunakan otentikasi Dasar untuk konektor AS2 Anda, peran akses memerlukan `secretsmanager:GetSecretValue` izin untuk rahasia tersebut. Jika rahasia dienkripsi menggunakan kunci yang dikelola pelanggan alih-alih kunci yang dikelola di AWS Secrets Manager, maka peran tersebut juga memerlukan `kms:Decrypt` izin untuk kunci tersebut.

Untuk konektor SFTP

Pastikan bahwa peran akses menyediakan akses baca dan tulis ke direktori induk dari lokasi file yang digunakan dalam `StartFileTransfer` permintaan. Selain itu, pastikan bahwa peran tersebut memberikan `secretsmanager:GetSecretValue` izin untuk AWS Secrets Manager.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 2048.

Pola: `arn:.*role/\S+`

Diperlukan: Ya

BaseDirectory

Direktori pendaratan (folder) untuk file yang ditransfer dengan menggunakan protokol AS2.

Contoh `BaseDirectory` adalah `/DOC-EXAMPLE-BUCKET/home/mydirectory`.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 1024.

Pola: `(|/.*)`

Diperlukan: Ya

Description

Nama atau deskripsi singkat untuk mengidentifikasi perjanjian.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 200.

Pola: `[\p{Graph}]+`

Diperlukan: Tidak

LocalProfileId

Pengidentifikasi unik untuk profil lokal AS2.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `p-([0-9a-f]{17})`

Diperlukan: Ya

PartnerProfileId

Pengenal unik untuk profil mitra yang digunakan dalam perjanjian.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `p-([0-9a-f]{17})`

Diperlukan: Ya

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk instans server. Ini adalah server khusus yang digunakan perjanjian.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

Status

Status perjanjian. Kesepakatan itu bisa berupa `ACTIVE` atau`INACTIVE`.

Jenis: String

Nilai yang Valid: `ACTIVE | INACTIVE`

Diperlukan: Tidak

Tags

Pasangan nilai kunci yang dapat digunakan untuk mengelompokkan dan mencari perjanjian.

Tipe: Array objek Tag

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50 item.

Diperlukan: Tidak

Sintaksis Respons

```
{
   "AgreementId": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

AgreementId

Pengidentifikasi unik untuk perjanjian. Gunakan ID ini untuk menghapus, atau memperbarui perjanjian, serta panggilan API lainnya yang mengharuskan Anda menentukan ID perjanjian.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `a-([0-9a-f]{17})`

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilemparkan ketika kesalahan terjadi dalam AWS Transfer Family layanan.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceExistsException

Sumber daya yang diminta tidak ada, atau ada di wilayah selain yang ditentukan untuk perintah.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

ThrottlingException

Permintaan ditolak karena throttling permintaan.

Kode Status HTTP: 400

Contoh-contoh

Contoh

Contoh berikut membuat perjanjian, dan mengembalikan ID perjanjian.

```
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
```

Contoh Respons

Panggilan API mengembalikan ID perjanjian untuk perjanjian baru.

```
{
    "AgreementId": "a-11112222333344444"
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

CreateConnector

Membuat konektor, yang menangkap parameter untuk koneksi untuk protokol AS2 atau SFTP. Untuk AS2, konektor diperlukan untuk mengirim file ke server AS2 yang dihosting secara eksternal. Untuk SFTP, konektor diperlukan saat mengirim file ke server SFTP atau menerima file dari server SFTP. Untuk detail selengkapnya tentang konektor, lihat Mengkonfigurasi konektor AS2 dan Membuat konektor SFTP.

> **Note**
>
> Anda harus menentukan tepat satu objek konfigurasi: baik untuk AS2 (`As2Config`) atau SFTP (). `SftpConfig`

Sintaksis Permintaan

```
{
   "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"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

AccessRole

Konektor digunakan untuk mengirim file menggunakan protokol AS2 atau SFTP. Untuk peran akses, berikan Nama Sumber Daya Amazon (ARN) AWS Identity and Access Management peran yang akan digunakan.

Untuk konektor AS2

Dengan AS2, Anda dapat mengirim file dengan memanggil `StartFileTransfer` dan menentukan jalur file dalam parameter permintaan,. `SendFilePaths` Kami menggunakan direktori induk file (misalnya, untuk, direktori induk`/bucket/dir/`) untuk `--send-file-paths /bucket/dir/file.txt` sementara menyimpan file pesan AS2 yang diproses, menyimpan MDN ketika kami menerimanya dari mitra, dan menulis file JSON akhir yang berisi metadata transmisi yang relevan. Jadi, `AccessRole` kebutuhan untuk menyediakan akses baca dan tulis ke direktori induk dari lokasi file yang digunakan dalam `StartFileTransfer` permintaan. Selain itu, Anda perlu menyediakan akses baca dan tulis ke direktori induk dari file yang ingin Anda kirim`StartFileTransfer`.

Jika Anda menggunakan otentikasi Dasar untuk konektor AS2 Anda, peran akses memerlukan `secretsmanager:GetSecretValue` izin untuk rahasia tersebut. Jika rahasia dienkripsi menggunakan kunci yang dikelola pelanggan alih-alih kunci yang dikelola di AWS Secrets Manager, maka peran tersebut juga memerlukan `kms:Decrypt` izin untuk kunci tersebut.

Untuk konektor SFTP

Pastikan bahwa peran akses menyediakan akses baca dan tulis ke direktori induk dari lokasi file yang digunakan dalam `StartFileTransfer` permintaan. Selain itu, pastikan bahwa peran tersebut memberikan `secretsmanager:GetSecretValue` izin untuk AWS Secrets Manager.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 2048.

Pola: `arn:.*role/\S+`

Diperlukan: Ya

As2Config

Struktur yang berisi parameter untuk objek konektor AS2.

Tipe: Objek As2ConnectorConfig

Diperlukan: Tidak

LoggingRole

Nama Sumber Daya Amazon (ARN) dari peran AWS Identity and Access Management (IAM) yang memungkinkan konektor mengaktifkan CloudWatch logging untuk peristiwa Amazon S3. Saat disetel, Anda dapat melihat aktivitas konektor di CloudWatch log Anda.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 2048.

Pola: `arn:.*role/\S+`

Diperlukan: Tidak

SecurityPolicyName

Menentukan nama kebijakan keamanan untuk konektor.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 100.

Pola: `TransferSFTPConnectorSecurityPolicy-[A-Za-z0-9-]+`

Diperlukan: Tidak

SftpConfig

Struktur yang berisi parameter untuk objek konektor SFTP.

Tipe: Objek SftpConnectorConfig

Diperlukan: Tidak

Tags

Pasangan nilai kunci yang dapat digunakan untuk mengelompokkan dan mencari konektor. Tag adalah metadata yang terpasang pada konektor untuk tujuan apa pun.

Tipe: Array objek Tag

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50 item.

Diperlukan: Tidak

Url

URL titik akhir AS2 atau SFTP mitra.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum sebesar 255.

Diperlukan: Ya

Sintaksis Respons

```
{
   "ConnectorId": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

ConnectorId

Pengenal unik untuk konektor, dikembalikan setelah panggilan API berhasil.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `c-([0-9a-f]{17})`

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilemparkan ketika kesalahan terjadi dalam AWS Transfer Family layanan.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceExistsException

Sumber daya yang diminta tidak ada, atau ada di wilayah selain yang ditentukan untuk perintah.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

ThrottlingException

Permintaan ditolak karena throttling permintaan.

Kode Status HTTP: 400

Contoh-contoh

Contoh

Contoh berikut membuat konektor AS2. Dalam perintah, ganti item sebagai berikut:

- `url`: berikan URL untuk server AS2 mitra dagang.
- `your-IAM-role-for-bucket-access`: peran IAM yang memiliki akses ke bucket Amazon S3 yang Anda gunakan untuk menyimpan file Anda.
- Gunakan ARN untuk peran logging Anda, yang menyertakan ID Anda Akun AWS .
- Berikan path ke file yang berisi parameter konfigurasi konektor AS2. Objek konfigurasi konektor AS2 dijelaskan dalam ConnectorConfigAs2.

```
// 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
```

Contoh

Contoh berikut membuat konektor SFTP. Dalam perintah, ganti item sebagai berikut:

- `sftp-server-url`: berikan URL untuk server SFTP tempat Anda bertukar file.
- `your-IAM-role-for-bucket-access`: peran IAM yang memiliki akses ke bucket Amazon S3 yang Anda gunakan untuk menyimpan file Anda.
- Gunakan ARN untuk peran logging Anda, yang menyertakan ID Anda Akun AWS .
- Berikan jalur ke file yang berisi parameter konfigurasi konektor SFTP. Objek konfigurasi konektor SFTP dijelaskan dalam. SftpConnectorConfig

```
// Listing for testSFTPConfig.json
{
   "UserSecretId": "arn:aws:secretsmanager:us-east-2:123456789012:secret:aws/transfer/
example-username-key",
   "TrustedHostKeys": [
      "sftp.example.com ssh-rsa AAAAbbbb...EEEE="
   ]
}
```

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

Contoh

Panggilan API mengembalikan ID konektor untuk konektor baru.

Contoh Respons

```
{
    "ConnectorId": "a-11112222333344444"
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3

- AWS SDK untuk Python
- AWS SDK for Ruby V3

CreateProfile

Membuat profil lokal atau mitra untuk digunakan untuk transfer AS2.

Sintaksis Permintaan

```
{
   "As2Id": "string",
   "CertificateIds": [ "string" ],
   "ProfileType": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

As2Id

`As2Id`Itu adalah nama AS2, seperti yang didefinisikan dalam RFC 4130. Untuk transfer masuk, ini adalah `AS2-From` header untuk pesan AS2 yang dikirim dari mitra. Untuk konektor keluar, ini adalah `AS2-To` header untuk pesan AS2 yang dikirim ke mitra menggunakan operasi `StartFileTransfer` API. ID ini tidak dapat menyertakan spasi.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: `[\p{Print}\s]*`

Diperlukan: Ya

CertificateIds

Array pengidentifikasi untuk sertifikat yang diimpor. Anda menggunakan pengenal ini untuk bekerja dengan profil dan profil mitra.

Tipe: Array string

Kendala Panjang: Panjang tetap 22.

Pola: `cert-([0-9a-f]{17})`

Diperlukan: Tidak

ProfileType

Menentukan jenis profil yang akan dibuat:

- Tentukan `LOCAL` untuk membuat profil lokal. Profil lokal mewakili organisasi atau pihak server Transfer Family AS2 yang diaktifkan.
- Tentukan `PARTNER` untuk membuat profil mitra. Profil mitra mewakili organisasi jarak jauh, di luar Transfer Family.

Jenis: String

Nilai yang Valid: `LOCAL | PARTNER`

Diperlukan: Ya

Tags

Pasangan nilai kunci yang dapat digunakan untuk mengelompokkan dan mencari profil AS2.

Tipe: Array objek Tag

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50 item.

Diperlukan: Tidak

Sintaksis Respons

```
{
   "ProfileId": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

ProfileId

Pengenal unik untuk profil AS2, ditampilkan setelah panggilan API berhasil.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `p-([0-9a-f]{17})`

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilemparkan ketika kesalahan terjadi dalam AWS Transfer Family layanan.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

ThrottlingException

Permintaan ditolak karena throttling permintaan.

Kode Status HTTP: 400

Contoh-contoh

Contoh

Contoh berikut membuat profil, dan mengembalikan ID profil.

ID sertifikat dibuat saat Anda menjalankan`import-certificate`, satu untuk sertifikat penandatanganan, dan satu untuk sertifikat enkripsi.

```
aws transfer create-profile --as2-id MYCORP --certificate-ids c-abcdefg123456hijk
               c-987654aaaa321bbbb
```

Contoh Respons

Panggilan API mengembalikan ID profil untuk profil baru.

```
{
    "ProfileId": "p-11112222333344444"
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

CreateServer

Membuat instance server virtual auto-scaling berdasarkan protokol transfer file yang dipilih di. AWS Saat Anda melakukan pembaruan ke server berkemampuan protokol transfer file atau saat Anda bekerja dengan pengguna, gunakan `ServerId` properti yang dihasilkan layanan yang ditetapkan ke server yang baru dibuat.

Sintaksis Permintaan

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

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

Certificate

Nama Sumber Daya Amazon (ARN) dari sertifikat AWS Certificate Manager (ACM). Diperlukan saat `Protocols` diatur ke `FTPS`.

Untuk meminta sertifikat publik baru, lihat Meminta sertifikat publik di Panduan AWS Certificate Manager Pengguna.

Untuk mengimpor sertifikat yang ada ke ACM, lihat Mengimpor sertifikat ke ACM di Panduan Pengguna. AWS Certificate Manager

Untuk meminta sertifikat pribadi untuk menggunakan FTPS melalui alamat IP pribadi, lihat Meminta sertifikat pribadi di Panduan AWS Certificate Manager Pengguna.

Sertifikat dengan algoritme kriptografi dan ukuran kunci berikut didukung:

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

> **Note**
>
> Sertifikat harus berupa sertifikat SSL/TLS X.509 versi 3 yang valid dengan FQDN atau alamat IP yang ditentukan dan informasi tentang penerbitnya.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 1600.

Diperlukan: Tidak

Domain

Domain dari sistem penyimpanan yang digunakan untuk transfer file. Ada dua domain yang tersedia: Amazon Simple Storage Service (Amazon S3) dan Amazon Elastic File System (Amazon EFS). Nilai defaultnya adalah S3.

> **Note**
>
> Setelah server dibuat, domain tidak dapat diubah.

Jenis: String

Nilai yang Valid: `S3 | EFS`

Diperlukan: Tidak

EndpointDetails

Pengaturan titik akhir virtual private cloud (VPC) yang dikonfigurasi untuk server Anda. Ketika Anda meng-host titik akhir Anda dalam VPC Anda, Anda dapat membuat titik akhir Anda hanya dapat diakses oleh sumber daya dalam VPC Anda, atau Anda dapat melampirkan alamat IP

Elastis dan membuat titik akhir Anda dapat diakses oleh klien melalui internet. Grup keamanan default VPC Anda secara otomatis ditetapkan ke titik akhir Anda.

Tipe: Objek EndpointDetails

Diperlukan: Tidak

EndpointType

Jenis endpoint yang Anda ingin server Anda gunakan. Anda dapat memilih untuk membuat endpoint server Anda dapat diakses publik (PUBLIK) atau menghostingnya di dalam VPC Anda. Dengan endpoint yang di-host di VPC, Anda dapat membatasi akses ke server dan sumber daya hanya dalam VPC Anda atau memilih untuk membuatnya menghadap internet dengan melampirkan alamat IP Elastis langsung ke sana.

> **Note**
>
> Setelah 19 Mei 2021, Anda tidak akan dapat membuat server menggunakan `EndpointType=VPC_ENDPOINT` di akun Anda Akun AWS jika akun Anda belum melakukannya sebelum 19 Mei 2021. Jika Anda telah membuat server dengan `EndpointType=VPC_ENDPOINT` di Anda Akun AWS pada atau sebelum 19 Mei 2021, Anda tidak akan terpengaruh. Setelah tanggal ini, gunakan `EndpointType =VPC`. Untuk informasi selengkapnya, lihat Menghentikan penggunaan VPC_ENDPOINT. Direkomendasikan agar Anda menggunakan `VPC` sebagai `EndpointType`. Dengan jenis titik akhir ini, Anda memiliki pilihan untuk secara langsung mengaitkan hingga tiga alamat IPv4 Elastis (termasuk IP BYO) dengan titik akhir server Anda dan menggunakan grup keamanan VPC untuk membatasi lalu lintas berdasarkan alamat IP publik klien. Hal ini tidak mungkin terjadi jika `EndpointType` diatur ke `VPC_ENDPOINT`.

Jenis: String

Nilai yang Valid: `PUBLIC | VPC | VPC_ENDPOINT`

Diperlukan: Tidak

HostKey

Kunci pribadi RSA, ECDSA, atau ED25519 untuk digunakan untuk server berkemampuan SFTP Anda. Anda dapat menambahkan beberapa kunci host, jika Anda ingin memutar tombol, atau memiliki satu set kunci aktif yang menggunakan algoritma yang berbeda.

Gunakan perintah berikut untuk menghasilkan kunci RSA 2048 bit tanpa frasa sandi:

```
ssh-keygen -t rsa -b 2048 -N "" -m PEM -f my-new-server-key.
```

Gunakan nilai minimum 2048 untuk `-b` opsi. Anda dapat membuat kunci yang lebih kuat dengan menggunakan 3072 atau 4096.

Gunakan perintah berikut untuk menghasilkan kunci ECDSA 256 bit tanpa frasa sandi:

```
ssh-keygen -t ecdsa -b 256 -N "" -m PEM -f my-new-server-key.
```

Nilai yang valid untuk `-b` opsi ECDSA adalah 256, 384, dan 521.

Gunakan perintah berikut untuk menghasilkan kunci ED25519 tanpa frasa sandi:

```
ssh-keygen -t ed25519 -N "" -f my-new-server-key.
```

Untuk semua perintah ini, Anda dapat mengganti my-new-server-keydengan string pilihan Anda.

> **Important**
>
> Jika Anda tidak berencana untuk memigrasikan pengguna yang ada dari server berkemampuan SFTP yang ada ke server baru, jangan perbarui kunci host. Mengubah kunci host server secara tidak sengaja dapat mengganggu.

Untuk informasi selengkapnya, lihat Memperbarui kunci host untuk server berkemampuan SFTP di Panduan Pengguna. AWS Transfer Family

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 4096.

Diperlukan: Tidak

IdentityProviderDetails

Diperlukan saat `IdentityProviderType` diatur ke`AWS_DIRECTORY_SERVICE`, `AWS_LAMBDA` atau`API_GATEWAY`. Menerima array yang berisi semua informasi yang diperlukan untuk menggunakan direktori di `AWS_DIRECTORY_SERVICE` atau menjalankan API autentikasi yang disediakan pelanggan, termasuk URL API Gateway. Tidak diperlukan saat `IdentityProviderType` diatur ke `SERVICE_MANAGED`.

Tipe: Objek IdentityProviderDetails

Diperlukan: Tidak

IdentityProviderType

Modus otentikasi untuk server. Nilai defaultnya adalahSERVICE_MANAGED, yang memungkinkan Anda untuk menyimpan dan mengakses kredensyal pengguna dalam layanan. AWS Transfer Family

Gunakan AWS_DIRECTORY_SERVICE untuk menyediakan akses ke grup Direktori Aktif di AWS Directory Service for Microsoft Active Directory atau Microsoft Active Directory di lingkungan lokal Anda atau AWS menggunakan AD Connector. Opsi ini juga mengharuskan Anda untuk memberikan ID Direktori dengan menggunakan IdentityProviderDetails parameter.

Gunakan nilai API_GATEWAY untuk mengintegrasikan dengan penyedia identitas pilihan Anda. API_GATEWAYPengaturan mengharuskan Anda untuk menyediakan URL titik akhir Amazon API Gateway untuk memanggil otentikasi dengan menggunakan parameter. IdentityProviderDetails

Gunakan AWS_LAMBDA nilai untuk langsung menggunakan AWS Lambda fungsi sebagai penyedia identitas Anda. Jika Anda memilih nilai ini, Anda harus menentukan ARN untuk fungsi Lambda dalam Function parameter untuk tipe data. IdentityProviderDetails

Jenis: String

Nilai yang Valid: SERVICE_MANAGED | API_GATEWAY | AWS_DIRECTORY_SERVICE | AWS_LAMBDA

Diperlukan: Tidak

LoggingRole

Nama Sumber Daya Amazon (ARN) dari peran AWS Identity and Access Management (IAM) yang memungkinkan server mengaktifkan CloudWatch pencatatan Amazon untuk Amazon S3 atau Amazon EFSevents. Saat disetel, Anda dapat melihat aktivitas pengguna di CloudWatch log Anda.

Jenis: String

Batasan Panjang: Panjang minimum sebesar 0. Panjang maksimum 2048.

Pola: (|arn:.*role/\S+)

Diperlukan: Tidak

PostAuthenticationLoginBanner

Menentukan string untuk ditampilkan ketika pengguna terhubung ke server. String ini ditampilkan setelah pengguna mengautentikasi.

> **Note**
>
> Protokol SFTP tidak mendukung spanduk tampilan pasca-otentikasi.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 4096.

Pola: `[\x09-\x0D\x20-\x7E]*`

Diperlukan: Tidak

PreAuthenticationLoginBanner

Menentukan string untuk ditampilkan ketika pengguna terhubung ke server. String ini ditampilkan sebelum pengguna mengautentikasi. Misalnya, spanduk berikut menampilkan detail tentang penggunaan sistem:

```
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.
```

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 4096.

Pola: `[\x09-\x0D\x20-\x7E]*`

Diperlukan: Tidak

ProtocolDetails

Pengaturan protokol yang dikonfigurasi untuk server Anda.

- Untuk menunjukkan mode pasif (untuk protokol FTP dan FTPS), gunakan parameter. `PassiveIp` Masukkan satu alamat IPv4 bertitik quad, seperti alamat IP eksternal dari firewall, router, atau penyeimbang beban.
- Untuk mengabaikan kesalahan yang dihasilkan saat klien mencoba menggunakan `SETSTAT` perintah pada file yang Anda unggah ke bucket Amazon S3, gunakan `SetStatOption` parameternya. Agar AWS Transfer Family server mengabaikan `SETSTAT` perintah dan mengunggah file tanpa perlu membuat perubahan apa pun pada klien SFTP Anda, atur nilainya. `ENABLE_NO_OP` Jika Anda menyetel `SetStatOption` parameternya`ENABLE_NO_OP`, Transfer Family akan menghasilkan entri CloudWatch log ke Amazon Logs, sehingga Anda dapat menentukan kapan klien melakukan `SETSTAT` panggilan.
- Untuk menentukan apakah AWS Transfer Family server Anda melanjutkan sesi terbaru yang dinegosiasikan melalui ID sesi unik, gunakan parameter. `TlsSessionResumptionMode`
- `As2Transports`menunjukkan metode transport untuk pesan AS2. Saat ini, hanya HTTP yang didukung.

Tipe: Objek ProtocolDetails

Diperlukan: Tidak

Protocols

Menentukan protokol transfer file atau protokol di mana klien protokol transfer file Anda dapat terhubung ke titik akhir server Anda. Protokol yang tersedia adalah:

- `SFTP` (Secure Shell (SSH) Protokol Transfer File): Transfer file melalui SSH
- `FTPS` (File Transfer Protocol Secure): Transfer file dengan enkripsi TLS
- `FTP` (Protokol Transfer File): Transfer file tidak terenkripsi
- `AS2`(Pernyataan Penerapan 2): digunakan untuk mengangkut data terstruktur business-to-business

> **Note**
>
> - Jika Anda memilih`FTPS`, Anda harus memilih sertifikat yang disimpan di AWS Certificate Manager (ACM) yang digunakan untuk mengidentifikasi server Anda ketika klien terhubung ke sana melalui FTPS.
> - Jika `Protocol` termasuk salah satu `FTP` atau`FTPS`, maka `EndpointType` harus `VPC` dan `IdentityProviderType` harus baik`AWS_DIRECTORY_SERVICE`,`AWS_LAMBDA`, atau`API_GATEWAY`.

- Jika `Protocol` termasuk `FTP`, maka `AddressAllocationIds` tidak dapat dikaitkan.
- Jika `Protocol` disetel hanya ke`SFTP`, `EndpointType` dapat diatur ke `PUBLIC` dan `IdentityProviderType` dapat disetel salah satu jenis identitas yang didukung:`SERVICE_MANAGED`,`AWS_DIRECTORY_SERVICE`,`AWS_LAMBDA`, atau`API_GATEWAY`.
- Jika `Protocol` termasuk`AS2`, maka `EndpointType` harus`VPC`, dan domain harus Amazon S3.

Tipe: Array string

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 4 item.

Nilai yang Valid: `SFTP | FTP | FTPS | AS2`

Diperlukan: Tidak

S3StorageOptions

Menentukan apakah atau tidak kinerja untuk direktori Amazon S3 Anda dioptimalkan. Ini dinonaktifkan secara default.

Secara default, pemetaan direktori home memiliki `TYPE` file. `DIRECTORY` Jika Anda mengaktifkan opsi ini, Anda kemudian perlu secara eksplisit mengatur `HomeDirectoryMapEntry` `Type` ke `FILE` jika Anda ingin pemetaan memiliki target file.

Tipe: Objek S3StorageOptions

Diperlukan: Tidak

SecurityPolicyName

Menentukan nama kebijakan keamanan untuk server.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 100.

Pola: `Transfer[A-Za-z0-9]*SecurityPolicy-[A-Za-z0-9-]+`

Diperlukan: Tidak

StructuredLogDestinations

Menentukan grup log yang log server Anda dikirim.

Untuk menentukan grup log, Anda harus memberikan ARN untuk grup log yang ada. Dalam hal ini, format grup log adalah sebagai berikut:

`arn:aws:logs:region-name:amazon-account-id:log-group:log-group-name:*`

Misalnya, `arn:aws:logs:us-east-1:111122223333:log-group:mytestgroup:*`

Jika sebelumnya Anda telah menentukan grup log untuk server, Anda dapat menghapusnya, dan pada dasarnya mematikan logging terstruktur, dengan memberikan nilai kosong untuk parameter ini dalam `update-server` panggilan. Sebagai contoh:

```
update-server --server-id s-1234567890abcdef0 --structured-log-
destinations
```

Tipe: Array string.

Anggota Array: Jumlah minimum 0 item. Jumlah maksimum 1 item.

Batasan Panjang: Panjang minimum 20. Panjang maksimum 1600.

Pola: `arn:\S+`

Diperlukan: Tidak

Tags

Pasangan nilai kunci yang dapat digunakan untuk grup dan mencari server.

Tipe: Array objek Tag

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50 item.

Diperlukan: Tidak

WorkflowDetails

Menentukan ID alur kerja untuk alur kerja yang akan ditetapkan dan peran eksekusi yang digunakan untuk mengeksekusi alur kerja.

Selain alur kerja untuk mengeksekusi ketika file diunggah sepenuhnya, juga `WorkflowDetails` dapat berisi ID alur kerja (dan peran eksekusi) untuk alur kerja untuk mengeksekusi pada upload sebagian. Upload sebagian terjadi ketika sesi server terputus saat file masih diunggah.

Tipe: Objek WorkflowDetails

Wajib: Tidak

Sintaksis Respons

```
{
   "ServerId": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

ServerId

Pengenal yang ditugaskan layanan dari server yang dibuat.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

AccessDeniedException

Anda tidak memiliki akses yang memadai untuk melakukan tindakan ini.

Kode Status HTTP: 400

InternalServiceError

Pengecualian ini dilemparkan ketika kesalahan terjadi dalam AWS Transfer Family layanan.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceExistsException

Sumber daya yang diminta tidak ada, atau ada di wilayah selain yang ditentukan untuk perintah.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

ThrottlingException

Permintaan ditolak karena throttling permintaan.

Kode Status HTTP: 400

Contoh-contoh

Contoh

Contoh berikut membuat server baru menggunakan file`VPC_ENDPOINT`.

Permintaan Sampel

```
{
   "EndpointType": "VPC",
   "EndpointDetails":...,
   "HostKey": "Your RSA private key",
   "IdentityProviderDetails": "IdentityProvider",
   "IdentityProviderType": "SERVICE_MANAGED",
```

```
   "LoggingRole": "CloudWatchLoggingRole",
   "Tags": [
      {
         "Key": "Name",
         "Value": "MyServer"
      }
   ]
}
```

Contoh

Ini adalah contoh respons untuk panggilan API ini.

Contoh Respons

```
{
   "ServerId": "s-01234567890abcdef"
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

CreateUser

Membuat pengguna dan mengaitkannya dengan server berkemampuan protokol transfer file yang ada. Anda hanya dapat membuat dan mengaitkan pengguna dengan server yang memiliki `IdentityProviderType` atur ke `SERVICE_MANAGED`. Menggunakan parameter untuk`CreateUser`, Anda dapat menentukan nama pengguna, mengatur direktori home, menyimpan kunci publik pengguna, dan menetapkan peran pengguna AWS Identity and Access Management (IAM). Anda juga dapat menambahkan kebijakan sesi secara opsional, dan menetapkan metadata dengan tag yang dapat digunakan untuk mengelompokkan dan mencari pengguna.

Sintaksis Permintaan

```
{
   "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"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

HomeDirectory

Direktori arahan (folder) untuk pengguna ketika mereka masuk ke server menggunakan klien.

Contoh `HomeDirectory` adalah `/bucket_name/home/mydirectory`.

> **Note**
>
> Parameter `HomeDirectory` hanya digunakan jika `HomeDirectoryType` diatur ke `PATH`.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 1024.

Pola: `(|/.*)`

Diperlukan: Tidak

HomeDirectoryMappings

Pemetaan direktori logis yang menentukan jalur dan kunci Amazon S3 atau Amazon EFS apa yang harus terlihat oleh pengguna Anda dan bagaimana Anda ingin membuatnya terlihat. Anda harus menentukan `Entry` dan `Target` memasangkan, di mana `Entry` menunjukkan bagaimana jalur dibuat terlihat dan `Target` merupakan jalur Amazon S3 atau Amazon EFS yang sebenarnya. Jika Anda hanya menentukan target, itu ditampilkan apa adanya. Anda juga harus memastikan bahwa peran AWS Identity and Access Management (IAM) Anda menyediakan akses ke jalur masuk`Target`. Nilai ini dapat diatur hanya ketika `HomeDirectoryType` diatur ke LOGICAL.

Berikut ini adalah contoh `Entry` dan `Target` pair.

```
[ { "Entry": "/directory1", "Target": "/bucket_name/home/
mydirectory" } ]
```

Dalam kebanyakan kasus, Anda dapat menggunakan nilai ini alih-alih kebijakan sesi untuk mengunci pengguna Anda ke direktori home yang ditunjuk (`chroot`“”). Untuk melakukan ini, Anda

dapat mengatur `Entry` ke `/` dan mengatur `Target` ke nilai yang harus dilihat pengguna untuk direktori home mereka ketika mereka masuk.

Berikut ini adalah contoh `Entry` dan `Target` pair untuk`chroot`.

```
[ { "Entry": "/", "Target": "/bucket_name/home/mydirectory" } ]
```

Tipe: Array objek HomeDirectoryMapEntry

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50000 item.

Diperlukan: Tidak

HomeDirectoryType

Jenis direktori pendaratan (folder) yang Anda inginkan direktori home pengguna Anda ketika mereka masuk ke server. Jika Anda mengaturnya`PATH`, pengguna akan melihat bucket Amazon S3 absolut atau jalur Amazon EFS seperti pada klien protokol transfer file mereka. Jika Anda menyetelnya`LOGICAL`, Anda harus menyediakan pemetaan `HomeDirectoryMappings` untuk bagaimana Anda ingin membuat jalur Amazon S3 atau Amazon EFS terlihat oleh pengguna Anda.

> **Note**
>
> Jika `HomeDirectoryType` ya`LOGICAL`, Anda harus memberikan pemetaan, menggunakan parameter. `HomeDirectoryMappings` Jika, di sisi lain, `HomeDirectoryType` adalah`PATH`, Anda memberikan jalur absolut menggunakan `HomeDirectory` parameter. Anda tidak dapat memiliki keduanya `HomeDirectory` dan `HomeDirectoryMappings` di template Anda.

Jenis: String

Nilai yang Valid: `PATH | LOGICAL`

Diperlukan: Tidak

Policy

Kebijakan sesi untuk pengguna Anda sehingga Anda dapat menggunakan peran yang sama AWS Identity and Access Management (IAM) di beberapa pengguna. Kebijakan ini mencakup akses pengguna ke sebagian bucket Amazon S3 mereka. Variabel yang dapat Anda gunakan

dalam kebijakan ini meliputi `${Transfer:UserName}`, `${Transfer:HomeDirectory}`, dan `${Transfer:HomeBucket}`.

> **Note**
>
> Kebijakan ini hanya berlaku jika domainnya `ServerId` adalah Amazon S3. Amazon EFS tidak menggunakan kebijakan sesi.
> Untuk kebijakan sesi, AWS Transfer Family menyimpan kebijakan sebagai gumpalan JSON, bukan Nama Sumber Daya Amazon (ARN) kebijakan. Anda menyimpan kebijakan sebagai blob JSON dan meneruskan dalam argumen `Policy`.
> Untuk contoh kebijakan sesi, lihat Contoh kebijakan sesi.
> Untuk informasi selengkapnya, lihat AssumeRoledi Referensi API Layanan Token AWS Keamanan.

Jenis: String

Batasan Panjang: Panjang minimum sebesar 0. Panjang maksimum 2048.

Diperlukan: Tidak

PosixProfile

Menentukan identitas POSIX lengkap, termasuk ID pengguna (`Uid`), ID grup (`Gid`), dan ID grup sekunder apa pun (`SecondaryGids`), yang mengontrol akses pengguna ke sistem file Amazon EFS Anda. Izin POSIX yang ditetapkan pada file dan direktori di Amazon EFS menentukan tingkat akses yang didapat pengguna Anda saat mentransfer file masuk dan keluar dari sistem file Amazon EFS Anda.

Tipe: Objek PosixProfile

Diperlukan: Tidak

Role

Nama Sumber Daya Amazon (ARN) dari peran AWS Identity and Access Management (IAM) yang mengontrol akses pengguna ke bucket Amazon S3 atau sistem file Amazon EFS. Kebijakan yang dilampirkan pada peran ini menentukan tingkat akses yang ingin Anda berikan kepada pengguna saat mentransfer file masuk dan keluar dari bucket Amazon S3 atau sistem file Amazon EFS. IAM role juga harus berisi hubungan kepercayaan yang mengizinkan server untuk mengakses sumber daya Anda saat melayani permintaan transfer pengguna.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 2048.

Pola: `arn:.*role/\S+`

Diperlukan: Ya

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk instans server. Ini adalah server tertentu tempat Anda menambahkan pengguna.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

SshPublicKeyBody

Bagian publik dari kunci Secure Shell (SSH) yang digunakan untuk mengautentikasi pengguna ke server.

Tiga elemen format kunci publik SSH standar adalah`<key type>`,, dan opsional `<body base64><comment>`, dengan spasi di antara setiap elemen.

AWS Transfer Family menerima kunci RSA, ECDSA, dan ED25519.

- Untuk kunci RSA, tipe kuncinya adalah`ssh-rsa`.
- Untuk tombol ED25519, jenis kuncinya adalah. `ssh-ed25519`
- Untuk kunci ECDSA, jenis kuncinya adalah`ecdsa-sha2-nistp256`,`ecdsa-sha2-nistp384`, atau`ecdsa-sha2-nistp521`, tergantung pada ukuran kunci yang Anda hasilkan.

Jenis: String

Batasan Panjang: Panjang minimum sebesar 0. Panjang maksimum 2048.

Diperlukan: Tidak

Tags

Pasangan nilai-kunci yang dapat digunakan untuk grup dan mencari pengguna. Tanda adalah metadata yang dilampirkan ke pengguna untuk tujuan apa pun.

Tipe: Array objek Tag

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50 item.

Diperlukan: Tidak

UserName

String unik yang mengidentifikasi pengguna dan dikaitkan dengan file. `ServerId` Nama pengguna ini harus sepanjang minimal 3 dan maksimal 100 karakter. Berikut adalah karakter yang valid: a-z, A-Z, 0-9, garis bawah '_', tanda hubung '-', titik '.', dan tanda at '@'. Nama pengguna tidak dapat dimulai dengan tanda hubung, titik, atau tanda at.

Jenis: String

Batasan Panjang: Panjang minimum 3. Panjang maksimum 100.

Pola: `[\w][\w@.-]{2,99}`

Diperlukan: Ya

Sintaksis Respons

```
{
   "ServerId": "string",
   "UserName": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

ServerId

Pengidentifikasi server tempat pengguna dilampirkan.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

UserName

String unik yang mengidentifikasi pengguna Transfer Family.

Jenis: String

Batasan Panjang: Panjang minimum 3. Panjang maksimum 100.

Pola: `[\w][\w@.-]{2,99}`

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilemparkan ketika kesalahan terjadi dalam AWS Transfer Family layanan.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceExistsException

Sumber daya yang diminta tidak ada, atau ada di wilayah selain yang ditentukan untuk perintah.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Contoh-contoh

Contoh

Untuk membuat pengguna, pertama-tama Anda dapat menyimpan parameter ke dalam file JSON, misalnya`createUserParameters`, lalu jalankan perintah API create-user.

```
{
    "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"
}
```

Permintaan Sampel

```
aws transfer create-user --cli-input-json file://createUserParameters
```

Contoh Respons

```
{
    "ServerId": ""s-1111aaaa2222bbbb3",
    "UserName": "bobusa-API"
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript

- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

CreateWorkflow

Memungkinkan Anda membuat alur kerja dengan langkah-langkah tertentu dan detail langkah yang dipanggil alur kerja setelah transfer file selesai. Setelah membuat alur kerja, Anda dapat mengaitkan alur kerja yang dibuat dengan server transfer apa pun dengan menentukan `workflow-details` bidang `CreateServer` dan operasi. `UpdateServer`

Sintaksis Permintaan

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

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

Description

Deskripsi tekstual untuk alur kerja.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 256.

Pola: `[\w- ]*`

Diperlukan: Tidak

OnExceptionSteps

Menentukan langkah-langkah (tindakan) untuk mengambil jika kesalahan ditemui selama pelaksanaan alur kerja.

> **Note**
>
> Untuk langkah-langkah khusus, fungsi Lambda perlu mengirim `FAILURE` ke API panggilan balik untuk memulai langkah pengecualian. Selain itu, jika Lambda tidak mengirim `SUCCESS` sebelum waktu habis, langkah-langkah pengecualian dijalankan.

Tipe: Array objek WorkflowStep

Anggota Array: Jumlah minimum 0 item. Jumlah maksimum 8 item.

Diperlukan: Tidak

Steps

Menentukan rincian untuk langkah-langkah yang ada dalam alur kerja yang ditentukan.

TYPEMenentukan mana dari tindakan berikut yang sedang diambil untuk langkah ini.

- **`COPY`**- Salin file ke lokasi lain.

- **CUSTOM**- Lakukan langkah khusus dengan target AWS Lambda fungsi.
- **DECRYPT**- Dekripsi file yang dienkripsi sebelum diunggah.
- **DELETE**- Hapus file.
- **TAG**- Tambahkan tag ke file.

> **Note**
>
> Saat ini, penyalinan dan penandaan hanya didukung pada S3.

Untuk lokasi file, Anda menentukan bucket dan kunci Amazon S3, atau ID dan jalur sistem file Amazon EFS.

Tipe: Array objek WorkflowStep

Anggota Array: Jumlah minimum 0 item. Jumlah maksimum 8 item.

Diperlukan: Ya

Tags

Pasangan nilai kunci yang dapat digunakan untuk mengelompokkan dan mencari alur kerja. Tag adalah metadata yang dilampirkan ke alur kerja untuk tujuan apa pun.

Tipe: Array objek Tag

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50 item.

Diperlukan: Tidak

Sintaksis Respons

```
{
   "WorkflowId": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

WorkflowId

Pengidentifikasi unik untuk alur kerja.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `w-([a-z0-9]{17})`

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

AccessDeniedException

Anda tidak memiliki akses yang memadai untuk melakukan tindakan ini.

Kode Status HTTP: 400

InternalServiceError

Pengecualian ini dilemparkan ketika kesalahan terjadi dalam AWS Transfer Family layanan.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceExistsException

Sumber daya yang diminta tidak ada, atau ada di wilayah selain yang ditentukan untuk perintah.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

ThrottlingException

Permintaan ditolak karena throttling permintaan.

Kode Status HTTP: 400

Contoh-contoh

Contoh

Anda dapat menyimpan informasi langkah alur kerja ke dalam file teks, dan kemudian menggunakan file itu untuk membuat alur kerja, seperti pada contoh berikut. Contoh berikut mengasumsikan Anda telah menyimpan langkah-langkah alur kerja Anda ke *example-file.json* (dalam folder yang sama dari tempat Anda menjalankan perintah), dan bahwa Anda ingin membuat alur kerja di wilayah Virginia N. (us-east-1).

```
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}"
      }
    }
]
```

Contoh

`CreateWorkflow`Panggilan mengembalikan ID alur kerja untuk alur kerja baru.

Contoh Respons

```
{
    "WorkflowId": "w-1234abcd5678efghi"
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DeleteAccess

Memungkinkan Anda menghapus akses yang ditentukan dalam `ExternalID` parameter `ServerID` dan.

Sintaksis Permintaan

```
{
   "ExternalId": "string",
   "ServerId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

ExternalId

Pengidentifikasi unik yang diperlukan untuk mengidentifikasi grup tertentu dalam direktori Anda. Pengguna grup yang Anda asosiasikan memiliki akses ke sumber daya Amazon S3 atau Amazon EFS Anda melalui protokol yang diaktifkan. AWS Transfer Family Jika Anda tahu nama grup, Anda dapat melihat nilai SID dengan menjalankan perintah berikut menggunakan Windows PowerShell.

```
Get-ADGroup -Filter {samAccountName -like "YourGroupName*"} -Properties
* | Select SamAccountName,ObjectSid
```

Dalam perintah itu, ganti YourGroupNamedengan nama grup Active Directory Anda.

Ekspresi reguler yang digunakan untuk memvalidasi parameter ini adalah string karakter yang terdiri dari huruf besar dan huruf kecil karakter alfanumerik tanpa spasi. Anda juga dapat menyertakan garis bawah atau salah satu karakter berikut: =, . @: /-

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `S-1-[\d-]+`

Diperlukan: Ya

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk server yang telah ditetapkan pengguna ini.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteAgreement

Hapus perjanjian yang ditentukan dalam yang disediakan`AgreementId`.

Sintaksis Permintaan

```
{
   "AgreementId": "string",
   "ServerId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

AgreementId

Pengidentifikasi unik untuk perjanjian. Pengenal ini dikembalikan saat Anda membuat perjanjian.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `a-([0-9a-f]{17})`

Diperlukan: Ya

ServerId

Pengenal server yang terkait dengan perjanjian yang Anda hapus.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3

- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteCertificate

Menghapus sertifikat yang ditentukan dalam `CertificateId` parameter.

Sintaksis Permintaan

```
{
   "CertificateId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

CertificateId

Pengidentifikasi objek sertifikat yang Anda hapus.

Jenis: String

Kendala Panjang: Panjang tetap 22.

Pola: `cert-([0-9a-f]{17})`

Diperlukan: Ya

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteConnector

Menghapus konektor yang ditentukan dalam yang disediakan`ConnectorId`.

Sintaksis Permintaan

```
{
   "ConnectorId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

ConnectorId

Pengidentifikasi unik untuk konektor.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `c-([0-9a-f]{17})`

Diperlukan: Ya

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteHostKey

Menghapus kunci host yang ditentukan dalam `HostKeyId` parameter.

Sintaksis Permintaan

```
{
   "HostKeyId": "string",
   "ServerId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

HostKeyId

Pengidentifikasi kunci host yang Anda hapus.

Jenis: String

Kendala Panjang: Panjang tetap 25.

Pola: `hostkey-[0-9a-f]{17}`

Diperlukan: Ya

ServerId

Pengidentifikasi server yang berisi kunci host yang Anda hapus.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

ThrottlingException

Permintaan ditolak karena throttling permintaan.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.

- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteProfile

Menghapus profil yang ditentukan dalam `ProfileId` parameter.

Sintaksis Permintaan

```
{
   "ProfileId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

ProfileId

Pengidentifikasi profil yang Anda hapus.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `p-([0-9a-f]{17})`

Diperlukan: Ya

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteServer

Menghapus server berkemampuan protokol transfer file yang Anda tentukan.

Tidak ada tanggapan yang kembali dari operasi ini.

Sintaksis Permintaan

```
{
   "ServerId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk instance server.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

AccessDeniedException

Anda tidak memiliki akses yang memadai untuk melakukan tindakan ini.

Kode Status HTTP: 400

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Contoh-contoh

Contoh

Contoh berikut menghapus server.

Permintaan Sampel

```
{
   "ServerId": "s-01234567890abcdef"
}
```

Contoh

Jika berhasil, tidak ada yang dikembalikan.

Contoh Respons

```
{
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteSshPublicKey

Menghapus kunci publik Secure Shell (SSH) pengguna.

Sintaksis Permintaan

```
{
   "ServerId": "string",
   "SshPublicKeyId": "string",
   "UserName": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk instance server berkemampuan protokol transfer file yang memiliki pengguna yang ditugaskan padanya.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

SshPublicKeyId

Pengenal unik yang digunakan untuk mereferensikan kunci SSH spesifik pengguna Anda.

Jenis: String

Kendala Panjang: Panjang tetap 21.

Pola: `key-[0-9a-f]{17}`

Diperlukan: Ya

UserName

String unik yang mengidentifikasi pengguna yang kunci publiknya sedang dihapus.

Jenis: String

Batasan Panjang: Panjang minimum 3. Panjang maksimum 100.

Pola: `[\w][\w@.-]{2,99}`

Diperlukan: Ya

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

ThrottlingException

Permintaan ditolak karena throttling permintaan.

Kode Status HTTP: 400

Contoh-contoh

Contoh

Contoh berikut menghapus kunci publik SSH pengguna.

Permintaan Sampel

```
{
   "ServerId": "s-01234567890abcdef",
   "SshPublicKeyId": "MyPublicKey",
   "UserName": "my_user"
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteUser

Menghapus pengguna milik server berkemampuan protokol transfer file yang Anda tentukan.

Tidak ada tanggapan yang kembali dari operasi ini.

> **Note**
>
> Ketika Anda menghapus pengguna dari server, informasi pengguna hilang.

Sintaksis Permintaan

```
{
   "ServerId": "string",
   "UserName": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk instance server yang memiliki pengguna yang ditugaskan padanya.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

UserName

String unik yang mengidentifikasi pengguna yang sedang dihapus dari server.

Jenis: String

Batasan Panjang: Panjang minimum 3. Panjang maksimum 100.

Pola: `[\w][\w@.-]{2,99}`

Diperlukan: Ya

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Contoh-contoh

Contoh

Contoh berikut menghapus pengguna Transfer Family.

Permintaan Sampel

```
{
   "ServerId": "s-01234567890abcdef",
   "UserNames": "my_user"
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DeleteWorkflow

Menghapus alur kerja yang ditentukan.

Sintaksis Permintaan

```
{
   "WorkflowId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

WorkflowId

Pengidentifikasi unik untuk alur kerja.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `w-([a-z0-9]{17})`

Diperlukan: Ya

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

AccessDeniedException

Anda tidak memiliki akses yang memadai untuk melakukan tindakan ini.

Kode Status HTTP: 400

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeAccess

Menjelaskan akses yang ditetapkan ke server berkemampuan protokol transfer file tertentu, seperti yang diidentifikasi oleh `ServerId` propertinya dan miliknya. `ExternalId`

Respons dari panggilan ini mengembalikan properti akses yang terkait dengan `ServerId` nilai yang ditentukan.

Sintaksis Permintaan

```
{
   "ExternalId": "string",
   "ServerId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

ExternalId

Pengidentifikasi unik yang diperlukan untuk mengidentifikasi grup tertentu dalam direktori Anda. Pengguna grup yang Anda kaitkan memiliki akses ke sumber daya Amazon S3 atau Amazon EFS Anda melalui protokol yang diaktifkan. AWS Transfer Family Jika Anda tahu nama grup, Anda dapat melihat nilai SID dengan menjalankan perintah berikut menggunakan Windows PowerShell.

```
Get-ADGroup -Filter {samAccountName -like "YourGroupName*"} -Properties
* | Select SamAccountName,ObjectSid
```

Dalam perintah itu, ganti YourGroupNamedengan nama grup Active Directory Anda.

Ekspresi reguler yang digunakan untuk memvalidasi parameter ini adalah string karakter yang terdiri dari huruf besar dan huruf kecil karakter alfanumerik tanpa spasi. Anda juga dapat menyertakan garis bawah atau salah satu karakter berikut: =, . @: /-

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `S-1-[\d-]+`

Diperlukan: Ya

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk server yang memiliki akses ini ditetapkan.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

Sintaksis Respons

```
{
   "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"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

Access

Pengidentifikasi eksternal server tempat akses dilampirkan.

Tipe: Objek DescribedAccess

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk server yang memiliki akses ini ditetapkan.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeAgreement

Menjelaskan perjanjian yang diidentifikasi olehAgreementId.

Sintaksis Permintaan

```
{
   "AgreementId": "string",
   "ServerId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

AgreementId

Pengidentifikasi unik untuk perjanjian. Pengenal ini dikembalikan saat Anda membuat perjanjian.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: a-([0-9a-f]{17})

Diperlukan: Ya

ServerId

Pengidentifikasi server yang terkait dengan perjanjian.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: s-([0-9a-f]{17})

Diperlukan: Ya

Sintaksis Respons

```
{
```

```
   "Agreement": {
      "AccessRole": "string",
      "AgreementId": "string",
      "Arn": "string",
      "BaseDirectory": "string",
      "Description": "string",
      "LocalProfileId": "string",
      "PartnerProfileId": "string",
      "ServerId": "string",
      "Status": "string",
      "Tags": [
         {
            "Key": "string",
            "Value": "string"
         }
      ]
   }
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

Agreement

Rincian untuk perjanjian yang ditentukan, dikembalikan sebagai `DescribedAgreement` objek.

Tipe: Objek DescribedAgreement

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeCertificate

Menjelaskan sertifikat yang diidentifikasi olehCertificateId.

Sintaksis Permintaan

```
{
   "CertificateId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

CertificateId

Array pengidentifikasi untuk sertifikat yang diimpor. Anda menggunakan pengenal ini untuk bekerja dengan profil dan profil mitra.

Jenis: String

Kendala Panjang: Panjang tetap 22.

Pola: `cert-([0-9a-f]{17})`

Diperlukan: Ya

Sintaksis Respons

```
{
   "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"
   }
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

Certificate

Rincian untuk sertifikat yang ditentukan, dikembalikan sebagai objek.

Tipe: Objek DescribedCertificate

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeConnector

Menjelaskan konektor yang diidentifikasi oleh `ConnectorId`.

Sintaksis Permintaan

```
{
   "ConnectorId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

ConnectorId

Pengidentifikasi unik untuk konektor.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `c-([0-9a-f]{17})`

Diperlukan: Ya

Sintaksis Respons

```
{
   "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"
   }
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

Connector

Struktur yang berisi rincian konektor.

Tipe: Objek DescribedConnector

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilemparkan ketika kesalahan terjadi dalam AWS Transfer Family layanan.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DescribeExecution

Anda dapat menggunakan `DescribeExecution` untuk memeriksa rincian eksekusi alur kerja yang ditentukan.

> **Note**
>
> Panggilan API ini hanya menampilkan detail untuk alur kerja yang sedang berlangsung. Jika Anda memberikan ID untuk eksekusi yang tidak sedang berlangsung, atau jika eksekusi tidak cocok dengan ID alur kerja yang ditentukan, Anda akan menerima `ResourceNotFound` pengecualian.

Sintaksis Permintaan

```
{
   "ExecutionId": "string",
   "WorkflowId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

ExecutionId

Pengidentifikasi unik untuk eksekusi alur kerja.

Jenis: String

Batas Panjang: Panjang tetap 36.

Pola: `[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}`

Diperlukan: Ya

WorkflowId

Pengidentifikasi unik untuk alur kerja.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `w-([a-z0-9]{17})`

Diperlukan: Ya

Sintaksis Respons

```
{
   "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"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

Execution

Struktur yang berisi rincian eksekusi alur kerja.

Tipe: Objek DescribedExecution

WorkflowId

Pengidentifikasi unik untuk alur kerja.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `w-([a-z0-9]{17})`

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2

- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeHostKey

Mengembalikan rincian kunci host yang ditentukan oleh `HostKeyId` dan`ServerId`.

Sintaksis Permintaan

```
{
   "HostKeyId": "string",
   "ServerId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

HostKeyId

Pengidentifikasi kunci host yang ingin Anda jelaskan.

Jenis: String

Kendala Panjang: Panjang tetap 25.

Pola: `hostkey-[0-9a-f]{17}`

Diperlukan: Ya

ServerId

Pengidentifikasi server yang berisi kunci host yang ingin Anda jelaskan.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

Sintaksis Respons

```
{
```

```
   "HostKey": {
      "Arn": "string",
      "DateImported": number,
      "Description": "string",
      "HostKeyFingerprint": "string",
      "HostKeyId": "string",
      "Tags": [
         {
            "Key": "string",
            "Value": "string"
         }
      ],
      "Type": "string"
   }
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

HostKey

Mengembalikan rincian untuk kunci host yang ditentukan.

Tipe: Objek DescribedHostKey

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeProfile

Mengembalikan rincian profil yang ditentukan oleh`ProfileId`.

Sintaksis Permintaan

```
{
   "ProfileId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

ProfileId

Pengidentifikasi profil yang ingin Anda jelaskan.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `p-([0-9a-f]{17})`

Diperlukan: Ya

Sintaksis Respons

```
{
   "Profile": {
      "Arn": "string",
      "As2Id": "string",
      "CertificateIds": [ "string" ],
      "ProfileId": "string",
      "ProfileType": "string",
      "Tags": [
         {
            "Key": "string",
            "Value": "string"
         }
```

```
        ]
    }
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

Profile

Rincian profil yang ditentukan, dikembalikan sebagai objek.

Tipe: Objek DescribedProfile

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeSecurityPolicy

Menjelaskan kebijakan keamanan yang dilampirkan ke server atau konektor SFTP Anda. Respons berisi deskripsi properti kebijakan keamanan. Untuk informasi selengkapnya tentang kebijakan keamanan, lihat Bekerja dengan kebijakan keamanan untuk server atau Bekerja dengan kebijakan keamanan untuk konektor SFTP.

Sintaksis Permintaan

```
{
   "SecurityPolicyName": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

SecurityPolicyName

Tentukan nama teks kebijakan keamanan yang Anda inginkan detailnya.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 100.

Pola: `Transfer[A-Za-z0-9]*SecurityPolicy-[A-Za-z0-9-]+`

Diperlukan: Ya

Sintaksis Respons

```
{
   "SecurityPolicy": {
      "Fips": boolean,
      "Protocols": [ "string" ],
      "SecurityPolicyName": "string",
      "SshCiphers": [ "string" ],
      "SshHostKeyAlgorithms": [ "string" ],
      "SshKexs": [ "string" ],
```

```
        "SshMacs": [ "string" ],
        "TlsCiphers": [ "string" ],
        "Type": "string"
    }
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

SecurityPolicy

Array yang berisi properti kebijakan keamanan.

Tipe: Objek DescribedSecurityPolicy

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilemparkan ketika kesalahan terjadi dalam AWS Transfer Family layanan.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Contoh-contoh

Contoh

Perintah contoh berikut mengambil nama kebijakan keamanan sebagai argumen, dan mengembalikan algoritme untuk kebijakan keamanan yang ditentukan.

Permintaan Sampel

```
aws transfer describe-security-policy --security-policy-name "TransferSecurityPolicy-
FIPS-2023-05"
```

Contoh Respons

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

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

DescribeServer

Menjelaskan server berkemampuan protokol transfer file yang Anda tentukan dengan meneruskan parameter. `ServerId`

Respons berisi deskripsi properti server. Saat Anda mengatur `EndpointType` ke VPC, respons akan berisi file. `EndpointDetails`

Sintaksis Permintaan

```
{
   "ServerId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk server.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

Sintaksis Respons

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

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

Server

Array yang berisi properti server dengan yang `ServerID` Anda tentukan.

Tipe: Objek DescribedServer

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Contoh-contoh

Contoh

Contoh berikut mengembalikan properti yang ditugaskan ke server.

Permintaan Sampel

```
{
   "ServerId": "s-01234567890abcdef"
}
```

Contoh

Contoh ini menggambarkan salah satu penggunaan. DescribeServer

Contoh Respons

```
{
    "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
    }
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeUser

Menjelaskan pengguna yang ditugaskan ke server berkemampuan protokol transfer file tertentu, seperti yang diidentifikasi oleh propertinya. `ServerId`

Respons dari panggilan ini mengembalikan properti pengguna yang terkait dengan `ServerId` nilai yang ditentukan.

Sintaksis Permintaan

```
{
   "ServerId": "string",
   "UserName": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk server yang telah ditetapkan pengguna ini.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

UserName

Nama pengguna yang ditetapkan ke satu atau lebih server. Nama pengguna adalah bagian dari kredensi masuk untuk menggunakan AWS Transfer Family layanan dan melakukan tugas transfer file.

Jenis: String

Batasan Panjang: Panjang minimum 3. Panjang maksimum 100.

Pola: `[\w][\w@.-]{2,99}`

Diperlukan: Ya

Sintaksis Respons

```
{
   "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"
   }
```

```
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk server yang telah ditetapkan pengguna ini.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

User

Array yang berisi properti pengguna Transfer Family untuk `ServerID` nilai yang Anda tentukan.

Tipe: Objek DescribedUser

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Contoh-contoh

Contoh

Contoh berikut menunjukkan rincian untuk pengguna yang ada.

Permintaan Sampel

```
aws transfer describe-user --server-id s-1111aaaa2222bbbb3 --user-name bob-test
```

Contoh Respons

```
{
    "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"
    }
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

DescribeWorkflow

Menjelaskan alur kerja yang ditentukan.

Sintaksis Permintaan

```
{
   "WorkflowId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

WorkflowId

Pengidentifikasi unik untuk alur kerja.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `w-([a-z0-9]{17})`

Diperlukan: Ya

Sintaksis Respons

```
{
   "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"
   }
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

Workflow

Struktur yang berisi rincian alur kerja.

Tipe: Objek DescribedWorkflow

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ImportCertificate

Mengimpor sertifikat penandatanganan dan enkripsi yang Anda perlukan untuk membuat profil lokal (AS2) dan profil mitra.

Sintaksis Permintaan

```
{
   "ActiveDate": number,
   "Certificate": "string",
   "CertificateChain": "string",
   "Description": "string",
   "InactiveDate": number,
   "PrivateKey": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ],
   "Usage": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

ActiveDate

Tanggal opsional yang menentukan kapan sertifikat menjadi aktif.

Tipe: Timestamp

Diperlukan: Tidak

Certificate

- Untuk CLI, berikan jalur file untuk sertifikat dalam format URI. Misalnya, `--certificate file://encryption-cert.pem`. Atau, Anda dapat memberikan konten mentah.
- Untuk SDK, tentukan konten mentah dari file sertifikat. Sebagai contoh, `--certificate "`cat encryption-cert.pem`"`.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 16384.

Pola: `[\u0009\u000A\u000D\u0020-\u00FF]*`

Diperlukan: Ya

CertificateChain

Daftar opsional sertifikat yang membentuk rantai untuk sertifikat yang sedang diimpor.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 2097152.

Pola: `[\u0009\u000A\u000D\u0020-\u00FF]*`

Diperlukan: Tidak

Description

Deskripsi singkat yang membantu mengidentifikasi sertifikat.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 200.

Pola: `[\p{Graph}]+`

Diperlukan: Tidak

InactiveDate

Tanggal opsional yang menentukan kapan sertifikat menjadi tidak aktif.

Tipe: Timestamp

Diperlukan: Tidak

PrivateKey

- Untuk CLI, berikan jalur file untuk kunci pribadi dalam format URI. Misalnya,. `--private-key file://encryption-key.pem` Atau, Anda dapat memberikan konten mentah dari file kunci pribadi.

- Untuk SDK, tentukan konten mentah dari file kunci pribadi. Misalnya, `--private-key "`cat encryption-key.pem`"`

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 16384.

Pola: `[\u0009\u000A\u000D\u0020-\u00FF]*`

Diperlukan: Tidak

Tags

Pasangan nilai kunci yang dapat digunakan untuk mengelompokkan dan mencari sertifikat.

Tipe: Array objek Tag

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50 item.

Diperlukan: Tidak

Usage

Menentukan bagaimana sertifikat ini digunakan. Ini dapat digunakan dengan cara-cara berikut:

- `SIGNING`: Untuk menandatangani pesan AS2
- `ENCRYPTION`: Untuk mengenkripsi pesan AS2
- `TLS`: Untuk mengamankan komunikasi AS2 yang dikirim melalui HTTPS

Jenis: String

Nilai yang Valid: `SIGNING | ENCRYPTION`

Diperlukan: Ya

Sintaksis Respons

```
{
   "CertificateId": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

CertificateId

Array pengidentifikasi untuk sertifikat yang diimpor. Anda menggunakan pengenal ini untuk bekerja dengan profil dan profil mitra.

Jenis: String

Kendala Panjang: Panjang tetap 22.

Pola: `cert-([0-9a-f]{17})`

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilemparkan ketika kesalahan terjadi dalam AWS Transfer Family layanan.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Contoh-contoh

Contoh

Contoh berikut mengimpor sertifikat yang akan digunakan untuk enkripsi. Pada perintah pertama, kami menyediakan konten file sertifikat dan rantai sertifikat. Gunakan format ini untuk perintah SDK.

```
aws transfer import-certificate --usage ENCRYPTION --certificate "`cat encryption-
cert.pem`" \
    --private-key "`cat encryption-key.pem`" --certificate-chain "`cat root-ca.pem`"
```

Contoh

Contoh berikut identik dengan perintah sebelumnya, kecuali bahwa kami menyediakan lokasi file untuk kunci pribadi, sertifikat, dan file rantai sertifikat. Versi perintah ini tidak berfungsi jika Anda menggunakan SDK.

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

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

ImportHostKey

Menambahkan kunci host ke server yang ditentukan oleh `ServerId` parameter.

Sintaksis Permintaan

```
{
   "Description": "string",
   "HostKeyBody": "string",
   "ServerId": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

Description

Deskripsi teks yang mengidentifikasi kunci host ini.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 200.

Pola: `[\p{Print}]*`

Diperlukan: Tidak

HostKeyBody

Bagian kunci pribadi dari sebuah key pair SSH.

AWS Transfer Familymenerima kunci RSA, ECDSA, dan ED25519.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 4096.

Diperlukan: Ya

ServerId

Pengidentifikasi server yang berisi kunci host yang Anda impor.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

Tags

Pasangan kunci-nilai yang dapat digunakan untuk mengelompokkan dan mencari kunci host.

Tipe: Array objek Tag

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50 item.

Diperlukan: Tidak

Sintaksis Respons

```
{
   "HostKeyId": "string",
   "ServerId": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

HostKeyId

Mengembalikan pengenal kunci host untuk kunci yang diimpor.

Jenis: String

Kendala Panjang: Panjang tetap 25.

Pola: `hostkey-[0-9a-f]{17}`

ServerId

Mengembalikan pengenal server yang berisi kunci yang diimpor.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceExistsException

Sumber daya yang diminta tidak ada, atau ada di wilayah selain yang ditentukan untuk perintah.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

ThrottlingException

Permintaan ditolak karena throttling permintaan.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ImportSshPublicKey

Menambahkan kunci publik Secure Shell (SSH) ke pengguna Transfer Family yang diidentifikasi oleh `UserName` nilai yang ditetapkan ke server berkemampuan protokol transfer file tertentu, yang diidentifikasi oleh. `ServerId`

Respons mengembalikan `UserName` nilai, `ServerId` nilai, dan nama`SshPublicKeyId`.

Sintaksis Permintaan

```
{
   "ServerId": "string",
   "SshPublicKeyBody": "string",
   "UserName": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk server.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

SshPublicKeyBody

Bagian kunci publik dari SSH key pair.

AWS Transfer Family menerima kunci RSA, ECDSA, dan ED25519.

Jenis: String

Batasan Panjang: Panjang minimum sebesar 0. Panjang maksimum 2048.

Diperlukan: Ya

UserName

Nama pengguna Transfer Family yang ditetapkan ke satu atau beberapa server.

Jenis: String

Batasan Panjang: Panjang minimum 3. Panjang maksimum 100.

Pola: `[\w][\w@.-]{2,99}`

Diperlukan: Ya

Sintaksis Respons

```
{
   "ServerId": "string",
   "SshPublicKeyId": "string",
   "UserName": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk server.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

SshPublicKeyId

Nama yang diberikan untuk kunci publik oleh sistem yang diimpor.

Jenis: String

Kendala Panjang: Panjang tetap 21.

Pola: `key-[0-9a-f]{17}`

UserName

Nama pengguna yang ditetapkan ke `ServerID` nilai yang Anda tentukan.

Jenis: String

Batasan Panjang: Panjang minimum 3. Panjang maksimum 100.

Pola: `[\w][\w@.-]{2,99}`

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilemparkan ketika kesalahan terjadi dalam AWS Transfer Family layanan.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceExistsException

Sumber daya yang diminta tidak ada, atau ada di wilayah selain yang ditentukan untuk perintah.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

ThrottlingException

Permintaan ditolak karena throttling permintaan.

Kode Status HTTP: 400

Contoh-contoh

Contoh

Perintah ini mengimpor kunci ECDSA yang disimpan dalam file. `id_ecdsa.pub`

```
aws transfer import-ssh-public-key --server-id s-021345abcdef6789 --ssh-public-key-body
 file://id_ecdsa.pub --user-name jane-doe
```

Contoh

Jika Anda menjalankan perintah sebelumnya, sistem mengembalikan informasi berikut.

```
{
    "ServerId": "s-021345abcdef6789",
   "SshPublicKeyId": "key-1234567890abcdef0",
   "UserName": "jane-doe"
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3

- AWS SDK untuk Python
- AWS SDK for Ruby V3

ListAccesses

Daftar detail untuk semua akses yang Anda miliki di server Anda.

Sintaksis Permintaan

```
{
   "MaxResults": number,
   "NextToken": "string",
   "ServerId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

MaxResults

Menentukan jumlah maksimum akses SID untuk kembali.

Jenis: Integer

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 1000.

Diperlukan: Tidak

NextToken

Ketika Anda bisa mendapatkan hasil tambahan dari `ListAccesses` panggilan, `NextToken` parameter dikembalikan dalam output. Anda kemudian dapat meneruskan perintah berikutnya ke `NextToken` parameter untuk melanjutkan daftar akses tambahan.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 6144.

Diperlukan: Tidak

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk server yang memiliki pengguna yang ditugaskan padanya.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

Sintaksis Respons

```
{
   "Accesses": [
      {
         "ExternalId": "string",
         "HomeDirectory": "string",
         "HomeDirectoryType": "string",
         "Role": "string"
      }
   ],
   "NextToken": "string",
   "ServerId": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

Accesses

Mengembalikan akses dan properti mereka untuk `ServerId` nilai yang Anda tentukan.

Tipe: Array objek ListedAccess

NextToken

Ketika Anda bisa mendapatkan hasil tambahan dari `ListAccesses` panggilan, `NextToken` parameter dikembalikan dalam output. Anda kemudian dapat meneruskan perintah berikutnya ke `NextToken` parameter untuk melanjutkan daftar akses tambahan.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 6144.

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk server yang memiliki pengguna yang ditugaskan padanya.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidNextTokenException

`NextToken`Parameter yang dilewatkan tidak valid.

Kode Status HTTP: 400

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListAgreements

Mengembalikan daftar perjanjian untuk server yang diidentifikasi oleh `ServerId` yang Anda berikan. Jika Anda ingin membatasi hasil ke angka tertentu, berikan nilai untuk `MaxResults` parameter tersebut. Jika Anda menjalankan perintah sebelumnya dan menerima nilai untuk`NextToken`, Anda dapat memberikan nilai tersebut untuk melanjutkan daftar perjanjian dari tempat Anda tinggalkan.

Sintaksis Permintaan

```
{
   "MaxResults": number,
   "NextToken": "string",
   "ServerId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

MaxResults

Jumlah maksimum perjanjian untuk dikembalikan.

Jenis: Integer

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 1000.

Diperlukan: Tidak

NextToken

Ketika Anda bisa mendapatkan hasil tambahan dari `ListAgreements` panggilan, `NextToken` parameter dikembalikan dalam output. Anda kemudian dapat meneruskan perintah berikutnya ke `NextToken` parameter untuk melanjutkan daftar perjanjian tambahan.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 6144.

Diperlukan: Tidak

ServerId

Pengidentifikasi server yang Anda inginkan daftar perjanjian.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

Sintaksis Respons

```
{
   "Agreements": [
      {
         "AgreementId": "string",
         "Arn": "string",
         "Description": "string",
         "LocalProfileId": "string",
         "PartnerProfileId": "string",
         "ServerId": "string",
         "Status": "string"
      }
   ],
   "NextToken": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

Agreements

Mengembalikan array, di mana setiap item berisi rincian perjanjian.

Tipe: Array objek ListedAgreement

NextToken

Mengembalikan token yang dapat Anda gunakan untuk menelepon `ListAgreements` lagi dan menerima hasil tambahan, jika ada.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 6144.

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidNextTokenException

`NextToken`Parameter yang dilewatkan tidak valid.

Kode Status HTTP: 400

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListCertificates

Mengembalikan daftar sertifikat saat ini yang telah diimpor keAWS Transfer Family. Jika Anda ingin membatasi hasil ke angka tertentu, berikan nilai untuk `MaxResults` parameter tersebut. Jika Anda menjalankan perintah sebelumnya dan menerima nilai untuk `NextToken` parameter, Anda dapat memberikan nilai tersebut untuk melanjutkan daftar sertifikat dari tempat Anda tinggalkan.

Sintaksis Permintaan

```
{
   "MaxResults": number,
   "NextToken": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

MaxResults

Jumlah maksimum sertifikat untuk dikembalikan.

Jenis: Integer

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 1000.

Diperlukan: Tidak

NextToken

Ketika Anda bisa mendapatkan hasil tambahan dari `ListCertificates` panggilan, `NextToken` parameter dikembalikan dalam output. Anda kemudian dapat meneruskan perintah berikutnya ke `NextToken` parameter untuk melanjutkan daftar sertifikat tambahan.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 6144.

Diperlukan: Tidak

Sintaksis Respons

```
{
   "Certificates": [
      {
         "ActiveDate": number,
         "Arn": "string",
         "CertificateId": "string",
         "Description": "string",
         "InactiveDate": number,
         "Status": "string",
         "Type": "string",
         "Usage": "string"
      }
   ],
   "NextToken": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

Certificates

Mengembalikan array sertifikat yang ditentukan dalam `ListCertificates` panggilan.

Tipe: Array objek ListedCertificate

NextToken

Mengembalikan token berikutnya, yang dapat Anda gunakan untuk daftar sertifikat berikutnya.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 6144.

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidNextTokenException

`NextToken`Parameter yang dilewatkan tidak valid.

Kode Status HTTP: 400

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript

- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListConnectors

Daftar konektor untuk Wilayah yang ditentukan.

Sintaksis Permintaan

```
{
   "MaxResults": number,
   "NextToken": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

MaxResults

Jumlah maksimum konektor untuk kembali.

Jenis: Integer

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 1000.

Diperlukan: Tidak

NextToken

Ketika Anda bisa mendapatkan hasil tambahan dari `ListConnectors` panggilan, `NextToken` parameter dikembalikan dalam output. Anda kemudian dapat meneruskan perintah berikutnya ke `NextToken` parameter untuk melanjutkan daftar konektor tambahan.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 6144.

Diperlukan: Tidak

Sintaksis Respons

```
{
```

```
   "Connectors": [
      {
         "Arn": "string",
         "ConnectorId": "string",
         "Url": "string"
      }
   ],
   "NextToken": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

Connectors

Mengembalikan array, di mana setiap item berisi rincian konektor.

Tipe: Array objek ListedConnector

NextToken

Mengembalikan token yang dapat Anda gunakan untuk menelepon `ListConnectors` lagi dan menerima hasil tambahan, jika ada.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 6144.

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidNextTokenException

`NextToken`Parameter yang dilewatkan tidak valid.

Kode Status HTTP: 400

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListExecutions

Daftar semua eksekusi yang sedang berlangsung untuk alur kerja yang ditentukan.

> **Note**
>
> Jika ID alur kerja yang ditentukan tidak dapat ditemukan, `ListExecutions` mengembalikan `ResourceNotFound` pengecualian.

Sintaksis Permintaan

```
{
   "MaxResults": number,
   "NextToken": "string",
   "WorkflowId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

MaxResults

Menentukan jumlah maksimum eksekusi untuk kembali.

Jenis: Integer

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 1000.

Diperlukan: Tidak

NextToken

`ListExecutions`mengembalikan `NextToken` parameter dalam output. Anda kemudian dapat meneruskan `NextToken` parameter dalam perintah berikutnya untuk melanjutkan daftar eksekusi tambahan.

Ini berguna untuk pagination, misalnya. Jika Anda memiliki 100 eksekusi untuk alur kerja, Anda mungkin hanya ingin daftar pertama 10. Jika demikian, panggil API dengan menentukan: `max-results`

```
aws transfer list-executions --max-results 10
```

Ini mengembalikan rincian untuk 10 eksekusi pertama, serta pointer (`NextToken`) ke eksekusi kesebelas. Sekarang Anda dapat memanggil API lagi, memberikan `NextToken` nilai yang Anda terima:

```
aws transfer list-executions --max-results 10 --next-token
$somePointerReturnedFromPreviousListResult
```

Panggilan ini mengembalikan 10 eksekusi berikutnya, yang ke-11 hingga ke-20. Anda kemudian dapat mengulangi panggilan sampai rincian untuk semua 100 eksekusi telah dikembalikan.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 6144.

Diperlukan: Tidak

WorkflowId

Pengidentifikasi unik untuk alur kerja.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `w-([a-z0-9]{17})`

Diperlukan: Ya

Sintaksis Respons

```
{
   "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"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

Executions

Mengembalikan rincian untuk setiap eksekusi, dalam `ListedExecution` array.

Tipe: Array objek ListedExecution

NextToken

`ListExecutions`mengembalikan `NextToken` parameter dalam output. Anda kemudian dapat meneruskan `NextToken` parameter dalam perintah berikutnya untuk melanjutkan daftar eksekusi tambahan.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 6144.

WorkflowId

Pengidentifikasi unik untuk alur kerja.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `w-([a-z0-9]{17})`

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidNextTokenException

`NextToken`Parameter yang dilewatkan tidak valid.

Kode Status HTTP: 400

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListHostKeys

Mengembalikan daftar kunci host untuk server yang ditentukan oleh `ServerId` parameter.

Sintaksis Permintaan

```
{
   "MaxResults": number,
   "NextToken": "string",
   "ServerId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

MaxResults

Jumlah maksimum kunci host untuk kembali.

Jenis: Integer

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 1000.

Diperlukan: Tidak

NextToken

Ketika ada hasil tambahan yang tidak dikembalikan, `NextToken` parameter dikembalikan. Anda dapat menggunakan nilai tersebut untuk panggilan berikutnya `ListHostKeys` untuk melanjutkan hasil daftar.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 6144.

Diperlukan: Tidak

ServerId

Pengidentifikasi server yang berisi kunci host yang ingin Anda lihat.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

Sintaksis Respons

```
{
   "HostKeys": [
      {
         "Arn": "string",
         "DateImported": number,
         "Description": "string",
         "Fingerprint": "string",
         "HostKeyId": "string",
         "Type": "string"
      }
   ],
   "NextToken": "string",
   "ServerId": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

HostKeys

Mengembalikan array, di mana setiap item berisi rincian kunci host.

Tipe: Array objek ListedHostKey

NextToken

Mengembalikan token yang dapat Anda gunakan untuk menelepon `ListHostKeys` lagi dan menerima hasil tambahan, jika ada.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 6144.

ServerId

Mengembalikan pengenal server yang berisi kunci host yang terdaftar.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidNextTokenException

`NextToken`Parameter yang dilewatkan tidak valid.

Kode Status HTTP: 400

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListProfiles

Mengembalikan daftar profil untuk sistem Anda. Jika Anda ingin membatasi hasil ke angka tertentu, berikan nilai untuk `MaxResults` parameter tersebut. Jika Anda menjalankan perintah sebelumnya dan menerima nilai untuk`NextToken`, Anda dapat memberikan nilai itu untuk melanjutkan daftar profil dari tempat Anda tinggalkan.

Sintaksis Permintaan

```
{
   "MaxResults": number,
   "NextToken": "string",
   "ProfileType": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

MaxResults

Jumlah maksimum profil untuk kembali.

Jenis: Integer

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 1000.

Diperlukan: Tidak

NextToken

Ketika ada hasil tambahan yang tidak dikembalikan, `NextToken` parameter dikembalikan. Anda dapat menggunakan nilai tersebut untuk panggilan berikutnya `ListProfiles` untuk melanjutkan hasil daftar.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 6144.

Diperlukan: Tidak

ProfileType

Menunjukkan apakah hanya akan mencantumkan profil `LOCAL` tipe atau hanya `PARTNER` mengetik profil. Jika tidak disediakan dalam permintaan, perintah mencantumkan semua jenis profil.

Jenis: String

Nilai yang Valid: `LOCAL` | `PARTNER`

Diperlukan: Tidak

Sintaksis Respons

```
{
   "NextToken": "string",
   "Profiles": [
      {
         "Arn": "string",
         "As2Id": "string",
         "ProfileId": "string",
         "ProfileType": "string"
      }
   ]
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

NextToken

Mengembalikan token yang dapat Anda gunakan untuk menelepon `ListProfiles` lagi dan menerima hasil tambahan, jika ada.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 6144.

Profiles

Mengembalikan array, di mana setiap item berisi rincian profil.

Tipe: Array objek ListedProfile

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidNextTokenException

`NextToken`Parameter yang dilewatkan tidak valid.

Kode Status HTTP: 400

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListSecurityPolicies

Daftar kebijakan keamanan yang dilampirkan ke server dan konektor SFTP Anda. Untuk informasi selengkapnya tentang kebijakan keamanan, lihat Bekerja dengan kebijakan keamanan untuk server atau Bekerja dengan kebijakan keamanan untuk konektor SFTP.

Sintaksis Permintaan

```
{
   "MaxResults": number,
   "NextToken": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

MaxResults

Menentukan jumlah kebijakan keamanan untuk kembali sebagai respon terhadap `ListSecurityPolicies` query.

Jenis: Integer

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 1000.

Diperlukan: Tidak

NextToken

Ketika hasil tambahan diperoleh dari `ListSecurityPolicies` perintah, `NextToken` parameter dikembalikan dalam output. Anda kemudian dapat meneruskan `NextToken` parameter dalam perintah berikutnya untuk melanjutkan daftar kebijakan keamanan tambahan.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 6144.

Diperlukan: Tidak

Sintaksis Respons

```
{
   "NextToken": "string",
   "SecurityPolicyNames": [ "string" ]
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

NextToken

Ketika Anda bisa mendapatkan hasil tambahan dari `ListSecurityPolicies` operasi, `NextToken` parameter dikembalikan dalam output. Dalam perintah berikut, Anda dapat meneruskan `NextToken` parameter untuk melanjutkan daftar kebijakan keamanan.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 6144.

SecurityPolicyNames

Berbagai kebijakan keamanan yang terdaftar.

Tipe: Array string

Batasan Panjang: Panjang minimum sebesar 0. Panjang maksimum 100.

Pola: `Transfer[A-Za-z0-9]*SecurityPolicy-[A-Za-z0-9-]+`

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilemparkan ketika kesalahan terjadi dalam AWS Transfer Family layanan.

Kode Status HTTP: 500

InvalidNextTokenException

`NextToken`Parameter yang dilewatkan tidak valid.

Kode Status HTTP: 400

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Contoh-contoh

Contoh

Contoh berikut mencantumkan nama untuk semua kebijakan keamanan yang tersedia.

Permintaan Sampel

```
aws transfer list-security-policies
```

Contoh Respons

```
{
    "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"
    ]
```

```
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

ListServers

Daftar server berkemampuan protokol transfer file yang terkait dengan akun Anda. AWS

Sintaksis Permintaan

```
{
   "MaxResults": number,
   "NextToken": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

MaxResults

Menentukan jumlah server untuk kembali sebagai respon terhadap `ListServers` query.

Jenis: Integer

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 1000.

Diperlukan: Tidak

NextToken

Ketika hasil tambahan diperoleh dari `ListServers` perintah, `NextToken` parameter dikembalikan dalam output. Anda kemudian dapat meneruskan `NextToken` parameter dalam perintah berikutnya untuk melanjutkan daftar server tambahan.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 6144.

Diperlukan: Tidak

Sintaksis Respons

```
{
   "NextToken": "string",
```

```
   "Servers": [
      {
         "Arn": "string",
         "Domain": "string",
         "EndpointType": "string",
         "IdentityProviderType": "string",
         "LoggingRole": "string",
         "ServerId": "string",
         "State": "string",
         "UserCount": number
      }
   ]
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

NextToken

Ketika Anda bisa mendapatkan hasil tambahan dari `ListServers` operasi, `NextToken` parameter dikembalikan dalam output. Dalam perintah berikut, Anda dapat meneruskan `NextToken` parameter untuk melanjutkan daftar server tambahan.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 6144.

Servers

Sebuah array server yang terdaftar.

Tipe: Array objek ListedServer

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidNextTokenException

`NextToken`Parameter yang dilewatkan tidak valid.

Kode Status HTTP: 400

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Contoh-contoh

Contoh

Contoh berikut mencantumkan server yang ada di AndaAkun AWS.

Perhatikan bahwa `NextToken` nilai contoh tidak nyata: mereka dimaksudkan untuk menunjukkan cara menggunakan parameter.

Permintaan Sampel

```
{
   "MaxResults": 1,
   "NextToken": "token-from-previous-API-call"
}
```

Contoh Respons

```
{
   "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
        }
    ]
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListTagsForResource

Daftar semua tag yang terkait dengan Amazon Resource Name (ARN) yang Anda tentukan. Sumber daya dapat berupa pengguna, server, atau peran.

Sintaksis Permintaan

```
{
   "Arn": "string",
   "MaxResults": number,
   "NextToken": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

Arn

Meminta tag yang terkait dengan Nama Sumber Daya Amazon (ARN) tertentu. ARN adalah pengidentifikasi untuk AWS sumber daya tertentu, seperti server, pengguna, atau peran.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 1600.

Pola: `arn:\S+`

Diperlukan: Ya

MaxResults

Menentukan jumlah tag untuk kembali sebagai respon terhadap `ListTagsForResource` permintaan.

Jenis: Integer

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 1000.

Diperlukan: Tidak

NextToken

Ketika Anda meminta hasil tambahan dari `ListTagsForResource` operasi, `NextToken` parameter dikembalikan dalam input. Anda kemudian dapat meneruskan perintah berikutnya ke `NextToken` parameter untuk melanjutkan daftar tag tambahan.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 6144.

Diperlukan: Tidak

Sintaksis Respons

```
{
   "Arn": "string",
   "NextToken": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

Arn

ARN yang Anda tentukan untuk mencantumkan tag.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 1600.

Pola: `arn:\S+`

NextToken

Ketika Anda bisa mendapatkan hasil tambahan dari `ListTagsForResource` panggilan, `NextToken` parameter dikembalikan dalam output. Anda kemudian dapat meneruskan perintah berikutnya ke `NextToken` parameter untuk melanjutkan daftar tag tambahan.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 6144.

Tags

Pasangan kunci-nilai yang ditugaskan ke sumber daya, biasanya untuk tujuan pengelompokan dan pencarian item. Tag adalah metadata yang Anda tentukan.

Tipe: Array objek Tag

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50 item.

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidNextTokenException

`NextToken`Parameter yang dilewatkan tidak valid.

Kode Status HTTP: 400

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Contoh-contoh

Contoh

Contoh berikut mencantumkan tag untuk sumber daya dengan ARN yang Anda tentukan.

Permintaan Sampel

```
{
   "Arn": "arn:aws:transfer:us-east-1:176354371281:server/s-01234567890abcdef"
}
```

Contoh

Contoh ini menggambarkan salah satu penggunaan. ListTagsForResource

Contoh Respons

```
{
   "Tags": [
      {
         "Key": "Name",
         "Value": "MyServer"
      }
   ]
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.

- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListUsers

Daftar pengguna untuk server berkemampuan protokol transfer file yang Anda tentukan dengan meneruskan parameter. `ServerId`

Sintaksis Permintaan

```
{
   "MaxResults": number,
   "NextToken": "string",
   "ServerId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

MaxResults

Menentukan jumlah pengguna untuk kembali sebagai respon terhadap `ListUsers` permintaan.

Jenis: Integer

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 1000.

Diperlukan: Tidak

NextToken

Jika ada hasil tambahan dari `ListUsers` panggilan, `NextToken` parameter dikembalikan dalam output. Anda kemudian dapat meneruskan `NextToken` ke `ListUsers` perintah berikutnya, untuk melanjutkan daftar pengguna tambahan.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 6144.

Diperlukan: Tidak

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk server yang memiliki pengguna yang ditugaskan untuk itu.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

Sintaksis Respons

```
{
   "NextToken": "string",
   "ServerId": "string",
   "Users": [
      {
         "Arn": "string",
         "HomeDirectory": "string",
         "HomeDirectoryType": "string",
         "Role": "string",
         "SshPublicKeyCount": number,
         "UserName": "string"
      }
   ]
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

NextToken

Ketika Anda bisa mendapatkan hasil tambahan dari `ListUsers` panggilan, `NextToken` parameter dikembalikan dalam output. Anda kemudian dapat meneruskan perintah berikutnya ke `NextToken` parameter untuk melanjutkan daftar pengguna tambahan.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 6144.

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk server tempat pengguna ditugaskan.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Users

Mengembalikan pengguna Transfer Family dan propertinya untuk `ServerId` nilai yang Anda tentukan.

Tipe: Array objek ListedUser

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidNextTokenException

`NextToken`Parameter yang dilewatkan tidak valid.

Kode Status HTTP: 400

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Contoh-contoh

Contoh

Panggilan `ListUsers` API menampilkan daftar pengguna yang terkait dengan server yang Anda tentukan.

Permintaan Sampel

```
        {
         "MaxResults": 100,
         "NextToken": "eyJNYXJrZXIiOiBudWxsLCAiYm90b1X0cnVuU2F0ZV9hbW91bnQiOiAyfQ==",
         "ServerId": "s-01234567890abcdef"
         }
```

Contoh

Ini adalah contoh respons untuk panggilan API ini.

Contoh Respons

```
{
   "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"
```

```
        }
    ]
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

ListWorkflows

Daftar semua alur kerja yang terkait dengan wilayah Anda saat ini. Akun AWS

Sintaksis Permintaan

```
{
   "MaxResults": number,
   "NextToken": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

MaxResults

Menentukan jumlah maksimum alur kerja untuk kembali.

Jenis: Integer

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 1000.

Diperlukan: Tidak

NextToken

`ListWorkflows`mengembalikan `NextToken` parameter dalam output. Anda kemudian dapat meneruskan `NextToken` parameter dalam perintah berikutnya untuk melanjutkan daftar alur kerja tambahan.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 6144.

Diperlukan: Tidak

Sintaksis Respons

```
{
   "NextToken": "string",
```

```
   "Workflows": [
      {
         "Arn": "string",
         "Description": "string",
         "WorkflowId": "string"
      }
   ]
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

NextToken

`ListWorkflows`mengembalikan `NextToken` parameter dalam output. Anda kemudian dapat meneruskan `NextToken` parameter dalam perintah berikutnya untuk melanjutkan daftar alur kerja tambahan.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 6144.

Workflows

MengembalikanArn,`WorkflowId`, dan `Description` untuk setiap alur kerja.

Tipe: Array objek ListedWorkflow

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidNextTokenException

`NextToken`Parameter yang dilewatkan tidak valid.

Kode Status HTTP: 400

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

SendWorkflowStepState

Mengirim callback untuk langkah-langkah kustom asinkron.

TheExecutionId,WorkflowId, dan Token diteruskan ke sumber daya target selama pelaksanaan langkah kustom alur kerja. Anda harus menyertakan mereka yang memiliki panggilan balik mereka serta memberikan status.

Sintaksis Permintaan

```
{
   "ExecutionId": "string",
   "Status": "string",
   "Token": "string",
   "WorkflowId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

ExecutionId

Pengidentifikasi unik untuk eksekusi alur kerja.

Jenis: String

Batas Panjang: Panjang tetap 36.

Pola: `[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}`

Diperlukan: Ya

Status

Menunjukkan apakah langkah yang ditentukan berhasil atau gagal.

Jenis: String

Nilai yang Valid: `SUCCESS | FAILURE`

Diperlukan: Ya

Token

Digunakan untuk membedakan antara beberapa callback untuk beberapa langkah Lambda dalam eksekusi yang sama.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum adalah 64.

Pola: `\w+`

Diperlukan: Ya

WorkflowId

Pengidentifikasi unik untuk alur kerja.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `w-([a-z0-9]{17})`

Diperlukan: Ya

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

AccessDeniedException

Anda tidak memiliki akses yang memadai untuk melakukan tindakan ini.

Kode Status HTTP: 400

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

ThrottlingException

Permintaan ditolak karena throttling permintaan.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

StartFileTransfer

Memulai transfer file antara AWS penyimpanan lokal dan server AS2 atau SFTP jarak jauh.

- Untuk konektor AS2, Anda menentukan `ConnectorId` dan satu atau lebih `SendFilePaths` untuk mengidentifikasi file yang ingin Anda transfer.
- Untuk konektor SFTP, transfer file dapat berupa outbound atau inbound. Dalam kedua kasus, Anda menentukan`ConnectorId`. Tergantung pada arah transfer, Anda juga menentukan item berikut:
  - Jika Anda mentransfer file dari server SFTP mitra ke penyimpanan Amazon Web Services, Anda menentukan satu atau beberapa `RetreiveFilePaths` untuk mengidentifikasi file yang ingin ditransfer, dan `LocalDirectoryPath` untuk menentukan folder tujuan.
  - Jika Anda mentransfer file ke server SFTP mitra dari AWS penyimpanan, Anda menentukan satu atau lebih `SendFilePaths` untuk mengidentifikasi file yang ingin Anda transfer, dan `RemoteDirectoryPath` untuk menentukan folder tujuan.

Sintaksis Permintaan

```
{
   "ConnectorId": "string",
   "LocalDirectoryPath": "string",
   "RemoteDirectoryPath": "string",
   "RetrieveFilePaths": [ "string" ],
   "SendFilePaths": [ "string" ]
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

ConnectorId

Pengidentifikasi unik untuk konektor.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `c-([0-9a-f]{17})`

Diperlukan: Ya

LocalDirectoryPath

Untuk transfer masuk, `LocaDirectoryPath` menentukan tujuan untuk satu atau lebih file yang ditransfer dari server SFTP mitra.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `(.)+`

Diperlukan: Tidak

RemoteDirectoryPath

Untuk transfer keluar, `RemoteDirectoryPath` menentukan tujuan untuk satu atau lebih file yang ditransfer ke server SFTP mitra. Jika Anda tidak menentukan`RemoteDirectoryPath`, tujuan untuk file yang ditransfer adalah direktori home pengguna SFTP.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `(.)+`

Diperlukan: Tidak

RetrieveFilePaths

Satu atau lebih jalur sumber untuk server SFTP mitra. Setiap string mewakili jalur file sumber untuk satu transfer file masuk.

Tipe: Array string

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 10 item.

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `(.)+`

Diperlukan: Tidak

SendFilePaths

Satu atau lebih jalur sumber untuk penyimpanan Amazon S3. Setiap string mewakili jalur file sumber untuk satu transfer file keluar. Misalnya, *DOC-EXAMPLE-BUCKET/myfile.txt* .

> **Note**
>
> Ganti *DOC-EXAMPLE-BUCKET* dengan salah satu ember Anda yang sebenarnya.

Tipe: Array string

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 10 item.

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `(.)+`

Diperlukan: Tidak

Sintaksis Respons

```
{
   "TransferId": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

TransferId

Mengembalikan identifier unik untuk transfer file.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 512.

Pola: `[0-9a-zA-Z./-]+`

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

ThrottlingException

Permintaan ditolak karena throttling permintaan.

Kode Status HTTP: 400

Contoh-contoh

Contoh

Contoh berikut memulai transfer file AS2 dari server Transfer Family ke titik akhir mitra dagang jarak jauh. Ganti *DOC-EXAMPLE-BUCKET* dengan salah satu ember Anda yang sebenarnya.

Permintaan Sampel

```
{
    "ConnectorId": "c-AAAA1111BBBB2222C",
```

```
    "SendFilePaths": [
        "/DOC-EXAMPLE-BUCKET/myfile-1.txt",
        "/DOC-EXAMPLE-BUCKET/myfile-2.txt",
        "/DOC-EXAMPLE-BUCKET/myfile-3.txt"
    ]
}
```

Contoh Respons

```
{
   "TransferId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
}
```

Contoh

Contoh berikut memulai transfer file dari AWS penyimpanan lokal ke server SFTP jarak jauh.

Permintaan Sampel

```
{
    "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"
}
```

Contoh Respons

```
{
   "TransferId": "a1b2c3d4-5678-90ab-cdef-EXAMPLE22222"
}
```

Contoh

Contoh berikut memulai transfer file dari server SFTP jarak jauh ke penyimpanan lokalAWS.

Permintaan Sampel

```
{
```

```
    "ConnectorId": "c-111122223333AAAAA",
    "RetrieveFilePaths": [
        "/MySFTPFolder/toTranferFamily/myfile-1.txt",
        "/MySFTPFolder/toTranferFamily/myfile-2.txt",
        "/MySFTPFolder/toTranferFamily/myfile-3.txt"
    ],
    "LocalDirectoryPath": "/DOC-EXAMPLE-BUCKET/mySourceFiles"
}
```

Contoh Respons

```
{
   "TransferId": "a1b2c3d4-5678-90ab-cdef-EXAMPLEaaaaa"
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

StartServer

Mengubah status server berkemampuan protokol transfer file dari ke. `OFFLINE` `ONLINE` Ini tidak berdampak pada server yang sudah ada`ONLINE`. `ONLINE`Server dapat menerima dan memproses pekerjaan transfer file.

Status `STARTING` menunjukkan bahwa server berada dalam keadaan perantara, baik tidak sepenuhnya dapat merespons, atau tidak sepenuhnya online. Nilai `START_FAILED` dapat menunjukkan kondisi kesalahan.

Tidak ada tanggapan yang dikembalikan dari panggilan ini.

Sintaksis Permintaan

```
{
   "ServerId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk server yang Anda mulai.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

ThrottlingException

Permintaan ditolak karena throttling permintaan.

Kode Status HTTP: 400

Contoh-contoh

Contoh

Contoh berikut memulai server.

Permintaan Sampel

```
{
   "ServerId": "s-01234567890abcdef"
```

```
}
```

Contoh

Ini adalah contoh respons untuk panggilan API ini.

Contoh Respons

```
{
   "ServerId": "s-01234567890abcdef"
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

StopServer

Mengubah status server berkemampuan protokol transfer file dari ke. `ONLINE` `OFFLINE` `OFFLINE`Server tidak dapat menerima dan memproses pekerjaan transfer file. Informasi yang terkait dengan server Anda, seperti server dan properti pengguna, tidak terpengaruh dengan menghentikan server Anda.

> **Note**
>
> Menghentikan server tidak mengurangi atau memengaruhi penagihan titik akhir protokol transfer file Anda; Anda harus menghapus server untuk berhenti ditagih.

Status `STOPPING` menunjukkan bahwa server berada dalam keadaan perantara, baik tidak sepenuhnya dapat merespons, atau tidak sepenuhnya offline. Nilai `STOP_FAILED` dapat menunjukkan kondisi kesalahan.

Tidak ada tanggapan yang dikembalikan dari panggilan ini.

Sintaksis Permintaan

```
{
   "ServerId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk server yang Anda hentikan.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

ThrottlingException

Permintaan ditolak karena throttling permintaan.

Kode Status HTTP: 400

Contoh-contoh

Contoh

Contoh berikut menghentikan server.

Permintaan Sampel

```
{
   "ServerId": "s-01234567890abcdef"
}
```

Contoh

Ini adalah contoh respons untuk panggilan API ini.

Contoh Respons

```
{
   "ServerId": "s-01234567890abcdef"
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

TagResource

Melampirkan pasangan kunci-nilai ke sumber daya, seperti yang diidentifikasi oleh Amazon Resource Name (ARN). Sumber daya adalah pengguna, server, peran, dan entitas lainnya.

Tidak ada jawaban yang dikembalikan dari panggilan ini.

Sintaksis Permintaan

```
{
   "Arn": "string",
   "Tags": [
      {
         "Key": "string",
         "Value": "string"
      }
   ]
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

Arn

Nama Sumber Daya Amazon (ARN) untuk AWS sumber daya tertentu, seperti server, pengguna, atau peran.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 1600.

Pola: `arn:\S+`

Diperlukan: Ya

Tags

Pasangan nilai kunci yang ditetapkan ke ARN yang dapat Anda gunakan untuk mengelompokkan dan mencari sumber daya berdasarkan jenis. Anda dapat melampirkan metadata ini ke sumber daya (server, pengguna, alur kerja, dan sebagainya) untuk tujuan apa pun.

Tipe: Array objek Tag

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50 item.

Diperlukan: Ya

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Contoh-contoh

Contoh

Contoh berikut menambahkan tag ke server berkemampuan protokol transfer file.

Permintaan Sampel

```
{
   "Arn": "arn:aws:transfer:us-east-1:176354371281:server/s-01234567890abcdef",
   "Tags": [
      {
         "Key": "Group",
         "Value": "Europe"
      }
   ]
}
```

Contoh

Contoh ini menggambarkan salah satu penggunaan. TagResource

Contoh Respons

```
         HTTP 200 response with an empty HTTP body.
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

TestConnection

Menguji apakah konektor SFTP Anda berhasil diatur. Kami sangat menyarankan Anda memanggil operasi ini untuk menguji kemampuan Anda mentransfer file antara AWS penyimpanan lokal dan server SFTP mitra dagang.

Sintaksis Permintaan

```
{
   "ConnectorId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

ConnectorId

Pengidentifikasi unik untuk konektor.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `c-([0-9a-f]{17})`

Diperlukan: Ya

Sintaksis Respons

```
{
   "ConnectorId": "string",
   "Status": "string",
   "StatusMessage": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

ConnectorId

Mengembalikan identifier dari objek konektor yang Anda uji.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `c-([0-9a-f]{17})`

Status

Pengembalian `OK` untuk tes yang berhasil, atau `ERROR` jika tes gagal.

Jenis: String

StatusMessage

Kembali `Connection succeeded` jika tes berhasil. Atau, mengembalikan pesan kesalahan deskriptif jika tes gagal. Daftar berikut memberikan rincian pemecahan masalah, tergantung pada pesan kesalahan yang Anda terima.

- Verifikasi bahwa nama rahasia Anda sejajar dengan yang ada di izin Peran Transfer.
- Verifikasi URL server dalam konfigurasi konektor, dan verifikasi bahwa kredenal login berhasil bekerja di luar konektor.
- Verifikasi bahwa rahasia itu ada dan diformat dengan benar.
- Verifikasi bahwa kunci host tepercaya dalam konfigurasi konektor cocok dengan `ssh-keyscan` output.

Jenis: String

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilemparkan ketika kesalahan terjadi dalam AWS Transfer Family layanan.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Contoh-contoh

Contoh

Contoh berikut menguji koneksi ke server jarak jauh.

```
aws transfer test-connection --connector-id c-abcd1234567890fff
```

Contoh Respons

Jika berhasil, panggilan API mengembalikan detail berikut.

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

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

TestIdentityProvider

Jika server `IdentityProviderType` berkemampuan protokol transfer file `AWS_DIRECTORY_SERVICE` atau`API_Gateway`, uji apakah penyedia identitas Anda berhasil disiapkan. Kami sangat menyarankan Anda memanggil operasi ini untuk menguji metode otentikasi Anda segera setelah Anda membuat server Anda. Dengan demikian, Anda dapat memecahkan masalah dengan integrasi penyedia identitas untuk memastikan bahwa pengguna Anda dapat berhasil menggunakan layanan.

Parameter `ServerId` dan `UserName` diperlukan. Itu`ServerProtocol`,`SourceIp`, dan `UserPassword` semuanya opsional.

Perhatikan hal berikut:

- Anda tidak dapat menggunakan `TestIdentityProvider` jika server Anda`SERVICE_MANAGED`. `IdentityProviderType`
- `TestIdentityProvider`tidak berfungsi dengan kunci: hanya menerima kata sandi.
- `TestIdentityProvider`dapat menguji operasi kata sandi untuk Penyedia Identitas khusus yang menangani kunci dan kata sandi.
- Jika Anda memberikan nilai yang salah untuk parameter apa pun, `Response` bidang kosong.
- Jika Anda memberikan ID server untuk server yang menggunakan pengguna yang dikelola layanan, Anda mendapatkan kesalahan:

  ```
  An error occurred (InvalidRequestException) when calling the
  TestIdentityProvider operation: s-server-ID not configured for external
  auth
  ```

- Jika Anda memasukkan ID Server untuk `--server-id` parameter yang tidak mengidentifikasi server Transfer yang sebenarnya, Anda menerima kesalahan berikut:

  ```
  An error occurred (ResourceNotFoundException) when calling the
  TestIdentityProvider operation: Unknown server.
  ```

  Mungkin saja sever Anda berada di wilayah yang berbeda. Anda dapat menentukan wilayah dengan menambahkan yang berikut:`--region region-code`, seperti `--region us-east-2` menentukan server di AS Timur (Ohio).

Sintaksis Permintaan

```
{
   "ServerId": "string",
   "ServerProtocol": "string",
   "SourceIp": "string",
   "UserName": "string",
   "UserPassword": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

ServerId

Pengidentifikasi yang ditetapkan sistem untuk server tertentu. Metode otentikasi pengguna server itu diuji dengan nama pengguna dan kata sandi.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

ServerProtocol

Jenis protokol transfer file yang akan diuji.

Protokol yang tersedia adalah:

- Protokol Transfer File Secure Shell (SSH) (SFTP)
- Protokol Transfer File Aman (FTPS)
- Protokol Transfer File (FTP)
- Pernyataan Penerapan 2 (AS2)

Jenis: String

Nilai yang Valid: `SFTP | FTP | FTPS | AS2`

Diperlukan: Tidak

SourceIp

Alamat IP sumber akun yang akan diuji.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum sebesar 32.

Pola: `\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}`

Diperlukan: Tidak

UserName

Nama akun yang akan diuji.

Jenis: String

Batasan Panjang: Panjang minimum 3. Panjang maksimum 100.

Pola: `[\w][\w@.-]{2,99}`

Diperlukan: Ya

UserPassword

Kata sandi akun yang akan diuji.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 1024.

Wajib: Tidak

Sintaksis Respons

```
{
   "Message": "string",
   "Response": "string",
   "StatusCode": number,
   "Url": "string"
```

```
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

Message

Pesan yang menunjukkan apakah tes berhasil atau tidak.

> **Note**
>
> Jika string kosong dikembalikan, penyebab yang paling mungkin adalah otentikasi gagal karena nama pengguna atau kata sandi yang salah.

Jenis: String

Response

Respons yang dikembalikan dari API Gateway atau fungsi Lambda Anda.

Jenis: String

StatusCode

Kode status HTTP yang merupakan respons dari API Gateway atau fungsi Lambda Anda.

Jenis: Integer

Url

Titik akhir layanan yang digunakan untuk mengautentikasi pengguna.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum sebesar 255.

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilemparkan ketika kesalahan terjadi dalam AWS Transfer Family layanan.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Contoh-contoh

Contoh

Permintaan berikut mengembalikan pesan dari penyedia identitas bahwa kombinasi nama pengguna dan kata sandi adalah identitas yang valid untuk digunakan AWS Transfer Family.

Permintaan Sampel

```
{
   "ServerID": "s-01234567890abcdef",
   "UserName": "my_user",
   "UserPassword": "MyPassword-1"
}
```

Contoh

Respons berikut menunjukkan respons sampel untuk tes yang berhasil.

Contoh Respons

```
      "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": ""
```

Contoh

Respons berikut menunjukkan bahwa pengguna yang ditentukan milik lebih dari satu grup yang memiliki akses.

```
         "Response":"",
         "StatusCode":200,
         "Message":"More than one associated access found for user's groups."
```

Contoh

Jika Anda telah membuat dan mengonfigurasi penyedia identitas kustom dengan menggunakan API Gateway, Anda dapat memasukkan perintah berikut untuk menguji pengguna Anda:

`aws transfer test-identity-provider --server-id` *`s-0123456789abcdefg`* `--user-name` *`myuser`*

di mana s-0123456789abcdefg adalah server transfer Anda, dan myuser adalah nama pengguna untuk pengguna kustom Anda.

Jika perintah berhasil, respons Anda mirip dengan yang berikut ini, di mana:

- Akun AWS ID adalah 012345678901
- Peran pengguna adalah user-role-api-gateway
- Direktori home adalah myuser-bucket
- Kunci publik adalah kunci publik

- URL pemanggilan adalah URL pemanggilan

```
{
    "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"
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

UntagResource

Melepaskan pasangan kunci-nilai dari sumber daya, seperti yang diidentifikasi oleh Amazon Resource Name (ARN). Sumber daya adalah pengguna, server, peran, dan entitas lainnya.

Tidak ada tanggapan yang dikembalikan dari panggilan ini.

Sintaksis Permintaan

```
{
   "Arn": "string",
   "TagKeys": [ "string" ]
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

Arn

Nilai sumber daya yang akan menghapus tag. Nama Sumber Daya Amazon (ARN) adalah pengidentifikasi untuk AWS sumber daya tertentu, seperti server, pengguna, atau peran.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 1600.

Pola: `arn:\S+`

Diperlukan: Ya

TagKeys

TagKeys adalah pasangan nilai kunci yang ditetapkan ke ARN yang dapat digunakan untuk mengelompokkan dan mencari sumber daya berdasarkan jenis. Metadata ini dapat dilampirkan ke sumber daya untuk tujuan apa pun.

Tipe: Array string

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50 item.

Batasan Panjang: Panjang minimum sebesar 0. Panjang maksimum 128.

Diperlukan: Ya

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200 dengan isi HTTP kosong.

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

Contoh-contoh

Contoh

Contoh berikut menghapus tag server berkemampuan protokol transfer file.

Permintaan Sampel

```
{
   "Arn": "arn:aws:transfer:us-east-1:176354371281:server/s-01234567890abcdef",
   "TagKeys": "Europe" ]
}
```

Contoh

Contoh ini menggambarkan salah satu penggunaan. UntagResource

Contoh Respons

```
        HTTP 200 response with an empty HTTP body.
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateAccess

Memungkinkan Anda memperbarui parameter untuk akses yang ditentukan dalam `ExternalID` parameter `ServerID` dan.

Sintaksis Permintaan

```
{
   "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"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

ExternalId

Pengidentifikasi unik yang diperlukan untuk mengidentifikasi grup tertentu dalam direktori Anda. Pengguna grup yang Anda asosiasikan memiliki akses ke sumber daya Amazon S3 atau Amazon EFS Anda melalui protokol yang diaktifkan. AWS Transfer Family Jika Anda tahu nama grup, Anda dapat melihat nilai SID dengan menjalankan perintah berikut menggunakan Windows PowerShell.

```
Get-ADGroup -Filter {samAccountName -like "YourGroupName*"} -Properties
* | Select SamAccountName,ObjectSid
```

Dalam perintah itu, ganti YourGroupNamedengan nama grup Active Directory Anda.

Ekspresi reguler yang digunakan untuk memvalidasi parameter ini adalah string karakter yang terdiri dari huruf besar dan huruf kecil karakter alfanumerik tanpa spasi. Anda juga dapat menyertakan garis bawah atau salah satu karakter berikut: =, . @: /-

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `S-1-[\d-]+`

Diperlukan: Ya

HomeDirectory

Direktori arahan (folder) untuk pengguna ketika mereka masuk ke server menggunakan klien.

Contoh `HomeDirectory` adalah `/bucket_name/home/mydirectory`.

> **Note**
>
> Parameter `HomeDirectory` hanya digunakan jika `HomeDirectoryType` diatur ke `PATH`.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 1024.

Pola: `(|/.*)`

Diperlukan: Tidak

HomeDirectoryMappings

Pemetaan direktori logis yang menentukan jalur dan kunci Amazon S3 atau Amazon EFS apa yang harus terlihat oleh pengguna Anda dan bagaimana Anda ingin membuatnya terlihat. Anda harus menentukan `Entry` dan `Target` memasangkan, di mana `Entry` menunjukkan bagaimana jalur dibuat terlihat dan `Target` merupakan jalur Amazon S3 atau Amazon EFS yang sebenarnya.

Jika Anda hanya menentukan target, itu ditampilkan apa adanya. Anda juga harus memastikan bahwa peran AWS Identity and Access Management (IAM) Anda menyediakan akses ke jalur masuk`Target`. Nilai ini dapat diatur hanya ketika `HomeDirectoryType` diatur ke LOGICAL.

Berikut ini adalah contoh `Entry` dan `Target` pair.

```
[ { "Entry": "/directory1", "Target": "/bucket_name/home/
mydirectory" } ]
```

Dalam kebanyakan kasus, Anda dapat menggunakan nilai ini alih-alih kebijakan sesi untuk mengunci pengguna Anda ke direktori home yang ditunjuk (`chroot`“”). Untuk melakukan ini, Anda dapat mengatur `Entry` ke `/` dan mengatur `Target` ke nilai `HomeDirectory` parameter.

Berikut ini adalah contoh `Entry` dan `Target` pair untuk`chroot`.

```
[ { "Entry": "/", "Target": "/bucket_name/home/mydirectory" } ]
```

Tipe: Array objek HomeDirectoryMapEntry

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50000 item.

Diperlukan: Tidak

HomeDirectoryType

Jenis direktori pendaratan (folder) yang Anda inginkan direktori home pengguna Anda ketika mereka masuk ke server. Jika Anda mengaturnya`PATH`, pengguna akan melihat bucket Amazon S3 absolut atau jalur Amazon EFS seperti pada klien protokol transfer file mereka. Jika Anda menyetelnya`LOGICAL`, Anda harus menyediakan pemetaan `HomeDirectoryMappings` untuk bagaimana Anda ingin membuat jalur Amazon S3 atau Amazon EFS terlihat oleh pengguna Anda.

> **Note**
>
> Jika `HomeDirectoryType` ya`LOGICAL`, Anda harus memberikan pemetaan, menggunakan parameter. `HomeDirectoryMappings` Jika, di sisi lain, `HomeDirectoryType` adalah`PATH`, Anda memberikan jalur absolut menggunakan `HomeDirectory` parameter. Anda tidak dapat memiliki keduanya `HomeDirectory` dan `HomeDirectoryMappings` di template Anda.

Jenis: String

Nilai yang Valid: `PATH | LOGICAL`

Diperlukan: Tidak

Policy

Kebijakan sesi untuk pengguna Anda sehingga Anda dapat menggunakan peran yang sama AWS Identity and Access Management (IAM) di beberapa pengguna. Kebijakan ini mencakup akses pengguna ke sebagian bucket Amazon S3 mereka. Variabel yang dapat Anda gunakan dalam kebijakan ini meliputi `${Transfer:UserName}`, `${Transfer:HomeDirectory}`, dan `${Transfer:HomeBucket}`.

> **Note**
>
> Kebijakan ini hanya berlaku jika domainnya `ServerId` adalah Amazon S3. Amazon EFS tidak menggunakan kebijakan sesi.
> Untuk kebijakan sesi, AWS Transfer Family menyimpan kebijakan sebagai gumpalan JSON, bukan Nama Sumber Daya Amazon (ARN) kebijakan tersebut. Anda menyimpan kebijakan sebagai blob JSON dan meneruskan dalam argumen `Policy`.
> Untuk contoh kebijakan sesi, lihat Contoh kebijakan sesi.
> Untuk informasi selengkapnya, lihat AssumeRoledi Referensi API Layanan Token AWS Keamanan.

Jenis: String

Batasan Panjang: Panjang minimum sebesar 0. Panjang maksimum 2048.

Diperlukan: Tidak

PosixProfile

Identitas POSIX lengkap, termasuk ID pengguna (`Uid`), ID grup (`Gid`), dan setiap grup sekunder ID (`SecondaryGids`), yang mengendalikan akses pengguna Anda ke sistem file Amazon EFS Anda. POSIX izin yang ditetapkan pada file dan direktori dalam sistem file Anda menentukan tingkat akses yang pengguna Anda dapatkan ketika mentransfer file ke dalam dan keluar dari sistem file Amazon EFS Anda.

Tipe: Objek PosixProfile

Diperlukan: Tidak

Role

Nama Sumber Daya Amazon (ARN) dari peran AWS Identity and Access Management (IAM) yang mengontrol akses pengguna ke bucket Amazon S3 atau sistem file Amazon EFS. Kebijakan yang dilampirkan pada peran ini menentukan tingkat akses yang ingin Anda berikan kepada pengguna saat mentransfer file masuk dan keluar dari bucket Amazon S3 atau sistem file Amazon EFS. IAM role juga harus berisi hubungan kepercayaan yang mengizinkan server untuk mengakses sumber daya Anda saat melayani permintaan transfer pengguna.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 2048.

Pola: `arn:.*role/\S+`

Diperlukan: Tidak

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk instans server. Ini adalah server tertentu tempat Anda menambahkan pengguna.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

Sintaksis Respons

```
{
   "ExternalId": "string",
   "ServerId": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

ExternalId

Pengenal eksternal grup yang penggunanya memiliki akses ke sumber daya Amazon S3 atau Amazon EFS Anda melalui protokol yang diaktifkan menggunakan AWS Transfer Family.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `S-1-[\d-]+`

ServerId

Pengidentifikasi server tempat pengguna dilampirkan.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceExistsException

Sumber daya yang diminta tidak ada, atau ada di wilayah selain yang ditentukan untuk perintah.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

ThrottlingException

Permintaan ditolak karena throttling permintaan.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateAgreement

Memperbarui beberapa parameter untuk perjanjian yang ada. Berikan `AgreementId` dan `ServerId` untuk perjanjian yang ingin Anda perbarui, bersama dengan nilai baru untuk parameter yang akan diperbarui.

Sintaksis Permintaan

```
{
   "AccessRole": "string",
   "AgreementId": "string",
   "BaseDirectory": "string",
   "Description": "string",
   "LocalProfileId": "string",
   "PartnerProfileId": "string",
   "ServerId": "string",
   "Status": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

AccessRole

Konektor digunakan untuk mengirim file menggunakan protokol AS2 atau SFTP. Untuk peran akses, berikan Nama Sumber Daya Amazon (ARN) AWS Identity and Access Management peran yang akan digunakan.

Untuk konektor AS2

Dengan AS2, Anda dapat mengirim file dengan memanggil `StartFileTransfer` dan menentukan jalur file dalam parameter permintaan,. `SendFilePaths` Kami menggunakan direktori induk file (misalnya, untuk, direktori induk`/bucket/dir/`) untuk `--send-file-paths /bucket/dir/file.txt` sementara menyimpan file pesan AS2 yang diproses, menyimpan MDN ketika kami menerimanya dari mitra, dan menulis file JSON akhir yang berisi metadata transmisi yang relevan. Jadi, `AccessRole` kebutuhan untuk menyediakan akses baca dan tulis ke direktori induk dari lokasi file yang digunakan dalam `StartFileTransfer`

permintaan. Selain itu, Anda perlu menyediakan akses baca dan tulis ke direktori induk dari file yang ingin Anda kirim`StartFileTransfer`.

Jika Anda menggunakan otentikasi Dasar untuk konektor AS2 Anda, peran akses memerlukan `secretsmanager:GetSecretValue` izin untuk rahasia tersebut. Jika rahasia dienkripsi menggunakan kunci yang dikelola pelanggan alih-alih kunci yang dikelola di AWS Secrets Manager, maka peran tersebut juga memerlukan `kms:Decrypt` izin untuk kunci tersebut.

Untuk konektor SFTP

Pastikan bahwa peran akses menyediakan akses baca dan tulis ke direktori induk dari lokasi file yang digunakan dalam `StartFileTransfer` permintaan. Selain itu, pastikan bahwa peran tersebut memberikan `secretsmanager:GetSecretValue` izin untukAWS Secrets Manager.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 2048.

Pola: `arn:.*role/\S+`

Diperlukan: Tidak

AgreementId

Pengidentifikasi unik untuk perjanjian. Pengenal ini dikembalikan saat Anda membuat perjanjian.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `a-([0-9a-f]{17})`

Diperlukan: Ya

BaseDirectory

Untuk mengubah direktori pendaratan (folder) untuk file yang ditransfer, berikan folder bucket yang ingin Anda gunakan; misalnya,*`/DOC-EXAMPLE-BUCKET/home/mydirectory`* .

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 1024.

Pola: `(|/.*)`

Diperlukan: Tidak

Description

Untuk mengganti deskripsi yang ada, berikan deskripsi singkat untuk perjanjian tersebut.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 200.

Pola: `[\p{Graph}]+`

Diperlukan: Tidak

LocalProfileId

Pengidentifikasi unik untuk profil lokal AS2.

Untuk mengubah pengenal profil lokal, berikan nilai baru di sini.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `p-([0-9a-f]{17})`

Diperlukan: Tidak

PartnerProfileId

Pengenal unik untuk profil mitra. Untuk mengubah pengenal profil mitra, berikan nilai baru di sini.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `p-([0-9a-f]{17})`

Diperlukan: Tidak

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk instans server. Ini adalah server khusus yang digunakan perjanjian.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

Status

Anda dapat memperbarui status perjanjian, baik mengaktifkan perjanjian yang tidak aktif atau sebaliknya.

Jenis: String

Nilai yang Valid: `ACTIVE | INACTIVE`

Diperlukan: Tidak

Sintaksis Respons

```
{
   "AgreementId": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

AgreementId

Pengidentifikasi unik untuk perjanjian. Pengenal ini dikembalikan saat Anda membuat perjanjian.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `a-([0-9a-f]{17})`

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceExistsException

Sumber daya yang diminta tidak ada, atau ada di wilayah selain yang ditentukan untuk perintah.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

ThrottlingException

Permintaan ditolak karena throttling permintaan.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++

- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateCertificate

Memperbarui tanggal aktif dan tidak aktif untuk sertifikat.

Sintaksis Permintaan

```
{
   "ActiveDate": number,
   "CertificateId": "string",
   "Description": "string",
   "InactiveDate": number
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

ActiveDate

Tanggal opsional yang menentukan kapan sertifikat menjadi aktif.

Tipe: Timestamp

Diperlukan: Tidak

CertificateId

Pengidentifikasi objek sertifikat yang Anda perbarui.

Jenis: String

Kendala Panjang: Panjang tetap 22.

Pola: `cert-([0-9a-f]{17})`

Diperlukan: Ya

Description

Deskripsi singkat untuk membantu mengidentifikasi sertifikat.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 200.

Pola: `[\p{Graph}]+`

Diperlukan: Tidak

InactiveDate

Tanggal opsional yang menentukan kapan sertifikat menjadi tidak aktif.

Tipe: Timestamp

Diperlukan: Tidak

Sintaksis Respons

```
{
   "CertificateId": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

CertificateId

Mengembalikan identifier dari objek sertifikat yang Anda perbarui.

Jenis: String

Kendala Panjang: Panjang tetap 22.

Pola: `cert-([0-9a-f]{17})`

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

ThrottlingException

Permintaan ditolak karena throttling permintaan.

Kode Status HTTP: 400

Contoh-contoh

Contoh

Contoh berikut memperbarui tanggal aktif sertifikat, menyetel tanggal aktif ke 16 Januari 2022 pukul 16:12:07 UTC -5 jam.

Permintaan Sampel

```
aws transfer update-certificate --certificate-id c-abcdefg123456hijk --active-date
 2022-01-16T16:12:07-05:00
```

Contoh

Berikut ini adalah contoh respons untuk panggilan API ini.

Contoh Respons

```
"CertificateId": "c-abcdefg123456hijk"
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateConnector

Memperbarui beberapa parameter untuk konektor yang ada. Berikan konektor `ConnectorId` yang ingin Anda perbarui, bersama dengan nilai baru untuk parameter yang akan diperbarui.

Sintaksis Permintaan

```
{
   "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"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

AccessRole

Konektor digunakan untuk mengirim file menggunakan protokol AS2 atau SFTP. Untuk peran akses, berikan Nama Sumber Daya Amazon (ARN) AWS Identity and Access Management peran yang akan digunakan.

Untuk konektor AS2

Dengan AS2, Anda dapat mengirim file dengan memanggil `StartFileTransfer` dan menentukan jalur file dalam parameter permintaan,. `SendFilePaths` Kami menggunakan direktori induk file (misalnya, untuk, direktori induk`/bucket/dir/`) untuk `--send-file-paths /bucket/dir/file.txt` sementara menyimpan file pesan AS2 yang diproses, menyimpan MDN ketika kami menerimanya dari mitra, dan menulis file JSON akhir yang berisi metadata transmisi yang relevan. Jadi, `AccessRole` kebutuhan untuk menyediakan akses baca dan tulis ke direktori induk dari lokasi file yang digunakan dalam `StartFileTransfer` permintaan. Selain itu, Anda perlu menyediakan akses baca dan tulis ke direktori induk dari file yang ingin Anda kirim`StartFileTransfer`.

Jika Anda menggunakan otentikasi Dasar untuk konektor AS2 Anda, peran akses memerlukan `secretsmanager:GetSecretValue` izin untuk rahasia tersebut. Jika rahasia dienkripsi menggunakan kunci yang dikelola pelanggan alih-alih kunci yang dikelola di AWS Secrets Manager, maka peran tersebut juga memerlukan `kms:Decrypt` izin untuk kunci tersebut.

Untuk konektor SFTP

Pastikan bahwa peran akses menyediakan akses baca dan tulis ke direktori induk lokasi file yang digunakan dalam `StartFileTransfer` permintaan. Selain itu, pastikan bahwa peran tersebut memberikan `secretsmanager:GetSecretValue` izin untuk AWS Secrets Manager.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 2048.

Pola: `arn:.*role/\S+`

Diperlukan: Tidak

As2Config

Struktur yang berisi parameter untuk objek konektor AS2.

Tipe: Objek As2ConnectorConfig

Diperlukan: Tidak

ConnectorId

Pengidentifikasi unik untuk konektor.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `c-([0-9a-f]{17})`

Diperlukan: Ya

LoggingRole

Nama Sumber Daya Amazon (ARN) dari peran AWS Identity and Access Management (IAM) yang memungkinkan konektor mengaktifkan CloudWatch logging untuk peristiwa Amazon S3. Saat disetel, Anda dapat melihat aktivitas konektor di CloudWatch log Anda.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 2048.

Pola: `arn:.*role/\S+`

Diperlukan: Tidak

SecurityPolicyName

Menentukan nama kebijakan keamanan untuk konektor.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 100.

Pola: `TransferSFTPConnectorSecurityPolicy-[A-Za-z0-9-]+`

Diperlukan: Tidak

SftpConfig

Struktur yang berisi parameter untuk objek konektor SFTP.

Tipe: Objek SftpConnectorConfig

Diperlukan: Tidak

Url

URL titik akhir AS2 atau SFTP mitra.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum sebesar 255.

Diperlukan: Tidak

Sintaksis Respons

```
{
   "ConnectorId": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

ConnectorId

Mengembalikan identifier dari objek konektor yang Anda memperbarui.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `c-([0-9a-f]{17})`

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilemparkan ketika kesalahan terjadi dalam AWS Transfer Family layanan.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceExistsException

Sumber daya yang diminta tidak ada, atau ada di wilayah selain yang ditentukan untuk perintah.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

ThrottlingException

Permintaan ditolak karena throttling permintaan.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

UpdateHostKey

Memperbarui deskripsi untuk kunci host yang ditentukan oleh `HostKeyId` parameter `ServerId` dan.

Sintaksis Permintaan

```
{
   "Description": "string",
   "HostKeyId": "string",
   "ServerId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

Description

Deskripsi yang diperbarui untuk kunci host.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 200.

Pola: `[\p{Print}]*`

Diperlukan: Ya

HostKeyId

Pengidentifikasi kunci host yang Anda perbarui.

Jenis: String

Kendala Panjang: Panjang tetap 25.

Pola: `hostkey-[0-9a-f]{17}`

Diperlukan: Ya

ServerId

Pengidentifikasi server yang berisi kunci host yang Anda perbarui.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

Sintaksis Respons

```
{
   "HostKeyId": "string",
   "ServerId": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

HostKeyId

Mengembalikan pengenal kunci host untuk kunci host diperbarui.

Jenis: String

Kendala Panjang: Panjang tetap 25.

Pola: `hostkey-[0-9a-f]{17}`

ServerId

Mengembalikan pengenal server untuk server yang berisi kunci host diperbarui.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

ThrottlingException

Permintaan ditolak karena throttling permintaan.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateProfile

Memperbarui beberapa parameter untuk profil yang ada. Berikan profil `ProfileId` yang ingin Anda perbarui, bersama dengan nilai baru untuk parameter yang akan diperbarui.

Sintaksis Permintaan

```
{
   "CertificateIds": [ "string" ],
   "ProfileId": "string"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

CertificateIds

Array pengidentifikasi untuk sertifikat yang diimpor. Anda menggunakan pengenal ini untuk bekerja dengan profil dan profil mitra.

Tipe: Array string

Kendala Panjang: Panjang tetap 22.

Pola: `cert-([0-9a-f]{17})`

Diperlukan: Tidak

ProfileId

Pengidentifikasi objek profil yang Anda perbarui.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `p-([0-9a-f]{17})`

Diperlukan: Ya

Sintaksis Respons

```
{
   "ProfileId": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

ProfileId

Mengembalikan pengenal untuk profil yang sedang diperbarui.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `p-([0-9a-f]{17})`

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

ThrottlingException

Permintaan ditolak karena throttling permintaan.

Kode Status HTTP: 400

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

UpdateServer

Memperbarui properti server berkemampuan protokol transfer file setelah server tersebut dibuat.

`UpdateServer`Panggilan mengembalikan server `ServerId` yang Anda perbarui.

Sintaksis Permintaan

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

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

Certificate

Nama Sumber Daya Amazon (ARN) dari AWS sertifikat Certificate Manager (ACM). Diperlukan saat `Protocols` diatur ke `FTPS`.

Untuk meminta sertifikat publik baru, lihat Meminta sertifikat publik di Panduan Pengguna AWS Certificate Manager.

Untuk mengimpor sertifikat yang ada ke ACM, lihat Mengimpor sertifikat ke ACM di Panduan Pengguna AWS Certificate Manager.

Untuk meminta sertifikat pribadi menggunakan FTPS melalui alamat IP pribadi, lihat Meminta sertifikat pribadi di Panduan Pengguna AWS Certificate Manager.

Sertifikat dengan algoritme kriptografi dan ukuran kunci berikut didukung:

- 2048-bit RSA (RSA_2048)
- 4096-bit RSA (RSA_4096)
- Elliptic Prime Curve 256 bit (EC_prime256v1)
- Elliptic Prime Curve 384 bit (EC_secp384r1)

- Elliptic Prime Curve 521 bit (EC_secp521r1)

> **Note**
>
> Sertifikat harus berupa sertifikat SSL/TLS X.509 versi 3 yang valid dengan FQDN atau alamat IP yang ditentukan dan informasi tentang penerbitnya.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 1600.

Diperlukan: Tidak

EndpointDetails

Pengaturan titik akhir virtual private cloud (VPC) yang dikonfigurasi untuk server Anda. Ketika Anda meng-host titik akhir Anda dalam VPC Anda, Anda dapat membuat titik akhir Anda hanya dapat diakses oleh sumber daya dalam VPC Anda, atau Anda dapat melampirkan alamat IP Elastis dan membuat titik akhir Anda dapat diakses oleh klien melalui internet. Grup keamanan default VPC Anda secara otomatis ditetapkan ke titik akhir Anda.

Tipe: Objek EndpointDetails

Diperlukan: Tidak

EndpointType

Jenis endpoint yang Anda ingin server Anda gunakan. Anda dapat memilih untuk membuat endpoint server Anda dapat diakses publik (PUBLIK) atau menghostingnya di dalam VPC Anda. Dengan endpoint yang di-host di VPC, Anda dapat membatasi akses ke server dan sumber daya hanya dalam VPC Anda atau memilih untuk membuatnya menghadap internet dengan melampirkan alamat IP Elastis langsung ke sana.

> **Note**
>
> Setelah 19 Mei 2021, Anda tidak akan dapat membuat server menggunakan `EndpointType=VPC_ENDPOINT` di AWS akun Anda jika akun Anda belum melakukannya sebelum 19 Mei 2021. Jika Anda telah membuat server dengan `EndpointType=VPC_ENDPOINT` di AWS akun Anda pada atau sebelum 19 Mei 2021, Anda tidak akan terpengaruh. Setelah tanggal ini, gunakan `EndpointType =VPC`. Untuk informasi selengkapnya, lihat Menghentikan penggunaan VPC_ENDPOINT.

Direkomendasikan agar Anda menggunakan `VPC` sebagai `EndpointType`. Dengan jenis titik akhir ini, Anda memiliki pilihan untuk secara langsung mengaitkan hingga tiga alamat IPv4 Elastis (termasuk IP BYO) dengan titik akhir server Anda dan menggunakan grup keamanan VPC untuk membatasi lalu lintas berdasarkan alamat IP publik klien. Hal ini tidak mungkin terjadi jika `EndpointType` diatur ke `VPC_ENDPOINT`.

Jenis: String

Nilai yang Valid: `PUBLIC | VPC | VPC_ENDPOINT`

Diperlukan: Tidak

HostKey

Kunci pribadi RSA, ECDSA, atau ED25519 untuk digunakan untuk server berkemampuan SFTP Anda. Anda dapat menambahkan beberapa kunci host, jika Anda ingin memutar tombol, atau memiliki satu set kunci aktif yang menggunakan algoritma yang berbeda.

Gunakan perintah berikut untuk menghasilkan kunci RSA 2048 bit tanpa frasa sandi:

```
ssh-keygen -t rsa -b 2048 -N "" -m PEM -f my-new-server-key.
```

Gunakan nilai minimum 2048 untuk `-b` opsi. Anda dapat membuat kunci yang lebih kuat dengan menggunakan 3072 atau 4096.

Gunakan perintah berikut untuk menghasilkan kunci ECDSA 256 bit tanpa frasa sandi:

```
ssh-keygen -t ecdsa -b 256 -N "" -m PEM -f my-new-server-key.
```

Nilai yang valid untuk `-b` opsi ECDSA adalah 256, 384, dan 521.

Gunakan perintah berikut untuk menghasilkan kunci ED25519 tanpa frasa sandi:

```
ssh-keygen -t ed25519 -N "" -f my-new-server-key.
```

Untuk semua perintah ini, Anda dapat mengganti my-new-server-keydengan string pilihan Anda.

Important

Jika Anda tidak berencana untuk memigrasikan pengguna yang ada dari server berkemampuan SFTP yang sudah ada ke server baru, jangan perbarui kunci host. Mengubah kunci host server secara tidak sengaja dapat mengganggu.

Untuk informasi selengkapnya, lihat Memperbarui kunci host untuk server berkemampuan SFTP di Panduan Pengguna. AWS Transfer Family

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 4096.

Diperlukan: Tidak

IdentityProviderDetails

Array yang berisi semua informasi yang diperlukan untuk memanggil metode API otentikasi pelanggan.

Tipe: Objek IdentityProviderDetails

Diperlukan: Tidak

LoggingRole

Nama Sumber Daya Amazon (ARN) dari peran AWS Identity and Access Management (IAM) yang memungkinkan server mengaktifkan CloudWatch pencatatan Amazon untuk Amazon S3 atau Amazon EFSevents. Saat disetel, Anda dapat melihat aktivitas pengguna di CloudWatch log Anda.

Jenis: String

Batasan Panjang: Panjang minimum sebesar 0. Panjang maksimum 2048.

Pola: `(|arn:.*role/\S+)`

Diperlukan: Tidak

PostAuthenticationLoginBanner

Menentukan string untuk ditampilkan ketika pengguna terhubung ke server. String ini ditampilkan setelah pengguna mengautentikasi.

> **Note**
>
> Protokol SFTP tidak mendukung spanduk tampilan pasca-otentikasi.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 4096.

Pola: `[\x09-\x0D\x20-\x7E]*`

Diperlukan: Tidak

PreAuthenticationLoginBanner

Menentukan string untuk ditampilkan ketika pengguna terhubung ke server. String ini ditampilkan sebelum pengguna mengautentikasi. Misalnya, spanduk berikut menampilkan detail tentang penggunaan sistem:

```
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.
```

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 4096.

Pola: `[\x09-\x0D\x20-\x7E]*`

Diperlukan: Tidak

ProtocolDetails

Pengaturan protokol yang dikonfigurasi untuk server Anda.

- Untuk menunjukkan mode pasif (untuk protokol FTP dan FTPS), gunakan parameter. `PassiveIp` Masukkan satu alamat IPv4 bertitik quad, seperti alamat IP eksternal dari firewall, router, atau penyeimbang beban.
- Untuk mengabaikan kesalahan yang dihasilkan saat klien mencoba menggunakan `SETSTAT` perintah pada file yang Anda unggah ke bucket Amazon S3, gunakan `SetStatOption` parameternya. Agar AWS Transfer Family server mengabaikan `SETSTAT` perintah dan mengunggah file tanpa perlu membuat perubahan apa pun pada klien SFTP Anda, tetapkan nilainya. `ENABLE_NO_OP` Jika Anda menyetel `SetStatOption` parameternya`ENABLE_NO_OP`, Transfer Family akan menghasilkan entri CloudWatch log ke Amazon Logs, sehingga Anda dapat menentukan kapan klien melakukan `SETSTAT` panggilan.
- Untuk menentukan apakah AWS Transfer Family server Anda melanjutkan sesi terbaru yang dinegosiasikan melalui ID sesi unik, gunakan parameternya. `TlsSessionResumptionMode`

- `As2Transports`menunjukkan metode transport untuk pesan AS2. Saat ini, hanya HTTP yang didukung.

Tipe: Objek ProtocolDetails

Diperlukan: Tidak

Protocols

Menentukan protokol transfer file atau protokol di mana klien protokol transfer file Anda dapat terhubung ke titik akhir server Anda. Protokol yang tersedia adalah:

- `SFTP` (Secure Shell (SSH) Protokol Transfer File): Transfer file melalui SSH
- `FTPS` (File Transfer Protocol Secure): Transfer file dengan enkripsi TLS
- `FTP` (Protokol Transfer File): Transfer file tidak terenkripsi
- `AS2`(Pernyataan Penerapan 2): digunakan untuk mengangkut data terstruktur business-to-business

> **Note**
>
> - Jika Anda memilih`FTPS`, Anda harus memilih sertifikat yang disimpan di AWS Certificate Manager (ACM) yang digunakan untuk mengidentifikasi server Anda ketika klien terhubung ke sana melalui FTPS.
> - Jika `Protocol` termasuk salah satu `FTP` atau`FTPS`, maka `EndpointType` harus `VPC` dan `IdentityProviderType` harus baik`AWS_DIRECTORY_SERVICE`,`AWS_LAMBDA`, atau`API_GATEWAY`.
> - Jika `Protocol` termasuk `FTP`, maka `AddressAllocationIds` tidak dapat dikaitkan.
> - Jika `Protocol` disetel hanya ke`SFTP`, `EndpointType` dapat diatur ke `PUBLIC` dan `IdentityProviderType` dapat disetel salah satu jenis identitas yang didukung:`SERVICE_MANAGED`,`AWS_DIRECTORY_SERVICE`,`AWS_LAMBDA`, atau`API_GATEWAY`.
> - Jika `Protocol` termasuk`AS2`, maka `EndpointType` harus`VPC`, dan domain harus Amazon S3.

Tipe: Array string

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 4 item.

Nilai yang Valid: `SFTP | FTP | FTPS | AS2`

Diperlukan: Tidak

S3StorageOptions

Menentukan apakah atau tidak kinerja untuk direktori Amazon S3 Anda dioptimalkan. Ini dinonaktifkan secara default.

Secara default, pemetaan direktori home memiliki `TYPE` file. `DIRECTORY` Jika Anda mengaktifkan opsi ini, Anda kemudian perlu secara eksplisit menyetel `HomeDirectoryMapEntry` `Type` ke `FILE` jika Anda ingin pemetaan memiliki target file.

Tipe: Objek S3StorageOptions

Diperlukan: Tidak

SecurityPolicyName

Menentukan nama kebijakan keamanan untuk server.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 100.

Pola: `Transfer[A-Za-z0-9]*SecurityPolicy-[A-Za-z0-9-]+`

Diperlukan: Tidak

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk instance server yang ditetapkan oleh pengguna Transfer Family.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

StructuredLogDestinations

Menentukan grup log yang log server Anda dikirim.

Untuk menentukan grup log, Anda harus memberikan ARN untuk grup log yang ada. Dalam hal ini, format grup log adalah sebagai berikut:

`arn:aws:logs:region-name:amazon-account-id:log-group:log-group-name:*`

Misalnya, `arn:aws:logs:us-east-1:111122223333:log-group:mytestgroup:*`

Jika sebelumnya Anda telah menentukan grup log untuk server, Anda dapat menghapusnya, dan pada dasarnya mematikan logging terstruktur, dengan memberikan nilai kosong untuk parameter ini dalam `update-server` panggilan. Sebagai contoh:

```
update-server --server-id s-1234567890abcdef0 --structured-log-
destinations
```

Tipe: Array string.

Anggota Array: Jumlah minimum 0 item. Jumlah maksimum 1 item.

Batasan Panjang: Panjang minimum 20. Panjang maksimum 1600.

Pola: `arn:\S+`

Diperlukan: Tidak

WorkflowDetails

Menentukan ID alur kerja untuk alur kerja yang akan ditetapkan dan peran eksekusi yang digunakan untuk mengeksekusi alur kerja.

Selain alur kerja untuk mengeksekusi ketika file diunggah sepenuhnya, juga `WorkflowDetails` dapat berisi ID alur kerja (dan peran eksekusi) untuk alur kerja untuk mengeksekusi pada upload sebagian. Upload sebagian terjadi ketika sesi server terputus saat file masih diunggah.

Untuk menghapus alur kerja terkait dari server, Anda dapat memberikan `OnUpload` objek kosong, seperti pada contoh berikut.

```
aws transfer update-server --server-id s-01234567890abcdef --workflow-
details '{"OnUpload":[]}'
```

Tipe: Objek WorkflowDetails

Wajib: Tidak

Sintaksis Respons

```
{
   "ServerId": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk server yang ditetapkan oleh pengguna Transfer Family.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

AccessDeniedException

Anda tidak memiliki akses yang memadai untuk melakukan tindakan ini.

Kode Status HTTP: 400

ConflictException

Pengecualian ini dilemparkan ketika `UpdateServer` dipanggil untuk server berkemampuan protokol transfer file yang memiliki VPC sebagai tipe titik akhir dan server tidak dalam keadaan `VpcEndpointID` tersedia.

Kode Status HTTP: 400

InternalServiceError

Pengecualian ini dilemparkan ketika kesalahan terjadi dalam AWS Transfer Family layanan.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceExistsException

Sumber daya yang diminta tidak ada, atau ada di wilayah selain yang ditentukan untuk perintah.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

ThrottlingException

Permintaan ditolak karena throttling permintaan.

Kode Status HTTP: 400

Contoh-contoh

Contoh

Contoh berikut memperbarui peran server.

Permintaan Sampel

```
{
   "EndpointDetails": {
   "VpcEndpointId": "vpce-01234f056f3g13",
```

```
   "LoggingRole": "CloudWatchS3Events",
   "ServerId": "s-01234567890abcdef"
   }
}
```

Contoh

Contoh berikut menghapus alur kerja terkait dari server.

Permintaan Sampel

```
aws transfer update-server --server-id s-01234567890abcdef --workflow-details
 '{"OnUpload":[]}'
```

Contoh

Ini adalah contoh respons untuk panggilan API ini.

Contoh Respons

```
{
   "ServerId": "s-01234567890abcdef"
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Antarmuka Baris Perintah
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK untuk Python
- AWS SDK for Ruby V3

UpdateUser

Menetapkan properti baru untuk pengguna. Parameter yang Anda lewati mengubah salah satu atau semua hal berikut: direktori home, peran, dan kebijakan untuk `UserName` dan yang `ServerId` Anda tentukan.

Respons mengembalikan `ServerId` dan `UserName` untuk pengguna yang diperbarui.

Di konsol, Anda dapat memilih Dibatasi saat membuat atau memperbarui pengguna. Ini memastikan bahwa pengguna tidak dapat mengakses apa pun di luar direktori home mereka. Cara terprogram untuk mengonfigurasi perilaku ini adalah dengan memperbarui pengguna. Setel `HomeDirectoryType` ke`LOGICAL`, dan tentukan `HomeDirectoryMappings` dengan `Entry` as root (`/`) dan `Target` sebagai direktori home mereka.

Misalnya, jika direktori home pengguna`/test/admin-user`, perintah berikut akan memperbarui pengguna sehingga konfigurasi mereka di konsol menunjukkan flag Restricted seperti yang dipilih.

```
aws transfer update-user --server-id <server-id> --user-name admin-user --
home-directory-type LOGICAL --home-directory-mappings "[{\"Entry\":\"/\",
\"Target\":\"/test/admin-user\"}]"
```

Sintaksis Permintaan

```
{
   "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"
}
```

Parameter Permintaan

Untuk informasi tentang parameter yang umum untuk semua tindakan, lihat Parameter Umum.

Permintaan menerima data berikut dalam format JSON.

HomeDirectory

Direktori arahan (folder) untuk pengguna ketika mereka masuk ke server menggunakan klien.

Contoh `HomeDirectory` adalah `/bucket_name/home/mydirectory`.

> **Note**
>
> Parameter `HomeDirectory` hanya digunakan jika `HomeDirectoryType` diatur ke `PATH`.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 1024.

Pola: `(|/.*)`

Diperlukan: Tidak

HomeDirectoryMappings

Pemetaan direktori logis yang menentukan jalur dan kunci Amazon S3 atau Amazon EFS apa yang harus terlihat oleh pengguna Anda dan bagaimana Anda ingin membuatnya terlihat. Anda harus menentukan `Entry` dan `Target` memasangkan, di mana `Entry` menunjukkan bagaimana jalur dibuat terlihat dan `Target` merupakan jalur Amazon S3 atau Amazon EFS yang sebenarnya. Jika Anda hanya menentukan target, itu ditampilkan apa adanya. Anda juga harus memastikan bahwa peran AWS Identity and Access Management (IAM) Anda menyediakan akses ke jalur masuk`Target`. Nilai ini dapat diatur hanya ketika `HomeDirectoryType` diatur ke LOGICAL.

Berikut ini adalah contoh `Entry` dan `Target` pair.

```
[ { "Entry": "/directory1", "Target": "/bucket_name/home/
mydirectory" } ]
```

Dalam kebanyakan kasus, Anda dapat menggunakan nilai ini alih-alih kebijakan sesi untuk mengunci pengguna Anda ke direktori home yang ditunjuk (`chroot`“”). Untuk melakukan ini, Anda dapat mengatur `Entry` ke '/' dan mengatur `Target` ke nilai HomeDirectory parameter.

Berikut ini adalah contoh `Entry` dan `Target` pair untuk`chroot`.

```
[ { "Entry": "/", "Target": "/bucket_name/home/mydirectory" } ]
```

Tipe: Array objek HomeDirectoryMapEntry

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50000 item.

Diperlukan: Tidak

HomeDirectoryType

Jenis direktori pendaratan (folder) yang Anda inginkan direktori home pengguna Anda ketika mereka masuk ke server. Jika Anda mengaturnya`PATH`, pengguna akan melihat bucket Amazon S3 absolut atau jalur Amazon EFS seperti pada klien protokol transfer file mereka. Jika Anda menyetelnya`LOGICAL`, Anda harus menyediakan pemetaan `HomeDirectoryMappings` untuk bagaimana Anda ingin membuat jalur Amazon S3 atau Amazon EFS terlihat oleh pengguna Anda.

> **Note**
>
> Jika `HomeDirectoryType` ya`LOGICAL`, Anda harus memberikan pemetaan, menggunakan parameter. `HomeDirectoryMappings` Jika, di sisi lain, `HomeDirectoryType` adalah`PATH`, Anda memberikan jalur absolut menggunakan `HomeDirectory` parameter. Anda tidak dapat memiliki keduanya `HomeDirectory` dan `HomeDirectoryMappings` di template Anda.

Jenis: String

Nilai yang Valid: `PATH | LOGICAL`

Diperlukan: Tidak

Policy

Kebijakan sesi untuk pengguna Anda sehingga Anda dapat menggunakan peran yang sama AWS Identity and Access Management (IAM) di beberapa pengguna. Kebijakan ini mencakup akses pengguna ke sebagian bucket Amazon S3 mereka. Variabel yang dapat Anda gunakan

dalam kebijakan ini meliputi `${Transfer:UserName}`, `${Transfer:HomeDirectory}`, dan `${Transfer:HomeBucket}`.

> **Note**
>
> Kebijakan ini hanya berlaku jika domainnya `ServerId` adalah Amazon S3. Amazon EFS tidak menggunakan kebijakan sesi.
> Untuk kebijakan sesi, AWS Transfer Family menyimpan kebijakan sebagai gumpalan JSON, bukan Nama Sumber Daya Amazon (ARN) kebijakan tersebut. Anda menyimpan kebijakan sebagai blob JSON dan meneruskan dalam argumen `Policy`.
> Untuk contoh kebijakan sesi, lihat Contoh kebijakan sesi.
> Untuk informasi selengkapnya, lihat AssumeRoledi Referensi API Layanan Token AWS Keamanan.

Jenis: String

Batasan Panjang: Panjang minimum sebesar 0. Panjang maksimum 2048.

Diperlukan: Tidak

PosixProfile

Menentukan identitas POSIX lengkap, termasuk ID pengguna (`Uid`), ID grup (`Gid`), dan ID grup sekunder apa pun (`SecondaryGids`), yang mengontrol akses pengguna Anda ke Amazon Elastic File Systems (Amazon EFS). Izin POSIX yang ditetapkan pada file dan direktori dalam sistem file Anda menentukan tingkat akses yang didapat pengguna Anda saat mentransfer file masuk dan keluar dari sistem file Amazon EFS Anda.

Tipe: Objek PosixProfile

Diperlukan: Tidak

Role

Nama Sumber Daya Amazon (ARN) dari peran AWS Identity and Access Management (IAM) yang mengontrol akses pengguna ke bucket Amazon S3 atau sistem file Amazon EFS. Kebijakan yang dilampirkan pada peran ini menentukan tingkat akses yang ingin Anda berikan kepada pengguna saat mentransfer file masuk dan keluar dari bucket Amazon S3 atau sistem file Amazon EFS. IAM role juga harus berisi hubungan kepercayaan yang mengizinkan server untuk mengakses sumber daya Anda saat melayani permintaan transfer pengguna.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 2048.

Pola: `arn:.*role/\S+`

Diperlukan: Tidak

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk instance server Transfer Family yang ditetapkan oleh pengguna.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

UserName

String unik yang mengidentifikasi pengguna dan dikaitkan dengan server seperti yang ditentukan oleh. `ServerId` Nama pengguna ini harus sepanjang minimal 3 dan maksimal 100 karakter. Berikut adalah karakter yang valid: a-z, A-Z, 0-9, garis bawah '_', tanda hubung '-', titik '.', dan tanda at '@'. Nama pengguna tidak dapat dimulai dengan tanda hubung, titik, atau tanda at.

Jenis: String

Batasan Panjang: Panjang minimum 3. Panjang maksimum 100.

Pola: `[\w][\w@.-]{2,99}`

Diperlukan: Ya

Sintaksis Respons

```
{
   "ServerId": "string",
   "UserName": "string"
}
```

Elemen Respons

Jika tindakan berhasil, layanan mengirimkan kembali respons HTTP 200.

Layanan mengembalikan data berikut dalam format JSON.

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk instance server Transfer Family tempat akun ditetapkan.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

UserName

Pengidentifikasi unik untuk pengguna yang ditugaskan ke instance server yang ditentukan dalam permintaan.

Jenis: String

Batasan Panjang: Panjang minimum 3. Panjang maksimum 100.

Pola: `[\w][\w@.-]{2,99}`

Kesalahan

Untuk informasi tentang kesalahan yang umum untuk semua tindakan, lihat Kesalahan Umum.

InternalServiceError

Pengecualian ini dilontarkan ketika terjadi kesalahan dalam layanan AWS Transfer Family.

Kode Status HTTP: 500

InvalidRequestException

Pengecualian ini dilontarkan ketika klien mengirimkan permintaan yang salah format.

Kode Status HTTP: 400

ResourceNotFoundException

Pengecualian ini dilemparkan ketika sumber daya tidak ditemukan oleh layanan AWS Transfer Family.

Kode Status HTTP: 400

ServiceUnavailableException

Permintaan gagal karena layanan AWS Transfer Family tidak tersedia.

Kode Status HTTP: 500

ThrottlingException

Permintaan ditolak karena throttling permintaan.

Kode Status HTTP: 400

Contoh-contoh

Contoh

Contoh berikut memperbarui pengguna Transfer Family.

Permintaan Sampel

```
{
   "HomeDirectory": "/bucket2/documentation",
   "HomeDirectoryMappings": [
      {
         "Entry": "/directory1",
         "Target": "/bucket_name/home/mydirectory"
      }
   ],
   "HomeDirectoryType:" "PATH",
   "Role": "AssumeRole",
   "ServerId": "s-01234567890abcdef",
   "UserName": "my_user"
}
```

Contoh

Ini adalah contoh respons untuk panggilan API ini.

Contoh Respons

```
{
   "ServerId": "s-01234567890abcdef",
   "UserName": "my_user"
}
```

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS Command Line Interface
- AWS SDK for .NET
- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWSSDK untuk V3 JavaScript
- AWS SDK for PHP V3
- AWS SDK for Python
- AWS SDK for Ruby V3

Tipe Data

tipe data berikut didukung:

- 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

Berisi rincian untuk objek konektor AS2. Objek konektor digunakan untuk proses keluar AS2, untuk menghubungkan AWS Transfer Family pelanggan dengan mitra dagang.

Daftar Isi

BasicAuthSecretId

Menyediakan dukungan otentikasi Dasar ke AS2 Connectors API. Untuk menggunakan otentikasi Dasar, Anda harus memberikan nama atau Nama Sumber Daya Amazon (ARN) rahasia di. AWS Secrets Manager

Nilai default untuk parameter ini adalah`null`, yang menunjukkan bahwa otentikasi Dasar tidak diaktifkan untuk konektor.

Jika konektor harus menggunakan otentikasi Dasar, rahasianya harus dalam format berikut:

```
{ "Username": "user-name", "Password": "user-password" }
```

Ganti `user-name` dan `user-password` dengan kredensi untuk pengguna sebenarnya yang sedang diautentikasi.

Perhatikan hal berikut:

- Anda menyimpan kredensi ini di Secrets Manager, tidak meneruskannya langsung ke API ini.
- Jika Anda menggunakan API, SDK, atau CloudFormation untuk mengkonfigurasi konektor Anda, maka Anda harus membuat rahasia sebelum Anda dapat mengaktifkan otentikasi Dasar. Namun, jika Anda menggunakan konsol AWS manajemen, Anda dapat meminta sistem membuat rahasia untuk Anda.

Jika sebelumnya Anda telah mengaktifkan otentikasi Dasar untuk konektor, Anda dapat menonaktifkannya dengan menggunakan panggilan `UpdateConnector` API. Misalnya, jika Anda menggunakan CLI, Anda dapat menjalankan perintah berikut untuk menghapus otentikasi Dasar:

```
update-connector --connector-id my-connector-id --as2-config
'BasicAuthSecretId=""'
```

Jenis: String

Batasan Panjang: Panjang minimum sebesar 0. Panjang maksimum 2048.

Diperlukan: Tidak

Compression

Menentukan apakah file AS2 dikompresi.

Jenis: String

Nilai yang Valid: `ZLIB | DISABLED`

Diperlukan: Tidak

EncryptionAlgorithm

Algoritma yang digunakan untuk mengenkripsi file.

Perhatikan hal berikut:

- Jangan gunakan `DES_EDE3_CBC` algoritma kecuali Anda harus mendukung klien lama yang membutuhkannya, karena ini adalah algoritma enkripsi yang lemah.
- Anda hanya dapat menentukan `NONE` apakah URL untuk konektor Anda menggunakan HTTPS. Menggunakan HTTPS memastikan bahwa tidak ada lalu lintas yang dikirim dalam teks yang jelas.

Jenis: String

Nilai yang Valid: `AES128_CBC | AES192_CBC | AES256_CBC | DES_EDE3_CBC | NONE`

Diperlukan: Tidak

LocalProfileId

Pengidentifikasi unik untuk profil lokal AS2.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `p-([0-9a-f]{17})`

Diperlukan: Tidak

MdnResponse

Digunakan untuk permintaan keluar (dari AWS Transfer Family server ke server AS2 mitra) untuk menentukan apakah respons mitra untuk transfer sinkron atau asinkron. Tentukan salah satu dari nilai berikut:

- SYNC: Sistem mengharapkan respons MDN sinkron, mengkonfirmasikan bahwa file berhasil ditransfer (atau tidak).
- NONE: Menentukan bahwa tidak ada respon MDN diperlukan.

Jenis: String

Nilai yang Valid: SYNC | NONE

Diperlukan: Tidak

MdnSigningAlgorithm

Algoritma penandatanganan untuk respons MDN.

> **Note**
>
> Jika disetel ke DEFAULT (atau tidak disetel sama sekali), nilai untuk SigningAlgorithm digunakan.

Jenis: String

Nilai yang Valid: SHA256 | SHA384 | SHA512 | SHA1 | NONE | DEFAULT

Diperlukan: Tidak

MessageSubject

Digunakan sebagai atribut header Subject HTTP dalam pesan AS2 yang sedang dikirim dengan konektor.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: [\p{Print}\p{Blank}]+

Diperlukan: Tidak

PartnerProfileId

Pengenal unik untuk profil mitra untuk konektor.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `p-([0-9a-f]{17})`

Diperlukan: Tidak

SigningAlgorithm

Algoritma yang digunakan untuk menandatangani pesan AS2 yang dikirim dengan konektor.

Jenis: String

Nilai yang Valid: `SHA256 | SHA384 | SHA512 | SHA1 | NONE`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

CopyStepDetails

Setiap tipe langkah memiliki `StepDetails` strukturnya sendiri.

Daftar Isi

DestinationFileLocation

Menentukan lokasi untuk file yang sedang disalin. Gunakan `${Transfer:UserName}` atau `${Transfer:UploadDate}` di bidang ini untuk membuat parameter awalan tujuan berdasarkan nama pengguna atau tanggal upload.

- Tetapkan nilai `DestinationFileLocation` `${Transfer:UserName}` to untuk menyalin file yang diunggah ke bucket Amazon S3 yang diawali dengan nama pengguna Transfer Family yang mengunggah file tersebut.
- Tetapkan nilai `DestinationFileLocation` `${Transfer:UploadDate}` to untuk menyalin file yang diunggah ke bucket Amazon S3 yang diawali dengan tanggal unggahan.

> **Note**
>
> Sistem menyelesaikan `UploadDate` ke format tanggal YYYY-MM-DD, berdasarkan tanggal file diunggah dalam UTC.

Tipe: Objek InputFileLocation

Diperlukan: Tidak

Name

Nama langkah, digunakan sebagai pengenal.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 30.

Pola: `[\w-]*`

Diperlukan: Tidak

OverwriteExisting

Bendera yang menunjukkan apakah akan menimpa file yang ada dengan nama yang sama. Default-nya adalah `FALSE`.

Jika alur kerja memproses file yang memiliki nama yang sama dengan file yang ada, perilakunya adalah sebagai berikut:

- Jika `OverwriteExisting` ya`TRUE`, file yang ada diganti dengan file yang sedang diproses.
- Jika `OverwriteExisting` ya`FALSE`, tidak ada yang terjadi, dan pemrosesan alur kerja berhenti.

Jenis: String

Nilai yang Valid: `TRUE | FALSE`

Diperlukan: Tidak

SourceFileLocation

Menentukan file mana yang akan digunakan sebagai masukan ke langkah alur kerja: baik output dari langkah sebelumnya, atau file yang awalnya diunggah untuk alur kerja.

- Untuk menggunakan file sebelumnya sebagai input, masukkan`${previous.file}`. Dalam hal ini, langkah alur kerja ini menggunakan file output dari langkah alur kerja sebelumnya sebagai input. Ini adalah nilai default.
- Untuk menggunakan lokasi file yang awalnya diunggah sebagai masukan untuk langkah ini, masukkan`${original.file}`.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 256.

Pola: `\$\{(\w+.)+\w+\}`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

CustomStepDetails

Setiap tipe langkah memiliki `StepDetails` strukturnya sendiri.

Daftar Isi

Name

Nama langkah, digunakan sebagai pengenal.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 30.

Pola: `[\w-]*`

Diperlukan: Tidak

SourceFileLocation

Menentukan file mana yang akan digunakan sebagai masukan ke langkah alur kerja: baik output dari langkah sebelumnya, atau file yang awalnya diunggah untuk alur kerja.

- Untuk menggunakan file sebelumnya sebagai input, masukkan`${previous.file}`. Dalam hal ini, langkah alur kerja ini menggunakan file output dari langkah alur kerja sebelumnya sebagai input. Ini adalah nilai default.
- Untuk menggunakan lokasi file yang awalnya diunggah sebagai masukan untuk langkah ini, masukkan`${original.file}`.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 256.

Pola: `\$\{(\w+.)+\w+\}`

Diperlukan: Tidak

Target

ARN untuk fungsi Lambda yang sedang dipanggil.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 170.

Pola: `arn:[a-z-]+:lambda:.*`

Diperlukan: Tidak

TimeoutSeconds

Batas waktu, dalam hitungan detik, untuk langkah tersebut.

Jenis: Integer

Rentang yang Valid: Nilai minimum 1. Nilai maksimum 1800.

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

DecryptStepDetails

Setiap tipe langkah memiliki `StepDetails` strukturnya sendiri.

Daftar Isi

DestinationFileLocation

Menentukan lokasi untuk file yang didekripsi. Gunakan `${Transfer:UserName}` atau `${Transfer:UploadDate}` di bidang ini untuk membuat parameter awalan tujuan berdasarkan nama pengguna atau tanggal upload.

- Tetapkan nilai `DestinationFileLocation` `${Transfer:UserName}` to untuk mendekripsi file yang diunggah ke bucket Amazon S3 yang diawali dengan nama pengguna Transfer Family yang mengunggah file tersebut.
- Tetapkan nilai `DestinationFileLocation` `${Transfer:UploadDate}` to untuk mendekripsi file yang diunggah ke bucket Amazon S3 yang diawali dengan tanggal unggahan.

> **Note**
>
> Sistem menyelesaikan `UploadDate` ke format tanggal YYYY-MM-DD, berdasarkan tanggal file diunggah dalam UTC.

Tipe: Objek InputFileLocation

Wajib: Ya

Type

Jenis enkripsi yang digunakan. Saat ini, nilai ini harus`PGP`.

Jenis: String

Nilai yang Valid: `PGP`

Diperlukan: Ya

Name

Nama langkah, digunakan sebagai pengenal.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 30.

Pola: `[\w-]*`

Diperlukan: Tidak

OverwriteExisting

Bendera yang menunjukkan apakah akan menimpa file yang ada dengan nama yang sama. Default-nya adalah `FALSE`.

Jika alur kerja memproses file yang memiliki nama yang sama dengan file yang ada, perilakunya adalah sebagai berikut:

- Jika `OverwriteExisting` ya`TRUE`, file yang ada diganti dengan file yang sedang diproses.
- Jika `OverwriteExisting` ya`FALSE`, tidak ada yang terjadi, dan pemrosesan alur kerja berhenti.

Jenis: String

Nilai yang Valid: `TRUE | FALSE`

Diperlukan: Tidak

SourceFileLocation

Menentukan file mana yang akan digunakan sebagai masukan ke langkah alur kerja: baik output dari langkah sebelumnya, atau file yang awalnya diunggah untuk alur kerja.

- Untuk menggunakan file sebelumnya sebagai input, masukkan`${previous.file}`. Dalam hal ini, langkah alur kerja ini menggunakan file output dari langkah alur kerja sebelumnya sebagai input. Ini adalah nilai default.
- Untuk menggunakan lokasi file yang awalnya diunggah sebagai masukan untuk langkah ini, masukkan`${original.file}`.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 256.

Pola: `\$\{(\w+.)+\w+\}`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

DeleteStepDetails

Nama langkah, digunakan untuk mengidentifikasi langkah hapus.

Daftar Isi

Name

Nama langkah, digunakan sebagai pengenal.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 30.

Pola: `[\w-]*`

Diperlukan: Tidak

SourceFileLocation

Menentukan file mana yang akan digunakan sebagai masukan ke langkah alur kerja: baik output dari langkah sebelumnya, atau file yang awalnya diunggah untuk alur kerja.

- Untuk menggunakan file sebelumnya sebagai input, masukkan`${previous.file}`. Dalam hal ini, langkah alur kerja ini menggunakan file output dari langkah alur kerja sebelumnya sebagai input. Ini adalah nilai default.
- Untuk menggunakan lokasi file yang awalnya diunggah sebagai masukan untuk langkah ini, masukkan`${original.file}`.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 256.

Pola: `\$\{(\w+.)+\w+\}`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++

- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

DescribedAccess

Menjelaskan properti akses yang ditentukan.

Daftar Isi

ExternalId

Pengidentifikasi unik yang diperlukan untuk mengidentifikasi grup tertentu dalam direktori Anda. Pengguna grup yang Anda asosiasikan memiliki akses ke sumber daya Amazon S3 atau Amazon EFS Anda melalui protokol yang diaktifkan. AWS Transfer Family Jika Anda tahu nama grup, Anda dapat melihat nilai SID dengan menjalankan perintah berikut menggunakan Windows PowerShell.

```
Get-ADGroup -Filter {samAccountName -like "YourGroupName*"} -Properties
* | Select SamAccountName,ObjectSid
```

Dalam perintah itu, ganti YourGroupNamedengan nama grup Active Directory Anda.

Ekspresi reguler yang digunakan untuk memvalidasi parameter ini adalah string karakter yang terdiri dari huruf besar dan huruf kecil karakter alfanumerik tanpa spasi. Anda juga dapat menyertakan garis bawah atau salah satu karakter berikut: =, . @: /-

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `S-1-[\d-]+`

Diperlukan: Tidak

HomeDirectory

Direktori arahan (folder) untuk pengguna ketika mereka masuk ke server menggunakan klien.

Contoh `HomeDirectory` adalah `/bucket_name/home/mydirectory`.

> **Note**
>
> Parameter `HomeDirectory` hanya digunakan jika `HomeDirectoryType` diatur ke `PATH`.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 1024.

Pola: `(|/.*)`

Diperlukan: Tidak

HomeDirectoryMappings

Pemetaan direktori logis yang menentukan jalur dan kunci Amazon S3 atau Amazon EFS apa yang harus terlihat oleh pengguna Anda dan bagaimana Anda ingin membuatnya terlihat. Anda harus menentukan `Entry` dan `Target` memasangkan, di mana `Entry` menunjukkan bagaimana jalur dibuat terlihat dan `Target` merupakan jalur Amazon S3 atau Amazon EFS yang sebenarnya. Jika Anda hanya menentukan target, itu ditampilkan apa adanya. Anda juga harus memastikan bahwa peran AWS Identity and Access Management (IAM) Anda menyediakan akses ke jalur masuk`Target`. Nilai ini dapat diatur hanya ketika `HomeDirectoryType` diatur ke LOGICAL.

Dalam kebanyakan kasus, Anda dapat menggunakan nilai ini alih-alih kebijakan sesi untuk mengunci akses terkait ke direktori home yang ditunjuk (" `chroot` "). Untuk melakukan ini, Anda dapat mengatur `Entry` ke '/' dan mengatur `Target` ke nilai `HomeDirectory` parameter.

Tipe: Array objek HomeDirectoryMapEntry

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50000 item.

Diperlukan: Tidak

HomeDirectoryType

Jenis direktori pendaratan (folder) yang Anda inginkan direktori home pengguna Anda ketika mereka masuk ke server. Jika Anda mengaturnya`PATH`, pengguna akan melihat bucket Amazon S3 absolut atau jalur Amazon EFS seperti pada klien protokol transfer file mereka. Jika Anda menyetelnya`LOGICAL`, Anda harus menyediakan pemetaan `HomeDirectoryMappings` untuk bagaimana Anda ingin membuat jalur Amazon S3 atau Amazon EFS terlihat oleh pengguna Anda.

> **Note**
>
> Jika `HomeDirectoryType` ya`LOGICAL`, Anda harus memberikan pemetaan, menggunakan parameter. `HomeDirectoryMappings` Jika, di sisi lain, `HomeDirectoryType` adalah`PATH`, Anda memberikan jalur absolut menggunakan `HomeDirectory` parameter. Anda tidak dapat memiliki keduanya `HomeDirectory` dan `HomeDirectoryMappings` di template Anda.

Jenis: String

Nilai yang Valid: `PATH | LOGICAL`

Diperlukan: Tidak

Policy

Kebijakan sesi untuk pengguna Anda sehingga Anda dapat menggunakan peran yang sama AWS Identity and Access Management (IAM) di beberapa pengguna. Kebijakan ini mencakup akses pengguna ke sebagian bucket Amazon S3 mereka. Variabel yang dapat Anda gunakan dalam kebijakan ini meliputi `${Transfer:UserName}`, `${Transfer:HomeDirectory}`, dan `${Transfer:HomeBucket}`.

Jenis: String

Batasan Panjang: Panjang minimum sebesar 0. Panjang maksimum 2048.

Diperlukan: Tidak

PosixProfile

Identitas POSIX lengkap, termasuk ID pengguna (`Uid`), ID grup (`Gid`), dan setiap grup sekunder ID (`SecondaryGids`), yang mengendalikan akses pengguna Anda ke sistem file Amazon EFS Anda. POSIX izin yang ditetapkan pada file dan direktori dalam sistem file Anda menentukan tingkat akses yang pengguna Anda dapatkan ketika mentransfer file ke dalam dan keluar dari sistem file Amazon EFS Anda.

Tipe: Objek PosixProfile

Diperlukan: Tidak

Role

Nama Sumber Daya Amazon (ARN) dari peran AWS Identity and Access Management (IAM) yang mengontrol akses pengguna ke bucket Amazon S3 atau sistem file Amazon EFS. Kebijakan yang dilampirkan pada peran ini menentukan tingkat akses yang ingin Anda berikan kepada pengguna saat mentransfer file masuk dan keluar dari bucket Amazon S3 atau sistem file Amazon EFS Anda. IAM role juga harus berisi hubungan kepercayaan yang mengizinkan server untuk mengakses sumber daya Anda saat melayani permintaan transfer pengguna.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 2048.

Pola: `arn:.*role/\S+`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

DescribedAgreement

Menjelaskan sifat-sifat suatu perjanjian.

Daftar Isi

Arn

Nama Sumber Daya Amazon (ARN) yang unik untuk perjanjian tersebut.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 1600.

Pola: `arn:\S+`

Diperlukan: Ya

AccessRole

Konektor digunakan untuk mengirim file menggunakan protokol AS2 atau SFTP. Untuk peran akses, berikan Nama Sumber Daya Amazon (ARN) AWS Identity and Access Management peran yang akan digunakan.

Untuk konektor AS2

Dengan AS2, Anda dapat mengirim file dengan memanggil `StartFileTransfer` dan menentukan jalur file dalam parameter permintaan,. `SendFilePaths` Kami menggunakan direktori induk file (misalnya, untuk, direktori induk`/bucket/dir/`) untuk `--send-file-paths /bucket/dir/file.txt` sementara menyimpan file pesan AS2 yang diproses, menyimpan MDN ketika kami menerimanya dari mitra, dan menulis file JSON akhir yang berisi metadata transmisi yang relevan. Jadi, `AccessRole` kebutuhan untuk menyediakan akses baca dan tulis ke direktori induk dari lokasi file yang digunakan dalam `StartFileTransfer` permintaan. Selain itu, Anda perlu menyediakan akses baca dan tulis ke direktori induk dari file yang ingin Anda kirim`StartFileTransfer`.

Jika Anda menggunakan otentikasi Dasar untuk konektor AS2 Anda, peran akses memerlukan `secretsmanager:GetSecretValue` izin untuk rahasia tersebut. Jika rahasia dienkripsi menggunakan kunci yang dikelola pelanggan alih-alih kunci yang dikelola di AWS Secrets Manager, maka peran tersebut juga memerlukan `kms:Decrypt` izin untuk kunci tersebut.

Untuk konektor SFTP

Pastikan bahwa peran akses menyediakan akses baca dan tulis ke direktori induk dari lokasi file yang digunakan dalam `StartFileTransfer` permintaan. Selain itu, pastikan bahwa peran tersebut memberikan `secretsmanager:GetSecretValue` izin untukAWS Secrets Manager.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 2048.

Pola: `arn:.*role/\S+`

Diperlukan: Tidak

AgreementId

Pengidentifikasi unik untuk perjanjian. Pengenal ini dikembalikan saat Anda membuat perjanjian.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `a-([0-9a-f]{17})`

Diperlukan: Tidak

BaseDirectory

Direktori pendaratan (folder) untuk file yang ditransfer dengan menggunakan protokol AS2.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 1024.

Pola: `(|/.*)`

Diperlukan: Tidak

Description

Nama atau deskripsi singkat yang digunakan untuk mengidentifikasi perjanjian.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 200.

Pola: `[\p{Graph}]+`

Diperlukan: Tidak

LocalProfileId

Pengidentifikasi unik untuk profil lokal AS2.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `p-([0-9a-f]{17})`

Diperlukan: Tidak

PartnerProfileId

Pengenal unik untuk profil mitra yang digunakan dalam perjanjian.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `p-([0-9a-f]{17})`

Diperlukan: Tidak

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk instans server. Pengenal ini menunjukkan server spesifik yang digunakan perjanjian.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Tidak

Status

Status perjanjian saat ini, baik `ACTIVE` atau`INACTIVE`.

Jenis: String

Nilai yang Valid: `ACTIVE | INACTIVE`

Diperlukan: Tidak

Tags

Pasangan nilai kunci yang dapat digunakan untuk mengelompokkan dan mencari perjanjian.

Tipe: Array objek Tag

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50 item.

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

DescribedCertificate

Menjelaskan properti sertifikat.

Daftar Isi

Arn

Nama Sumber Daya Amazon (ARN) unik untuk sertifikat.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 1600.

Pola: `arn:\S+`

Diperlukan: Ya

ActiveDate

Tanggal opsional yang menentukan kapan sertifikat menjadi aktif.

Tipe: Timestamp

Diperlukan: Tidak

Certificate

Nama file untuk sertifikat.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 16384.

Pola: `[\u0009\u000A\u000D\u0020-\u00FF]*`

Diperlukan: Tidak

CertificateChain

Daftar sertifikat yang membentuk rantai untuk sertifikat.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 2097152.

Pola: `[\u0009\u000A\u000D\u0020-\u00FF]*`

Diperlukan: Tidak

CertificateId

Array pengidentifikasi untuk sertifikat yang diimpor. Anda menggunakan pengenal ini untuk bekerja dengan profil dan profil mitra.

Jenis: String

Kendala Panjang: Panjang tetap 22.

Pola: `cert-([0-9a-f]{17})`

Diperlukan: Tidak

Description

Nama atau deskripsi yang digunakan untuk mengidentifikasi sertifikat.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 200.

Pola: `[\p{Graph}]+`

Diperlukan: Tidak

InactiveDate

Tanggal opsional yang menentukan kapan sertifikat menjadi tidak aktif.

Tipe: Timestamp

Diperlukan: Tidak

NotAfterDate

Tanggal akhir sertifikat itu valid.

Tipe: Timestamp

Diperlukan: Tidak

NotBeforeDate

Tanggal paling awal bahwa sertifikat itu valid.

Tipe: Timestamp

Diperlukan: Tidak

Serial

Nomor seri untuk sertifikat.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 48.

Pola: `[\p{XDigit}{2}:?]*`

Diperlukan: Tidak

Status

Sertifikat dapat berupa`ACTIVE`,`PENDING_ROTATION`, atau`INACTIVE`. `PENDING_ROTATION`berarti bahwa sertifikat ini akan menggantikan sertifikat saat ini ketika kedaluwarsa.

Jenis: String

Nilai yang Valid: `ACTIVE | PENDING_ROTATION | INACTIVE`

Diperlukan: Tidak

Tags

Pasangan nilai kunci yang dapat digunakan untuk mengelompokkan dan mencari sertifikat.

Tipe: Array objek Tag

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50 item.

Diperlukan: Tidak

Type

Jika kunci pribadi telah ditentukan untuk sertifikat, jenisnya adalah`CERTIFICATE_WITH_PRIVATE_KEY`. Jika tidak ada kunci pribadi, tipenya adalah`CERTIFICATE`.

Jenis: String

Nilai yang Valid: `CERTIFICATE | CERTIFICATE_WITH_PRIVATE_KEY`

Diperlukan: Tidak

Usage

Menentukan bagaimana sertifikat ini digunakan. Ini dapat digunakan dengan cara-cara berikut:

- `SIGNING`: Untuk menandatangani pesan AS2
- `ENCRYPTION`: Untuk mengenkripsi pesan AS2
- `TLS`: Untuk mengamankan komunikasi AS2 yang dikirim melalui HTTPS

Jenis: String

Nilai yang Valid: `SIGNING | ENCRYPTION`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DescribedConnector

Menjelaskan parameter untuk konektor, seperti yang diidentifikasi olehConnectorId.

Daftar Isi

Arn

Nama Sumber Daya Amazon (ARN) yang unik untuk konektor.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 1600.

Pola: arn:\S+

Diperlukan: Ya

AccessRole

Konektor digunakan untuk mengirim file menggunakan protokol AS2 atau SFTP. Untuk peran akses, berikan Nama Sumber Daya Amazon (ARN) AWS Identity and Access Management peran yang akan digunakan.

Untuk konektor AS2

Dengan AS2, Anda dapat mengirim file dengan memanggil StartFileTransfer dan menentukan jalur file dalam parameter permintaan,. SendFilePaths Kami menggunakan direktori induk file (misalnya, untuk, direktori induk/bucket/dir/) untuk --send-file-paths /bucket/dir/file.txt sementara menyimpan file pesan AS2 yang diproses, menyimpan MDN ketika kami menerimanya dari mitra, dan menulis file JSON akhir yang berisi metadata transmisi yang relevan. Jadi, AccessRole kebutuhan untuk menyediakan akses baca dan tulis ke direktori induk dari lokasi file yang digunakan dalam StartFileTransfer permintaan. Selain itu, Anda perlu menyediakan akses baca dan tulis ke direktori induk dari file yang ingin Anda kirimStartFileTransfer.

Jika Anda menggunakan otentikasi Dasar untuk konektor AS2 Anda, peran akses memerlukan secretsmanager:GetSecretValue izin untuk rahasia tersebut. Jika rahasia dienkripsi menggunakan kunci yang dikelola pelanggan alih-alih kunci terkelola di AWS Secrets Manager, maka peran tersebut juga memerlukan kms:Decrypt izin untuk kunci tersebut.

Untuk konektor SFTP

Pastikan bahwa peran akses menyediakan akses baca dan tulis ke direktori induk dari lokasi file yang digunakan dalam `StartFileTransfer` permintaan. Selain itu, pastikan bahwa peran tersebut memberikan `secretsmanager:GetSecretValue` izin untuk AWS Secrets Manager.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 2048.

Pola: `arn:.*role/\S+`

Diperlukan: Tidak

As2Config

Struktur yang berisi parameter untuk objek konektor AS2.

Tipe: Objek As2ConnectorConfig

Diperlukan: Tidak

ConnectorId

Pengidentifikasi unik untuk konektor.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `c-([0-9a-f]{17})`

Diperlukan: Tidak

LoggingRole

Nama Sumber Daya Amazon (ARN) dari peran AWS Identity and Access Management (IAM) yang memungkinkan konektor mengaktifkan CloudWatch logging untuk peristiwa Amazon S3. Saat disetel, Anda dapat melihat aktivitas konektor di CloudWatch log Anda.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 2048.

Pola: `arn:.*role/\S+`

Diperlukan: Tidak

SecurityPolicyName

Nama teks kebijakan keamanan untuk konektor yang ditentukan.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 100.

Pola: `TransferSFTPConnectorSecurityPolicy-[A-Za-z0-9-]+`

Diperlukan: Tidak

ServiceManagedEgressIpAddresses

Daftar alamat IP jalan keluar dari konektor ini. Alamat IP ini ditetapkan secara otomatis saat Anda membuat konektor.

Tipe: Array string

Pola: `\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}`

Diperlukan: Tidak

SftpConfig

Struktur yang berisi parameter untuk objek konektor SFTP.

Tipe: Objek SftpConnectorConfig

Diperlukan: Tidak

Tags

Pasangan nilai kunci yang dapat digunakan untuk mengelompokkan dan mencari konektor.

Tipe: Array objek Tag

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50 item.

Diperlukan: Tidak

Url

URL titik akhir AS2 atau SFTP mitra.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum sebesar 255.

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DescribedExecution

Detail untuk objek eksekusi.

Daftar Isi

ExecutionId

Pengidentifikasi unik untuk eksekusi alur kerja.

Jenis: String

Batas Panjang: Panjang tetap 36.

Pola: `[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}`

Diperlukan: Tidak

ExecutionRole

Peran IAM terkait dengan eksekusi.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 2048.

Pola: `arn:.*role/\S+`

Diperlukan: Tidak

InitialFileLocation

Struktur yang menjelaskan lokasi file Amazon S3 atau EFS. Ini adalah lokasi file ketika eksekusi dimulai: jika file sedang disalin, ini adalah lokasi file awal (sebagai lawan dari tujuan).

Tipe: Objek FileLocation

Diperlukan: Tidak

LoggingConfiguration

Peran logging IAM yang terkait dengan eksekusi.

Tipe: Objek LoggingConfiguration

Diperlukan: Tidak

PosixProfile

Identitas POSIX lengkap, termasuk ID pengguna (`Uid`), ID grup (`Gid`), dan setiap grup sekunder ID (`SecondaryGids`), yang mengendalikan akses pengguna Anda ke sistem file Amazon EFS Anda. POSIX izin yang ditetapkan pada file dan direktori dalam sistem file Anda menentukan tingkat akses yang pengguna Anda dapatkan ketika mentransfer file ke dalam dan keluar dari sistem file Amazon EFS Anda.

Tipe: Objek PosixProfile

Diperlukan: Tidak

Results

Struktur yang menggambarkan hasil eksekusi. Ini termasuk daftar langkah-langkah bersama dengan rincian setiap langkah, jenis kesalahan dan pesan (jika ada), dan `OnExceptionSteps` struktur.

Tipe: Objek ExecutionResults

Diperlukan: Tidak

ServiceMetadata

Objek kontainer untuk detail sesi yang terkait dengan alur kerja.

Tipe: Objek ServiceMetadata

Diperlukan: Tidak

Status

Status adalah salah satu eksekusi. Dapat dalam proses, selesai, pengecualian ditemui, atau menangani pengecualian.

Jenis: String

Nilai yang Valid: `IN_PROGRESS | COMPLETED | EXCEPTION | HANDLING_EXCEPTION`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

DescribedHostKey

Detail untuk kunci host server.

Daftar Isi

Arn

Nama Sumber Daya Amazon (ARN) unik untuk kunci host.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 1600.

Pola: `arn:\S+`

Diperlukan: Ya

DateImported

Tanggal di mana kunci host ditambahkan ke server.

Tipe: Timestamp

Diperlukan: Tidak

Description

Deskripsi teks untuk kunci host ini.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 200.

Pola: `[\p{Print}]*`

Diperlukan: Tidak

HostKeyFingerprint

Sidik jari kunci publik, yang merupakan urutan pendek byte yang digunakan untuk mengidentifikasi kunci publik yang lebih panjang.

Tipe: String

Wajib: Tidak

HostKeyId

Pengenal unik untuk kunci host.

Jenis: String

Kendala Panjang: Panjang tetap 25.

Pola: `hostkey-[0-9a-f]{17}`

Diperlukan: Tidak

Tags

Pasangan kunci-nilai yang dapat digunakan untuk mengelompokkan dan mencari kunci host.

Tipe: Array objek Tag

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50 item.

Diperlukan: Tidak

Type

Algoritma enkripsi yang digunakan untuk kunci host. `Type`Parameter ditentukan dengan menggunakan salah satu nilai berikut:

- `ssh-rsa`
- `ssh-ed25519`
- `ecdsa-sha2-nistp256`
- `ecdsa-sha2-nistp384`
- `ecdsa-sha2-nistp521`

Tipe: String

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

DescribedProfile

Detail untuk profil AS2 lokal atau mitra.

Daftar Isi

Arn

Nama Sumber Daya Amazon (ARN) unik untuk profil.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 1600.

Pola: `arn:\S+`

Diperlukan: Ya

As2Id

`As2Id`Itu adalah nama AS2, seperti yang didefinisikan dalam RFC 4130. Untuk transfer masuk, ini adalah `AS2-From` header untuk pesan AS2 yang dikirim dari mitra. Untuk konektor keluar, ini adalah `AS2-To` header untuk pesan AS2 yang dikirim ke mitra menggunakan operasi `StartFileTransfer` API. ID ini tidak dapat menyertakan spasi.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: `[\p{Print}\s]*`

Diperlukan: Tidak

CertificateIds

Array pengidentifikasi untuk sertifikat yang diimpor. Anda menggunakan pengenal ini untuk bekerja dengan profil dan profil mitra.

Tipe: Array string

Kendala Panjang: Panjang tetap 22.

Pola: `cert-([0-9a-f]{17})`

Diperlukan: Tidak

ProfileId

Pengenal unik untuk profil AS2 lokal atau mitra.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `p-([0-9a-f]{17})`

Diperlukan: Tidak

ProfileType

Menunjukkan apakah hanya akan mencantumkan profil `LOCAL` tipe atau hanya `PARTNER` mengetik profil. Jika tidak disediakan dalam permintaan, perintah mencantumkan semua jenis profil.

Jenis: String

Nilai yang Valid: `LOCAL | PARTNER`

Diperlukan: Tidak

Tags

Pasangan kunci-nilai yang dapat digunakan untuk mengelompokkan dan mencari profil.

Tipe: Array objek Tag

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50 item.

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.

- AWSSDK for Java V2
- AWS SDK for Ruby V3

DescribedSecurityPolicy

Menjelaskan properti kebijakan keamanan yang Anda tentukan. Untuk informasi selengkapnya tentang kebijakan keamanan, lihat Bekerja dengan kebijakan keamanan untuk server atau Bekerja dengan kebijakan keamanan untuk konektor SFTP.

Daftar Isi

SecurityPolicyName

Nama teks dari kebijakan keamanan yang ditentukan.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 100.

Pola: `Transfer[A-Za-z0-9]*SecurityPolicy-[A-Za-z0-9-]+`

Diperlukan: Ya

Fips

Menentukan apakah kebijakan ini memungkinkan Federal Information Processing Standards (FIPS). Parameter ini berlaku untuk kebijakan keamanan server dan konektor.

Tipe: Boolean

Wajib: Tidak

Protocols

Daftar protokol transfer file yang berlaku untuk kebijakan keamanan.

Tipe: Array string

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 5 item.

Nilai yang Valid: `SFTP | FTPS`

Diperlukan: Tidak

SshCiphers

Daftar algoritma enkripsi sandi Secure Shell (SSH) yang diaktifkan dalam kebijakan keamanan yang dilampirkan ke server atau konektor. Parameter ini berlaku untuk kebijakan keamanan server dan konektor.

Tipe: Array string

Batasan Panjang: Panjang minimum sebesar 0. Panjang maksimum 50.

Diperlukan: Tidak

SshHostKeyAlgorithms

Daftar algoritma kunci host untuk kebijakan keamanan.

> **Note**
>
> Parameter ini hanya berlaku untuk kebijakan keamanan untuk konektor.

Tipe: Array string

Batasan Panjang: Panjang minimum sebesar 0. Panjang maksimum 50.

Diperlukan: Tidak

SshKexs

Daftar algoritma enkripsi pertukaran kunci SSH (KEX) yang diaktifkan dalam kebijakan keamanan yang dilampirkan ke server atau konektor. Parameter ini berlaku untuk kebijakan keamanan server dan konektor.

Tipe: Array string

Batasan Panjang: Panjang minimum sebesar 0. Panjang maksimum 50.

Diperlukan: Tidak

SshMacs

Daftar algoritma enkripsi kode otentikasi pesan SSH (MAC) yang diaktifkan dalam kebijakan keamanan yang dilampirkan ke server atau konektor. Parameter ini berlaku untuk kebijakan keamanan server dan konektor.

Tipe: Array string

Batasan Panjang: Panjang minimum sebesar 0. Panjang maksimum 50.

Diperlukan: Tidak

TlsCiphers

Daftar algoritma enkripsi sandi Transport Layer Security (TLS) yang diaktifkan dalam kebijakan keamanan yang dilampirkan ke server.

> **Note**
>
> Parameter ini hanya berlaku untuk kebijakan keamanan untuk server.

Tipe: Array string

Batasan Panjang: Panjang minimum sebesar 0. Panjang maksimum 50.

Diperlukan: Tidak

Type

Jenis sumber daya yang diterapkan kebijakan keamanan, baik server atau konektor.

Jenis: String

Nilai yang Valid: `SERVER | CONNECTOR`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DescribedServer

Menjelaskan properti server berkemampuan protokol transfer file yang ditentukan.

Daftar Isi

Arn

Menentukan Nama Sumber Daya Amazon (ARN) unik dari server.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 1600.

Pola: `arn:\S+`

Diperlukan: Ya

As2ServiceManagedEgressIpAddresses

Daftar alamat IP jalan keluar dari server ini. Alamat IP ini hanya relevan untuk server yang menggunakan protokol AS2. Mereka digunakan untuk mengirim mDNS asinkron.

Alamat IP ini ditetapkan secara otomatis saat Anda membuat server AS2. Selain itu, jika Anda memperbarui server yang ada dan menambahkan protokol AS2, alamat IP statis juga ditetapkan.

Tipe: Array string

Pola: `\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}`

Diperlukan: Tidak

Certificate

Menentukan ARN dari sertifikat Certificate AWS Manager (ACM). Diperlukan saat `Protocols` diatur ke `FTPS`.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 1600.

Diperlukan: Tidak

Domain

Menentukan domain sistem penyimpanan yang digunakan untuk transfer file. Ada dua domain yang tersedia: Amazon Simple Storage Service (Amazon S3) dan Amazon Elastic File System (Amazon EFS). Nilai defaultnya adalah S3.

Jenis: String

Nilai yang Valid: `S3 | EFS`

Diperlukan: Tidak

EndpointDetails

Pengaturan titik akhir virtual private cloud (VPC) yang dikonfigurasi untuk server Anda. Ketika Anda meng-host titik akhir Anda dalam VPC Anda, Anda dapat membuat titik akhir Anda hanya dapat diakses oleh sumber daya dalam VPC Anda, atau Anda dapat melampirkan alamat IP Elastis dan membuat titik akhir Anda dapat diakses oleh klien melalui internet. Grup keamanan default VPC Anda secara otomatis ditetapkan ke titik akhir Anda.

Tipe: Objek EndpointDetails

Diperlukan: Tidak

EndpointType

Mendefinisikan jenis titik akhir yang terhubung dengan server Anda. Jika server Anda terhubung ke titik akhir VPC, server Anda tidak dapat diakses melalui internet publik.

Jenis: String

Nilai yang Valid: `PUBLIC | VPC | VPC_ENDPOINT`

Diperlukan: Tidak

HostKeyFingerprint

Menentukan sidik jari SHA256 yang dikodekan Base64 dari kunci host server. Nilai ini setara dengan output dari `ssh-keygen -l -f my-new-server-key` perintah.

Tipe: String

Wajib: Tidak

IdentityProviderDetails

Menentukan informasi untuk memanggil API otentikasi yang disediakan pelanggan. Bidang ini tidak diisi ketika server `AWS_DIRECTORY_SERVICE` atau`SERVICE_MANAGED`. `IdentityProviderType`

Tipe: Objek IdentityProviderDetails

Diperlukan: Tidak

IdentityProviderType

Modus otentikasi untuk server. Nilai defaultnya adalah`SERVICE_MANAGED`, yang memungkinkan Anda untuk menyimpan dan mengakses kredensyal pengguna dalam layanan. AWS Transfer Family

Gunakan `AWS_DIRECTORY_SERVICE` untuk menyediakan akses ke grup Direktori Aktif di AWS Directory Service for Microsoft Active Directory atau Microsoft Active Directory di lingkungan lokal Anda atau AWS menggunakan AD Connector. Opsi ini juga mengharuskan Anda untuk memberikan ID Direktori dengan menggunakan `IdentityProviderDetails` parameter.

Gunakan nilai `API_GATEWAY` untuk mengintegrasikan dengan penyedia identitas pilihan Anda. `API_GATEWAY`Pengaturan mengharuskan Anda untuk menyediakan URL titik akhir Amazon API Gateway untuk memanggil otentikasi dengan menggunakan parameter. `IdentityProviderDetails`

Gunakan `AWS_LAMBDA` nilai untuk langsung menggunakan AWS Lambda fungsi sebagai penyedia identitas Anda. Jika Anda memilih nilai ini, Anda harus menentukan ARN untuk fungsi Lambda dalam `Function` parameter untuk tipe data. `IdentityProviderDetails`

Jenis: String

Nilai yang Valid: `SERVICE_MANAGED | API_GATEWAY | AWS_DIRECTORY_SERVICE | AWS_LAMBDA`

Diperlukan: Tidak

LoggingRole

Nama Sumber Daya Amazon (ARN) dari peran AWS Identity and Access Management (IAM) yang memungkinkan server mengaktifkan CloudWatch pencatatan Amazon untuk Amazon S3 atau Amazon EFSevents. Saat disetel, Anda dapat melihat aktivitas pengguna di CloudWatch log Anda.

Jenis: String

Batasan Panjang: Panjang minimum sebesar 0. Panjang maksimum 2048.

Pola: `(|arn:.*role/\S+)`

Diperlukan: Tidak

PostAuthenticationLoginBanner

Menentukan string untuk ditampilkan ketika pengguna terhubung ke server. String ini ditampilkan setelah pengguna mengautentikasi.

> **Note**
>
> Protokol SFTP tidak mendukung spanduk tampilan pasca-otentikasi.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 4096.

Pola: `[\x09-\x0D\x20-\x7E]*`

Diperlukan: Tidak

PreAuthenticationLoginBanner

Menentukan string untuk ditampilkan ketika pengguna terhubung ke server. String ini ditampilkan sebelum pengguna mengautentikasi. Misalnya, spanduk berikut menampilkan detail tentang penggunaan sistem:

```
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.
```

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 4096.

Pola: `[\x09-\x0D\x20-\x7E]*`

Diperlukan: Tidak

ProtocolDetails

Pengaturan protokol yang dikonfigurasi untuk server Anda.

- Untuk menunjukkan mode pasif (untuk protokol FTP dan FTPS), gunakan parameter. `PassiveIp` Masukkan satu alamat IPv4 bertitik quad, seperti alamat IP eksternal dari firewall, router, atau penyeimbang beban.
- Untuk mengabaikan kesalahan yang dihasilkan saat klien mencoba menggunakan `SETSTAT` perintah pada file yang Anda unggah ke bucket Amazon S3, gunakan `SetStatOption` parameternya. Agar AWS Transfer Family server mengabaikan `SETSTAT` perintah dan mengunggah file tanpa perlu membuat perubahan apa pun pada klien SFTP Anda, tetapkan nilainya. `ENABLE_NO_OP` Jika Anda menyetel `SetStatOption` parameternya`ENABLE_NO_OP`, Transfer Family akan menghasilkan entri CloudWatch log ke Amazon Logs, sehingga Anda dapat menentukan kapan klien melakukan `SETSTAT` panggilan.
- Untuk menentukan apakah AWS Transfer Family server Anda melanjutkan sesi terbaru yang dinegosiasikan melalui ID sesi unik, gunakan parameternya. `TlsSessionResumptionMode`
- `As2Transports`menunjukkan metode transport untuk pesan AS2. Saat ini, hanya HTTP yang didukung.

Tipe: Objek ProtocolDetails

Diperlukan: Tidak

Protocols

Menentukan protokol transfer file atau protokol di mana klien protokol transfer file Anda dapat terhubung ke titik akhir server Anda. Protokol yang tersedia adalah:

- `SFTP` (Secure Shell (SSH) Protokol Transfer File): Transfer file melalui SSH
- `FTPS` (File Transfer Protocol Secure): Transfer file dengan enkripsi TLS
- `FTP` (Protokol Transfer File): Transfer file tidak terenkripsi
- `AS2`(Pernyataan Penerapan 2): digunakan untuk mengangkut data terstruktur business-to-business

> **Note**
>
> - Jika Anda memilih`FTPS`, Anda harus memilih sertifikat yang disimpan di AWS Certificate Manager (ACM) yang digunakan untuk mengidentifikasi server Anda ketika klien terhubung ke sana melalui FTPS.

- Jika `Protocol` termasuk salah satu `FTP` atau`FTPS`, maka `EndpointType` harus `VPC` dan `IdentityProviderType` harus baik`AWS_DIRECTORY_SERVICE`,`AWS_LAMBDA`, atau`API_GATEWAY`.
- Jika `Protocol` termasuk `FTP`, maka `AddressAllocationIds` tidak dapat dikaitkan.
- Jika `Protocol` disetel hanya ke`SFTP`, `EndpointType` dapat diatur ke `PUBLIC` dan `IdentityProviderType` dapat disetel salah satu jenis identitas yang didukung:`SERVICE_MANAGED`,`AWS_DIRECTORY_SERVICE`,`AWS_LAMBDA`, atau`API_GATEWAY`.
- Jika `Protocol` termasuk`AS2`, maka `EndpointType` harus`VPC`, dan domain harus Amazon S3.

Tipe: Array string

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 4 item.

Nilai yang Valid: `SFTP | FTP | FTPS | AS2`

Diperlukan: Tidak

S3StorageOptions

Menentukan apakah atau tidak kinerja untuk direktori Amazon S3 Anda dioptimalkan. Ini dinonaktifkan secara default.

Secara default, pemetaan direktori home memiliki `TYPE` file. `DIRECTORY` Jika Anda mengaktifkan opsi ini, Anda kemudian perlu secara eksplisit menyetel `HomeDirectoryMapEntry` `Type` ke `FILE` jika Anda ingin pemetaan memiliki target file.

Tipe: Objek S3StorageOptions

Diperlukan: Tidak

SecurityPolicyName

Menentukan nama kebijakan keamanan untuk server.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 100.

Pola: `Transfer[A-Za-z0-9]*SecurityPolicy-[A-Za-z0-9-]+`

Diperlukan: Tidak

ServerId

Menentukan pengidentifikasi unik yang ditetapkan sistem untuk server yang Anda buat instance.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Tidak

State

Kondisi server yang dijelaskan. Nilai `ONLINE` menunjukkan bahwa server dapat menerima pekerjaan dan mentransfer file. `State`Nilai `OFFLINE` berarti bahwa server tidak dapat melakukan operasi transfer file.

Status `STARTING` dan `STOPPING` menunjukkan bahwa server berada dalam keadaan perantara, baik tidak sepenuhnya dapat merespons, atau tidak sepenuhnya offline. Nilai `START_FAILED` atau `STOP_FAILED` dapat menunjukkan kondisi kesalahan.

Jenis: String

Nilai yang Valid: `OFFLINE | ONLINE | STARTING | STOPPING | START_FAILED | STOP_FAILED`

Diperlukan: Tidak

StructuredLogDestinations

Menentukan grup log yang log server Anda dikirim.

Untuk menentukan grup log, Anda harus memberikan ARN untuk grup log yang ada. Dalam hal ini, format grup log adalah sebagai berikut:

`arn:aws:logs:region-name:amazon-account-id:log-group:log-group-name:*`

Misalnya, `arn:aws:logs:us-east-1:111122223333:log-group:mytestgroup:*`

Jika sebelumnya Anda telah menentukan grup log untuk server, Anda dapat menghapusnya, dan pada dasarnya mematikan logging terstruktur, dengan memberikan nilai kosong untuk parameter ini dalam `update-server` panggilan. Sebagai contoh:

`update-server --server-id s-1234567890abcdef0 --structured-log-destinations`

Tipe: Array string.

Anggota Array: Jumlah minimum 0 item. Jumlah maksimum 1 item.

Batasan Panjang: Panjang minimum 20. Panjang maksimum 1600.

Pola: `arn:\S+`

Diperlukan: Tidak

Tags

Menentukan pasangan kunci-nilai yang dapat Anda gunakan untuk mencari dan mengelompokkan server yang ditugaskan ke server yang dijelaskan.

Tipe: Array objek Tag

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50 item.

Diperlukan: Tidak

UserCount

Menentukan jumlah pengguna yang ditugaskan ke server yang Anda tentukan dengan. `ServerId`

Tipe: Integer

Wajib: Tidak

WorkflowDetails

Menentukan ID alur kerja untuk alur kerja yang akan ditetapkan dan peran eksekusi yang digunakan untuk mengeksekusi alur kerja.

Selain alur kerja untuk mengeksekusi ketika file diunggah sepenuhnya, juga `WorkflowDetails` dapat berisi ID alur kerja (dan peran eksekusi) untuk alur kerja untuk mengeksekusi pada upload sebagian. Upload sebagian terjadi ketika sesi server terputus saat file masih diunggah.

Tipe: Objek WorkflowDetails

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

DescribedUser

Menjelaskan properti pengguna yang telah ditentukan.

Daftar Isi

Arn

Menentukan Nama Sumber Daya Amazon (ARN) unik untuk pengguna yang diminta untuk dijelaskan.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 1600.

Pola: `arn:\S+`

Diperlukan: Ya

HomeDirectory

Direktori arahan (folder) untuk pengguna ketika mereka masuk ke server menggunakan klien.

Contoh `HomeDirectory` adalah `/bucket_name/home/mydirectory`.

> **Note**
>
> Parameter `HomeDirectory` hanya digunakan jika `HomeDirectoryType` diatur ke `PATH`.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 1024.

Pola: `(|/.*)`

Diperlukan: Tidak

HomeDirectoryMappings

Pemetaan direktori logis yang menentukan jalur dan kunci Amazon S3 atau Amazon EFS apa yang harus terlihat oleh pengguna Anda dan bagaimana Anda ingin membuatnya terlihat. Anda harus menentukan `Entry` dan `Target` memasangkan, di mana `Entry` menunjukkan bagaimana

jalur dibuat terlihat dan `Target` merupakan jalur Amazon S3 atau Amazon EFS yang sebenarnya. Jika Anda hanya menentukan target, itu ditampilkan apa adanya. Anda juga harus memastikan bahwa peran AWS Identity and Access Management (IAM) Anda menyediakan akses ke jalur masuk`Target`. Nilai ini dapat diatur hanya ketika `HomeDirectoryType` diatur ke LOGICAL.

Dalam kebanyakan kasus, Anda dapat menggunakan nilai ini alih-alih kebijakan sesi untuk mengunci pengguna Anda ke direktori home yang ditunjuk (`chroot`“”). Untuk melakukan ini, Anda dapat mengatur `Entry` ke '/' dan mengatur `Target` ke nilai HomeDirectory parameter.

Tipe: Array objek HomeDirectoryMapEntry

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50000 item.

Diperlukan: Tidak

HomeDirectoryType

Jenis direktori pendaratan (folder) yang Anda inginkan direktori home pengguna Anda ketika mereka masuk ke server. Jika Anda mengaturnya`PATH`, pengguna akan melihat bucket Amazon S3 absolut atau jalur Amazon EFS seperti pada klien protokol transfer file mereka. Jika Anda menyetelnya`LOGICAL`, Anda harus menyediakan pemetaan `HomeDirectoryMappings` untuk bagaimana Anda ingin membuat jalur Amazon S3 atau Amazon EFS terlihat oleh pengguna Anda.

> **Note**
>
> Jika `HomeDirectoryType` ya`LOGICAL`, Anda harus memberikan pemetaan, menggunakan parameter. `HomeDirectoryMappings` Jika, di sisi lain, `HomeDirectoryType` adalah`PATH`, Anda memberikan jalur absolut menggunakan `HomeDirectory` parameter. Anda tidak dapat memiliki keduanya `HomeDirectory` dan `HomeDirectoryMappings` di template Anda.

Jenis: String

Nilai yang Valid: `PATH | LOGICAL`

Diperlukan: Tidak

Policy

Kebijakan sesi untuk pengguna Anda sehingga Anda dapat menggunakan peran yang sama AWS Identity and Access Management (IAM) di beberapa pengguna. Kebijakan ini mencakup

akses pengguna ke sebagian bucket Amazon S3 mereka. Variabel yang dapat Anda gunakan dalam kebijakan ini meliputi `${Transfer:UserName}`, `${Transfer:HomeDirectory}`, dan `${Transfer:HomeBucket}`.

Jenis: String

Batasan Panjang: Panjang minimum sebesar 0. Panjang maksimum 2048.

Diperlukan: Tidak

PosixProfile

Menentukan identitas POSIX penuh, termasuk ID pengguna (`Uid`), ID grup (`Gid`), dan setiap ID grup sekunder (`SecondaryGids`), yang mengontrol akses pengguna Anda ke sistem file Amazon Elastic File System (Amazon EFS) Anda. POSIX izin yang ditetapkan pada file dan direktori dalam sistem file Anda menentukan tingkat akses pengguna Anda mendapatkan ketika mentransfer file ke dalam dan keluar dari sistem file Amazon EFS Anda.

Tipe: Objek PosixProfile

Diperlukan: Tidak

Role

Nama Sumber Daya Amazon (ARN) dari peran AWS Identity and Access Management (IAM) yang mengontrol akses pengguna ke bucket Amazon S3 atau sistem file Amazon EFS. Kebijakan yang dilampirkan pada peran ini menentukan tingkat akses yang ingin Anda berikan kepada pengguna saat mentransfer file masuk dan keluar dari bucket Amazon S3 atau sistem file Amazon EFS Anda. IAM role juga harus berisi hubungan kepercayaan yang mengizinkan server untuk mengakses sumber daya Anda saat melayani permintaan transfer pengguna.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 2048.

Pola: `arn:.*role/\S+`

Diperlukan: Tidak

SshPublicKeys

Menentukan bagian kunci publik dari kunci Secure Shell (SSH) yang disimpan untuk pengguna yang dijelaskan.

Tipe: Array objek SshPublicKey

Anggota Array: Jumlah minimum 0 item. Jumlah maksimum 5 item.

Diperlukan: Tidak

Tags

Menentukan pasangan kunci-nilai untuk pengguna yang diminta. Tag dapat digunakan untuk mencari dan mengelompokkan pengguna untuk berbagai tujuan.

Tipe: Array objek Tag

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50 item.

Diperlukan: Tidak

UserName

Menentukan nama pengguna yang diminta untuk dijelaskan. Nama pengguna digunakan untuk tujuan otentikasi. Ini adalah string yang akan digunakan oleh pengguna Anda ketika mereka masuk ke server Anda.

Jenis: String

Batasan Panjang: Panjang minimum 3. Panjang maksimum 100.

Pola: `[\w][\w@.-]{2,99}`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

DescribedWorkflow

Menjelaskan properti alur kerja yang ditentukan

Daftar Isi

Arn

Menentukan Nama Sumber Daya Amazon (ARN) unik untuk alur kerja.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 1600.

Pola: `arn:\S+`

Diperlukan: Ya

Description

Menentukan deskripsi teks untuk alur kerja.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 256.

Pola: `[\w- ]*`

Diperlukan: Tidak

OnExceptionSteps

Menentukan langkah-langkah (tindakan) untuk mengambil jika kesalahan ditemui selama pelaksanaan alur kerja.

Tipe: Array objek WorkflowStep

Anggota Array: Jumlah minimum 0 item. Jumlah maksimum 8 item.

Diperlukan: Tidak

Steps

Menentukan rincian untuk langkah-langkah yang ada dalam alur kerja yang ditentukan.

Tipe: Array objek WorkflowStep

Anggota Array: Jumlah minimum 0 item. Jumlah maksimum 8 item.

Diperlukan: Tidak

Tags

Pasangan nilai kunci yang dapat digunakan untuk mengelompokkan dan mencari alur kerja. Tag adalah metadata yang dilampirkan ke alur kerja untuk tujuan apa pun.

Tipe: Array objek Tag

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50 item.

Diperlukan: Tidak

WorkflowId

Pengidentifikasi unik untuk alur kerja.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `w-([a-z0-9]{17})`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

EfsFileLocation

Menentukan rincian untuk lokasi file untuk file yang sedang digunakan dalam alur kerja. Hanya berlaku jika Anda menggunakan Amazon Elastic File Systems (Amazon EFS) untuk penyimpanan.

Daftar Isi

FileSystemId

Pengidentifikasi sistem file, yang ditugaskan oleh Amazon EFS.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 128.

Pola: `(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})`

Diperlukan: Tidak

Path

Pathname untuk folder yang digunakan oleh alur kerja.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 65536.

Pola: `[^\x00]+`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

EndpointDetails

Pengaturan titik akhir virtual private cloud (VPC) yang dikonfigurasi untuk server berkemampuan protokol transfer file Anda. Dengan VPC endpoint, Anda dapat membatasi akses ke server dan sumber daya hanya di dalam VPC Anda. Untuk mengontrol lalu lintas internet yang masuk, panggil `UpdateServer` API dan lampirkan alamat IP Elastis ke titik akhir server Anda.

> **Note**
>
> Setelah 19 Mei 2021, Anda tidak akan dapat membuat server menggunakan `EndpointType=VPC_ENDPOINT` di AWS akun Anda jika akun Anda belum melakukannya sebelum 19 Mei 2021. Jika Anda telah membuat server dengan `EndpointType=VPC_ENDPOINT` di AWS akun Anda pada atau sebelum 19 Mei 2021, Anda tidak akan terpengaruh. Setelah tanggal ini, gunakan `EndpointType` =VPC.
> Untuk informasi selengkapnya, lihat Menghentikan penggunaan VPC_ENDPOINT.

Daftar Isi

AddressAllocationIds

Daftar ID alokasi alamat yang diperlukan untuk melampirkan alamat IP Elastis ke titik akhir server Anda.

ID alokasi alamat sesuai dengan ID alokasi alamat IP Elastis. Nilai ini dapat diambil dari `allocationId` bidang dari tipe data Alamat Amazon EC2. Salah satu cara untuk mengambil nilai ini adalah dengan memanggil EC2 API DescribeAddresses.

Parameter ini bersifat opsional. Tetapkan parameter ini jika Anda ingin membuat titik akhir VPC Anda menghadap publik. Untuk detailnya, lihat Membuat titik akhir yang menghadap internet untuk server Anda.

> **Note**
>
> Properti ini hanya dapat diatur sebagai berikut:
>
> - `EndpointType`harus diatur ke `VPC`
> - Server Transfer Family harus offline.
> - Anda tidak dapat mengatur parameter ini untuk server Transfer Family yang menggunakan protokol FTP.

- Server harus sudah `SubnetIds` terisi (`SubnetIds`dan `AddressAllocationIds` tidak dapat diperbarui secara bersamaan).
- `AddressAllocationIds`tidak dapat berisi duplikat, dan harus sama panjangnya dengan`SubnetIds`. Misalnya, jika Anda memiliki tiga ID subnet, Anda juga harus menentukan tiga ID alokasi alamat.
- Panggil `UpdateServer` API untuk mengatur atau mengubah parameter ini.

Tipe: Array string

Diperlukan: Tidak

SecurityGroupIds

Daftar ID grup keamanan yang tersedia untuk dilampirkan ke titik akhir server Anda.

> **Note**
>
> Properti ini hanya dapat diatur saat `EndpointType` diatur ke `VPC`.
> Anda dapat mengedit `SecurityGroupIds` properti di UpdateServerAPI hanya jika Anda mengubah `EndpointType` dari `PUBLIC` atau `VPC_ENDPOINT` ke`VPC`. Untuk mengubah grup keamanan yang terkait dengan titik akhir VPC server Anda setelah pembuatan, gunakan Amazon EC2 API. ModifyVpcEndpoint

Tipe: Array string

Kendala Panjang: Panjang minimum 11. Panjang maksimum 20.

Pola: `sg-[0-9a-f]{8,17}`

Diperlukan: Tidak

SubnetIds

Daftar ID subnet yang diperlukan untuk meng-host titik akhir server Anda di VPC Anda.

> **Note**
>
> Properti ini hanya dapat diatur saat `EndpointType` diatur ke `VPC`.

Tipe: Array string

Diperlukan: Tidak

VpcEndpointId

Pengidentifikasi titik akhir VPC.

> **Note**
>
> Properti ini hanya dapat diatur saat `EndpointType` diatur ke `VPC_ENDPOINT`. Untuk informasi selengkapnya, lihat Menghentikan penggunaan VPC_ENDPOINT.

Tipe: String

Kendala Panjang: Panjang tetap 22.

Pola: `vpce-[0-9a-f]{17}`

Diperlukan: Tidak

VpcId

Pengidentifikasi VPC dari VPC di mana titik akhir server akan di-host.

> **Note**
>
> Properti ini hanya dapat diatur saat `EndpointType` diatur ke `VPC`.

Tipe: String

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.

- AWSSDK for Java V2
- AWS SDK for Ruby V3

ExecutionError

Menentukan pesan kesalahan dan jenis, untuk kesalahan yang terjadi selama pelaksanaan alur kerja.

Daftar Isi

Message

Menentukan pesan deskriptif yang sesuai dengan. `ErrorType`

Tipe: String

Wajib: Ya

Type

Menentukan jenis kesalahan.

- `ALREADY_EXISTS`: terjadi untuk langkah salin, jika opsi timpa tidak dipilih dan file dengan nama yang sama sudah ada di lokasi target.
- `BAD_REQUEST`: permintaan buruk umum: misalnya, langkah yang mencoba menandai file EFS kembali`BAD_REQUEST`, karena hanya file S3 yang dapat ditandai.
- `CUSTOM_STEP_FAILED`: terjadi ketika langkah kustom memberikan callback yang menunjukkan kegagalan.
- `INTERNAL_SERVER_ERROR`: kesalahan catch-all yang dapat terjadi karena berbagai alasan.
- `NOT_FOUND`: terjadi ketika entitas yang diminta, misalnya file sumber untuk langkah penyalinan, tidak ada.
- `PERMISSION_DENIED`: terjadi jika kebijakan Anda tidak berisi izin yang benar untuk menyelesaikan satu atau beberapa langkah dalam alur kerja.
- `TIMEOUT`: terjadi ketika waktu eksekusi habis.

  > **Note**
  >
  > Anda dapat mengatur `TimeoutSeconds` untuk langkah khusus, di mana saja dari 1 detik hingga 1800 detik (30 menit).

- `THROTTLED`: terjadi jika Anda melebihi tingkat isi ulang eksekusi baru dari satu alur kerja per detik.

Jenis: String

Nilai yang Valid: `PERMISSION_DENIED | CUSTOM_STEP_FAILED | THROTTLED | ALREADY_EXISTS | NOT_FOUND | BAD_REQUEST | TIMEOUT | INTERNAL_SERVER_ERROR`

Wajib: Ya

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWSSDK for C++
- AWSSDK for Go
- AWSSDK for Java V2
- AWSSDK for Ruby V3

ExecutionResults

Menentukan langkah-langkah dalam alur kerja, serta langkah-langkah untuk mengeksekusi jika terjadi kesalahan selama eksekusi alur kerja.

Daftar Isi

OnExceptionSteps

Menentukan langkah-langkah (tindakan) untuk mengambil jika kesalahan ditemui selama pelaksanaan alur kerja.

Tipe: Array objek ExecutionStepResult

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50 item.

Wajib: Tidak

Steps

Menentukan rincian untuk langkah-langkah yang ada dalam alur kerja yang ditentukan.

Tipe: Array objek ExecutionStepResult

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 50 item.

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWSSDK for C++
- AWSSDK for Go
- AWSSDK for Java V2
- AWSSDK for Ruby V3

ExecutionStepResult

Menentukan rincian berikut untuk langkah: kesalahan (jika ada), output (jika ada), dan jenis langkah.

Daftar Isi

Error

Menentukan rincian untuk kesalahan, jika itu terjadi selama pelaksanaan langkah alur kerja yang ditentukan.

Tipe: Objek ExecutionError

Diperlukan: Tidak

Outputs

Nilai untuk pasangan kunci/nilai diterapkan sebagai tag ke file. Hanya berlaku jika tipe langkahnya`TAG`.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 65536.

Diperlukan: Tidak

StepType

Salah satu jenis langkah yang tersedia.

- **`COPY`**- Salin file ke lokasi lain.
- **`CUSTOM`**- Lakukan langkah kustom dengan target AWS Lambda fungsi.
- **`DECRYPT`**- Dekripsi file yang dienkripsi sebelum diunggah.
- **`DELETE`**- Hapus file.
- **`TAG`**- Tambahkan tag ke file.

Jenis: String

Nilai yang Valid: `COPY | CUSTOM | TAG | DELETE | DECRYPT`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

FileLocation

Menentukan detail file Amazon S3 atau EFS yang akan digunakan dalam langkah.

Daftar Isi

EfsFileLocation

Menentukan pengenal Amazon EFS dan jalur untuk file yang digunakan.

Tipe: Objek EfsFileLocation

Wajib: Tidak

S3FileLocation

Menentukan rincian S3 untuk file yang digunakan, seperti bucket, ETag, dan sebagainya.

Tipe: Objek S3FileLocation

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWSSDK for C++
- AWSSDK for Go
- AWSSDK for Java V2
- AWSSDK for Ruby V3

HomeDirectoryMapEntry

Mewakili sebuah objek yang berisi entri dan target untuk `HomeDirectoryMappings`.

Berikut ini adalah contoh `Entry` dan `Target` pair untuk`chroot`.

```
[ { "Entry": "/", "Target": "/bucket_name/home/mydirectory" } ]
```

Daftar Isi

Entry

Mewakili entri untuk `HomeDirectoryMappings`.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 1024.

Pola: `/.*`

Diperlukan: Ya

Target

Mewakili target peta yang digunakan dalam `HomeDirectoryMapEntry`.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 1024.

Pola: `/.*`

Diperlukan: Ya

Type

Menentukan jenis pemetaan. Atur tipe ke `FILE` jika Anda ingin pemetaan menunjuk ke file, atau `DIRECTORY` direktori mengarah ke direktori.

> **Note**
>
> Secara default, pemetaan direktori home memiliki `Type` of `DIRECTORY` saat Anda membuat server Transfer Family. Anda perlu mengatur `Type` secara eksplisit `FILE` jika Anda ingin pemetaan memiliki target file.

Jenis: String

Nilai yang Valid: `FILE | DIRECTORY`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

IdentityProviderDetails

Mengembalikan informasi yang terkait dengan jenis otentikasi pengguna yang digunakan untuk pengguna server berkemampuan protokol transfer file. Server hanya dapat memiliki satu metode otentikasi.

Daftar Isi

DirectoryId

Pengidentifikasi AWS Directory Service direktori yang ingin Anda gunakan sebagai penyedia identitas Anda.

Jenis: String

Panjang Batasan: Panjang maksimum 12.

Pola: `d-[0-9a-f]{10}`

Diperlukan: Tidak

Function

ARN untuk fungsi Lambda untuk digunakan untuk penyedia Identitas.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 170.

Pola: `arn:[a-z-]+:lambda:.*`

Diperlukan: Tidak

InvocationRole

Parameter ini hanya berlaku jika Anda `IdentityProviderTypeAPI_GATEWAY`. Menyediakan tipe `InvocationRole` yang digunakan untuk mengotentikasi akun pengguna.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 2048.

Pola: `arn:.*role/\S+`

Diperlukan: Tidak

SftpAuthenticationMethods

Untuk server berkemampuan SFTP, dan hanya untuk penyedia identitas khusus, Anda dapat menentukan apakah akan mengautentikasi menggunakan kata sandi, SSH key pair, atau keduanya.

- `PASSWORD`- pengguna harus memberikan kata sandi mereka untuk terhubung.
- `PUBLIC_KEY`- pengguna harus memberikan kunci pribadi mereka untuk terhubung.
- `PUBLIC_KEY_OR_PASSWORD`- pengguna dapat mengautentikasi dengan kata sandi atau kunci mereka. Ini adalah nilai default.
- `PUBLIC_KEY_AND_PASSWORD`- pengguna harus memberikan kunci pribadi dan kata sandi mereka untuk terhubung. Server memeriksa kunci terlebih dahulu, dan kemudian jika kuncinya valid, sistem meminta kata sandi. Jika kunci pribadi yang diberikan tidak cocok dengan kunci publik yang disimpan, otentikasi gagal.

Jenis: String

Nilai yang Valid: `PASSWORD | PUBLIC_KEY | PUBLIC_KEY_OR_PASSWORD | PUBLIC_KEY_AND_PASSWORD`

Diperlukan: Tidak

Url

Memberikan lokasi titik akhir layanan yang digunakan untuk mengotentikasi pengguna.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum sebesar 255.

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.

- AWSSDK for Java V2
- AWS SDK for Ruby V3

InputFileLocation

Menentukan lokasi untuk file yang sedang diproses.

Daftar Isi

EfsFileLocation

Menentukan detail untuk file Amazon Elastic File System (Amazon EFS) yang sedang didekripsi.

Tipe: Objek EfsFileLocation

Wajib: Tidak

S3FileLocation

Menentukan detail untuk file Amazon S3 yang sedang disalin atau didekripsi.

Tipe: Objek S3InputFileLocation

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWSSDK for C++
- AWSSDK for Go
- AWSSDK for Java V2
- AWSSDK for Ruby V3

ListedAccess

Daftar properti untuk satu atau beberapa akses terkait yang ditentukan.

Daftar Isi

ExternalId

Pengidentifikasi unik yang diperlukan untuk mengidentifikasi grup tertentu dalam direktori Anda. Pengguna grup yang Anda asosiasikan memiliki akses ke sumber daya Amazon S3 atau Amazon EFS Anda melalui protokol yang diaktifkan. AWS Transfer Family Jika Anda tahu nama grup, Anda dapat melihat nilai SID dengan menjalankan perintah berikut menggunakan Windows PowerShell.

```
Get-ADGroup -Filter {samAccountName -like "YourGroupName*"} -Properties
* | Select SamAccountName,ObjectSid
```

Dalam perintah itu, ganti YourGroupNamedengan nama grup Active Directory Anda.

Ekspresi reguler yang digunakan untuk memvalidasi parameter ini adalah string karakter yang terdiri dari huruf besar dan huruf kecil karakter alfanumerik tanpa spasi. Anda juga dapat menyertakan garis bawah atau salah satu karakter berikut: =, . @: /-

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 256.

Pola: `S-1-[\d-]+`

Diperlukan: Tidak

HomeDirectory

Direktori arahan (folder) untuk pengguna ketika mereka masuk ke server menggunakan klien.

Contoh `HomeDirectory` adalah `/bucket_name/home/mydirectory`.

> **Note**
>
> Parameter `HomeDirectory` hanya digunakan jika `HomeDirectoryType` diatur ke `PATH`.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 1024.

Pola: `(|/.*)`

Diperlukan: Tidak

HomeDirectoryType

Jenis direktori pendaratan (folder) yang Anda inginkan direktori home pengguna Anda ketika mereka masuk ke server. Jika Anda mengaturnya`PATH`, pengguna akan melihat bucket Amazon S3 absolut atau jalur Amazon EFS seperti pada klien protokol transfer file mereka. Jika Anda menyetelnya`LOGICAL`, Anda harus menyediakan pemetaan `HomeDirectoryMappings` untuk bagaimana Anda ingin membuat jalur Amazon S3 atau Amazon EFS terlihat oleh pengguna Anda.

> **Note**
>
> Jika `HomeDirectoryType` ya`LOGICAL`, Anda harus memberikan pemetaan, menggunakan parameter. `HomeDirectoryMappings` Jika, di sisi lain, `HomeDirectoryType` adalah`PATH`, Anda memberikan jalur absolut menggunakan `HomeDirectory` parameter. Anda tidak dapat memiliki keduanya `HomeDirectory` dan `HomeDirectoryMappings` di template Anda.

Jenis: String

Nilai yang Valid: `PATH | LOGICAL`

Diperlukan: Tidak

Role

Nama Sumber Daya Amazon (ARN) dari peran AWS Identity and Access Management (IAM) yang mengontrol akses pengguna ke bucket Amazon S3 atau sistem file Amazon EFS. Kebijakan yang dilampirkan pada peran ini menentukan tingkat akses yang ingin Anda berikan kepada pengguna saat mentransfer file masuk dan keluar dari bucket Amazon S3 atau sistem file Amazon EFS. IAM role juga harus berisi hubungan kepercayaan yang mengizinkan server untuk mengakses sumber daya Anda saat melayani permintaan transfer pengguna.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 2048.

Pola: `arn:.*role/\S+`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

ListedAgreement

Menjelaskan sifat-sifat suatu perjanjian.

Daftar Isi

AgreementId

Pengidentifikasi unik untuk perjanjian. Pengenal ini dikembalikan saat Anda membuat perjanjian.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `a-([0-9a-f]{17})`

Diperlukan: Tidak

Arn

Nama Sumber Daya Amazon (ARN) dari perjanjian yang ditentukan.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 1600.

Pola: `arn:\S+`

Diperlukan: Tidak

Description

Deskripsi saat ini untuk perjanjian. Anda dapat mengubahnya dengan memanggil `UpdateAgreement` operasi dan memberikan deskripsi baru.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 200.

Pola: `[\p{Graph}]+`

Diperlukan: Tidak

LocalProfileId

Pengidentifikasi unik untuk profil lokal AS2.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `p-([0-9a-f]{17})`

Diperlukan: Tidak

PartnerProfileId

Pengenal unik untuk profil mitra.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `p-([0-9a-f]{17})`

Diperlukan: Tidak

ServerId

Pengidentifikasi unik untuk perjanjian.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Tidak

Status

Kesepakatan itu bisa berupa `ACTIVE` atau`INACTIVE`.

Jenis: String

Nilai yang Valid: `ACTIVE | INACTIVE`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

ListedCertificate

Menjelaskan properti sertifikat.

Daftar Isi

ActiveDate

Tanggal opsional yang menentukan kapan sertifikat menjadi aktif.

Tipe: Timestamp

Diperlukan: Tidak

Arn

Nama Sumber Daya Amazon (ARN) dari sertifikat yang ditentukan.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 1600.

Pola: `arn:\S+`

Diperlukan: Tidak

CertificateId

Array pengidentifikasi untuk sertifikat yang diimpor. Anda menggunakan pengenal ini untuk bekerja dengan profil dan profil mitra.

Jenis: String

Kendala Panjang: Panjang tetap 22.

Pola: `cert-([0-9a-f]{17})`

Diperlukan: Tidak

Description

Nama atau deskripsi singkat yang digunakan untuk mengidentifikasi sertifikat.

Jenis: String

Panjang Batasan: Panjang minimum 1. Panjang maksimum 200.

Pola: `[\p{Graph}]+`

Diperlukan: Tidak

InactiveDate

Tanggal opsional yang menentukan kapan sertifikat menjadi tidak aktif.

Tipe: Timestamp

Diperlukan: Tidak

Status

Sertifikat dapat berupa`ACTIVE`,`PENDING_ROTATION`, atau`INACTIVE`. `PENDING_ROTATION`berarti bahwa sertifikat ini akan menggantikan sertifikat saat ini ketika kedaluwarsa.

Jenis: String

Nilai yang Valid: `ACTIVE | PENDING_ROTATION | INACTIVE`

Diperlukan: Tidak

Type

Jenis untuk sertifikat. Jika kunci pribadi telah ditentukan untuk sertifikat, jenisnya adalah`CERTIFICATE_WITH_PRIVATE_KEY`. Jika tidak ada kunci pribadi, tipenya adalah`CERTIFICATE`.

Jenis: String

Nilai yang Valid: `CERTIFICATE | CERTIFICATE_WITH_PRIVATE_KEY`

Diperlukan: Tidak

Usage

Menentukan bagaimana sertifikat ini digunakan. Ini dapat digunakan dengan cara-cara berikut:

- `SIGNING`: Untuk menandatangani pesan AS2
- `ENCRYPTION`: Untuk mengenkripsi pesan AS2
- `TLS`: Untuk mengamankan komunikasi AS2 yang dikirim melalui HTTPS

Jenis: String

Nilai yang Valid: `SIGNING | ENCRYPTION`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ListedConnector

Mengembalikan rincian konektor yang ditentukan.

Daftar Isi

Arn

Nama Sumber Daya Amazon (ARN) dari konektor yang ditentukan.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 1600.

Pola: `arn:\S+`

Diperlukan: Tidak

ConnectorId

Pengidentifikasi unik untuk konektor.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `c-([0-9a-f]{17})`

Diperlukan: Tidak

Url

URL titik akhir AS2 atau SFTP mitra.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum sebesar 255.

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

ListedExecution

Mengembalikan properti eksekusi yang ditentukan.

Daftar Isi

ExecutionId

Pengidentifikasi unik untuk eksekusi alur kerja.

Jenis: String

Batas Panjang: Panjang tetap 36.

Pola: `[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}`

Diperlukan: Tidak

InitialFileLocation

Struktur yang menjelaskan lokasi file Amazon S3 atau EFS. Ini adalah lokasi file ketika eksekusi dimulai: jika file sedang disalin, ini adalah lokasi file awal (sebagai lawan dari tujuan).

Tipe: Objek FileLocation

Diperlukan: Tidak

ServiceMetadata

Objek kontainer untuk detail sesi yang terkait dengan alur kerja.

Tipe: Objek ServiceMetadata

Diperlukan: Tidak

Status

Status adalah salah satu eksekusi. Dapat dalam proses, selesai, pengecualian ditemui, atau menangani pengecualian.

Jenis: String

Nilai yang Valid: `IN_PROGRESS | COMPLETED | EXCEPTION | HANDLING_EXCEPTION`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

ListedHostKey

Mengembalikan properti dari kunci host yang ditentukan.

Daftar Isi

Arn

Nama Sumber Daya Amazon (ARN) unik dari kunci host.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 1600.

Pola: `arn:\S+`

Diperlukan: Ya

DateImported

Tanggal di mana kunci host ditambahkan ke server.

Tipe: Timestamp

Diperlukan: Tidak

Description

Deskripsi saat ini untuk kunci host. Anda dapat mengubahnya dengan memanggil `UpdateHostKey` operasi dan memberikan deskripsi baru.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 200.

Pola: `[\p{Print}]*`

Diperlukan: Tidak

Fingerprint

Sidik jari kunci publik, yang merupakan urutan pendek byte yang digunakan untuk mengidentifikasi kunci publik yang lebih panjang.

Tipe: String

Wajib: Tidak

HostKeyId

Pengidentifikasi unik untuk kunci host.

Jenis: String

Kendala Panjang: Panjang tetap 25.

Pola: `hostkey-[0-9a-f]{17}`

Diperlukan: Tidak

Type

Algoritma enkripsi yang digunakan untuk kunci host. `Type`Parameter ditentukan dengan menggunakan salah satu nilai berikut:

- `ssh-rsa`
- `ssh-ed25519`
- `ecdsa-sha2-nistp256`
- `ecdsa-sha2-nistp384`
- `ecdsa-sha2-nistp521`

Tipe: String

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

ListedProfile

Mengembalikan properti dari profil yang telah ditentukan.

Daftar Isi

Arn

Nama Sumber Daya Amazon (ARN) dari profil yang ditentukan.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 1600.

Pola: `arn:\S+`

Diperlukan: Tidak

As2Id

`As2Id`Itu adalah nama AS2, seperti yang didefinisikan dalam RFC 4130. Untuk transfer masuk, ini adalah `AS2-From` header untuk pesan AS2 yang dikirim dari mitra. Untuk konektor keluar, ini adalah `AS2-To` header untuk pesan AS2 yang dikirim ke mitra menggunakan operasi `StartFileTransfer` API. ID ini tidak dapat menyertakan spasi.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: `[\p{Print}\s]*`

Diperlukan: Tidak

ProfileId

Pengenal unik untuk profil AS2 lokal atau mitra.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `p-([0-9a-f]{17})`

Diperlukan: Tidak

ProfileType

Menunjukkan apakah hanya akan mencantumkan profil `LOCAL` tipe atau hanya `PARTNER` mengetik profil. Jika tidak disediakan dalam permintaan, perintah mencantumkan semua jenis profil.

Jenis: String

Nilai yang Valid: `LOCAL | PARTNER`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

ListedServer

Mengembalikan properti dari server berkemampuan protokol transfer file yang ditentukan.

Daftar Isi

Arn

Menentukan Nama Sumber Daya Amazon (ARN) unik untuk server yang akan dicantumkan.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 1600.

Pola: `arn:\S+`

Diperlukan: Ya

Domain

Menentukan domain sistem penyimpanan yang digunakan untuk transfer file. Ada dua domain yang tersedia: Amazon Simple Storage Service (Amazon S3) dan Amazon Elastic File System (Amazon EFS). Nilai defaultnya adalah S3.

Jenis: String

Nilai yang Valid: `S3 | EFS`

Diperlukan: Tidak

EndpointType

Menentukan jenis titik akhir VPC yang terhubung dengan server Anda. Jika server Anda terhubung ke titik akhir VPC, server Anda tidak dapat diakses melalui internet publik.

Jenis: String

Nilai yang Valid: `PUBLIC | VPC | VPC_ENDPOINT`

Diperlukan: Tidak

IdentityProviderType

Modus otentikasi untuk server. Nilai defaultnya adalah`SERVICE_MANAGED`, yang memungkinkan Anda untuk menyimpan dan mengakses kredensyal pengguna dalam layanan. AWS Transfer Family

Gunakan `AWS_DIRECTORY_SERVICE` untuk menyediakan akses ke grup Direktori Aktif di AWS Directory Service for Microsoft Active Directory atau Microsoft Active Directory di lingkungan lokal Anda atau AWS menggunakan AD Connector. Opsi ini juga mengharuskan Anda untuk memberikan ID Direktori dengan menggunakan `IdentityProviderDetails` parameter.

Gunakan nilai `API_GATEWAY` untuk mengintegrasikan dengan penyedia identitas pilihan Anda. `API_GATEWAY`Pengaturan mengharuskan Anda untuk menyediakan URL titik akhir Amazon API Gateway untuk memanggil otentikasi dengan menggunakan parameter. `IdentityProviderDetails`

Gunakan `AWS_LAMBDA` nilai untuk langsung menggunakan AWS Lambda fungsi sebagai penyedia identitas Anda. Jika Anda memilih nilai ini, Anda harus menentukan ARN untuk fungsi Lambda dalam `Function` parameter untuk tipe data. `IdentityProviderDetails`

Jenis: String

Nilai yang Valid: `SERVICE_MANAGED | API_GATEWAY | AWS_DIRECTORY_SERVICE | AWS_LAMBDA`

Diperlukan: Tidak

LoggingRole

Nama Sumber Daya Amazon (ARN) dari peran AWS Identity and Access Management (IAM) yang memungkinkan server mengaktifkan CloudWatch pencatatan Amazon untuk Amazon S3 atau Amazon EFSevents. Saat disetel, Anda dapat melihat aktivitas pengguna di CloudWatch log Anda.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 2048.

Pola: `arn:.*role/\S+`

Diperlukan: Tidak

ServerId

Menentukan sistem unik yang ditetapkan identifier untuk server yang terdaftar.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Tidak

State

Kondisi server yang dijelaskan. Nilai `ONLINE` menunjukkan bahwa server dapat menerima pekerjaan dan mentransfer file. `State`Nilai `OFFLINE` berarti bahwa server tidak dapat melakukan operasi transfer file.

Status `STARTING` dan `STOPPING` menunjukkan bahwa server berada dalam keadaan perantara, baik tidak sepenuhnya dapat merespons, atau tidak sepenuhnya offline. Nilai `START_FAILED` atau `STOP_FAILED` dapat menunjukkan kondisi kesalahan.

Jenis: String

Nilai yang Valid: `OFFLINE | ONLINE | STARTING | STOPPING | START_FAILED | STOP_FAILED`

Diperlukan: Tidak

UserCount

Menentukan jumlah pengguna yang ditugaskan ke server yang Anda tentukan dengan. `ServerId`

Tipe: Integer

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go
- AWS SDK for Java V2
- AWS SDK for Ruby V3

ListedUser

Mengembalikan properti pengguna yang Anda tentukan.

Daftar Isi

Arn

Menyediakan Nama Sumber Daya Amazon (ARN) unik untuk pengguna yang ingin Anda pelajari.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 1600.

Pola: `arn:\S+`

Diperlukan: Ya

HomeDirectory

Direktori arahan (folder) untuk pengguna ketika mereka masuk ke server menggunakan klien.

Contoh `HomeDirectory` adalah `/bucket_name/home/mydirectory`.

> **Note**
>
> Parameter `HomeDirectory` hanya digunakan jika `HomeDirectoryType` diatur ke `PATH`.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 1024.

Pola: `(|/.*)`

Diperlukan: Tidak

HomeDirectoryType

Jenis direktori pendaratan (folder) yang Anda inginkan direktori home pengguna Anda ketika mereka masuk ke server. Jika Anda mengaturnya`PATH`, pengguna akan melihat bucket Amazon S3 absolut atau jalur Amazon EFS seperti pada klien protokol transfer file mereka. Jika Anda menyetelnya`LOGICAL`, Anda harus menyediakan pemetaan `HomeDirectoryMappings` untuk bagaimana Anda ingin membuat jalur Amazon S3 atau Amazon EFS terlihat oleh pengguna Anda.

> **Note**
>
> Jika `HomeDirectoryType` yaLOGICAL, Anda harus memberikan pemetaan, menggunakan parameter. `HomeDirectoryMappings` Jika, di sisi lain, `HomeDirectoryType` adalah`PATH`, Anda memberikan jalur absolut menggunakan `HomeDirectory` parameter. Anda tidak dapat memiliki keduanya `HomeDirectory` dan `HomeDirectoryMappings` di template Anda.

Jenis: String

Nilai yang Valid: `PATH | LOGICAL`

Diperlukan: Tidak

Role

Nama Sumber Daya Amazon (ARN) dari peran AWS Identity and Access Management (IAM) yang mengontrol akses pengguna ke bucket Amazon S3 atau sistem file Amazon EFS. Kebijakan yang dilampirkan pada peran ini menentukan tingkat akses yang ingin Anda berikan kepada pengguna saat mentransfer file masuk dan keluar dari bucket Amazon S3 atau sistem file Amazon EFS. IAM role juga harus berisi hubungan kepercayaan yang mengizinkan server untuk mengakses sumber daya Anda saat melayani permintaan transfer pengguna.

> **Note**
>
> Peran IAM yang mengontrol akses pengguna ke bucket Amazon S3 untuk server `Domain=S3` dengan, atau sistem file EFS Anda untuk server. `Domain=EFS` Kebijakan yang dilampirkan pada peran ini menentukan tingkat akses yang ingin Anda berikan kepada pengguna saat mentransfer file masuk dan keluar dari bucket S3 atau sistem file EFS Anda.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 2048.

Pola: `arn:.*role/\S+`

Diperlukan: Tidak

SshPublicKeyCount

Menentukan jumlah kunci publik SSH yang disimpan untuk pengguna yang Anda tentukan.

Tipe: Integer

Wajib: Tidak

UserName

Menentukan nama pengguna yang ARN ditentukan. Nama pengguna digunakan untuk tujuan otentikasi.

Jenis: String

Batasan Panjang: Panjang minimum 3. Panjang maksimum 100.

Pola: `[\w][\w@.-]{2,99}`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

ListedWorkflow

Berisi pengenal, deskripsi teks, dan Nama Sumber Daya Amazon (ARN) untuk alur kerja.

Daftar Isi

Arn

Menentukan Nama Sumber Daya Amazon (ARN) unik untuk alur kerja.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 1600.

Pola: `arn:\S+`

Diperlukan: Tidak

Description

Menentukan deskripsi teks untuk alur kerja.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 256.

Pola: `[\w- ]*`

Diperlukan: Tidak

WorkflowId

Pengidentifikasi unik untuk alur kerja.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `w-([a-z0-9]{17})`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

LoggingConfiguration

Terdiri dari peran logging dan nama grup log.

Daftar Isi

LoggingRole

Nama Sumber Daya Amazon (ARN) dari peran AWS Identity and Access Management (IAM) yang memungkinkan server mengaktifkan CloudWatch pencatatan Amazon untuk Amazon S3 atau Amazon EFSevents. Saat disetel, Anda dapat melihat aktivitas pengguna di CloudWatch log Anda.

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 2048.

Pola: `arn:.*role/\S+`

Diperlukan: Tidak

LogGroupName

Nama grup CloudWatch logging untuk AWS Transfer Family server tempat alur kerja ini berada.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 512.

Pola: `[\.\-_/#A-Za-z0-9]*`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2

- AWS SDK for Ruby V3

PosixProfile

Identitas POSIX lengkap, termasuk ID pengguna (`Uid`), ID grup (`Gid`), dan setiap grup sekunder ID (`SecondaryGids`), yang mengendalikan akses pengguna Anda ke sistem file Amazon EFS Anda. POSIX izin yang ditetapkan pada file dan direktori dalam sistem file Anda menentukan tingkat akses yang pengguna Anda dapatkan ketika mentransfer file ke dalam dan keluar dari sistem file Amazon EFS Anda.

Daftar Isi

Gid

ID grup POSIX digunakan untuk semua operasi EFS oleh pengguna ini.

Tipe: Long

Rentang yang Valid: Nilai minimum 0. Nilai maksimum 4294967295.

Wajib: Ya

Uid

ID pengguna POSIX digunakan untuk semua operasi EFS oleh pengguna ini.

Tipe: Long

Rentang yang Valid: Nilai minimum 0. Nilai maksimum 4294967295.

Wajib: Ya

SecondaryGids

ID grup POSIX sekunder yang digunakan untuk semua operasi EFS oleh pengguna ini.

Tipe: Array panjang

Anggota Array: Jumlah minimum 0 item. Jumlah maksimum 16 item.

Rentang yang Valid: Nilai minimum 0. Nilai maksimum 4294967295.

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWSSDK for C++
- AWSSDK for Go
- AWSSDK for Java V2
- AWSSDK for Ruby V3

ProtocolDetails

Pengaturan protokol yang dikonfigurasi untuk server Anda.

Daftar Isi

As2Transports

Menunjukkan metode transport untuk pesan AS2. Saat ini, hanya HTTP yang didukung.

Tipe: Array string

Anggota Array: Jumlah tetap 1 item.

Nilai yang Valid: `HTTP`

Diperlukan: Tidak

PassiveIp

Menunjukkan mode pasif, untuk protokol FTP dan FTPS. Masukkan satu alamat IPv4, seperti alamat IP publik firewall, router, atau penyeimbang beban. Contoh:

```
aws transfer update-server --protocol-details PassiveIp=0.0.0.0
```

Ganti `0.0.0.0` pada contoh di atas dengan alamat IP aktual yang ingin Anda gunakan.

> **Note**
>
> Jika Anda mengubah `PassiveIp` nilainya, Anda harus berhenti dan kemudian memulai ulang server Transfer Family Anda agar perubahan diterapkan. Untuk detail tentang penggunaan mode pasif (PASV) di lingkungan NAT, lihat Mengonfigurasi server FTPS Anda di belakang firewall atau NAT dengan. AWS Transfer Family

Nilai khusus

`AUTO`Dan `0.0.0.0` merupakan nilai khusus untuk `PassiveIp` parameter. Nilai ditetapkan `PassiveIp=AUTO` secara default ke server tipe FTP dan FTPS. Dalam hal ini, server secara otomatis merespons dengan salah satu IP endpoint dalam respons PASV. `PassiveIp=0.0.0.0`memiliki aplikasi yang lebih unik untuk penggunaannya. Misalnya, jika Anda memiliki lingkungan High Availability (HA) Network Load Balancer (NLB), di mana Anda

memiliki 3 subnet, Anda hanya dapat menentukan satu alamat IP menggunakan parameter. `PassiveIp` Ini mengurangi efektivitas memiliki Ketersediaan Tinggi. Dalam hal ini, Anda dapat menentukan`PassiveIp=0.0.0.0`. Ini memberitahu klien untuk menggunakan alamat IP yang sama dengan koneksi Control dan memanfaatkan semua AZ untuk koneksi mereka. Namun, perhatikan bahwa tidak semua klien FTP mendukung `PassiveIp=0.0.0.0` respons tersebut. FileZilla dan WinSCP mendukungnya. Jika Anda menggunakan klien lain, periksa untuk melihat apakah klien Anda mendukung `PassiveIp=0.0.0.0` respons.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 15.

Diperlukan: Tidak

SetStatOption

Gunakan file `SetStatOption` untuk mengabaikan kesalahan yang dihasilkan saat klien mencoba menggunakan `SETSTAT` pada file yang Anda unggah ke bucket S3.

Beberapa klien transfer file SFTP dapat mencoba mengubah atribut file jarak jauh, termasuk stempel waktu dan izin, menggunakan perintah, seperti saat `SETSTAT` mengunggah file. Namun, perintah ini tidak kompatibel dengan sistem penyimpanan objek, seperti Amazon S3. Karena ketidakcocokan ini, unggahan file dari klien ini dapat mengakibatkan kesalahan bahkan ketika file tersebut berhasil diunggah.

Tetapkan nilainya `ENABLE_NO_OP` agar server Transfer Family mengabaikan `SETSTAT` perintah, dan unggah file tanpa perlu membuat perubahan apa pun pada klien SFTP Anda. Meskipun `SetStatOption` `ENABLE_NO_OP` pengaturan mengabaikan kesalahan, itu menghasilkan entri log di Amazon CloudWatch Logs, sehingga Anda dapat menentukan kapan klien melakukan `SETSTAT` panggilan.

> **Note**
>
> Jika Anda ingin mempertahankan stempel waktu asli untuk file Anda, dan memodifikasi atribut file lain menggunakan`SETSTAT`, Anda dapat menggunakan Amazon EFS sebagai penyimpanan backend dengan Transfer Family.

Jenis: String

Nilai yang Valid: `DEFAULT | ENABLE_NO_OP`

Diperlukan: Tidak

TlsSessionResumptionMode

Properti yang digunakan dengan server Transfer Family yang menggunakan protokol FTPS. TLS Session Resumption menyediakan mekanisme untuk melanjutkan atau berbagi kunci rahasia yang dinegosiasikan antara kontrol dan koneksi data untuk sesi FTPS. `TlsSessionResumptionMode`menentukan apakah server melanjutkan sesi terbaru yang dinegosiasikan melalui ID sesi unik atau tidak. Akomodasi ini tersedia selama `CreateServer` dan `UpdateServer` panggilan telepon. Jika `TlsSessionResumptionMode` nilai tidak ditentukan selama`CreateServer`, itu diatur ke secara `ENFORCED` default.

- `DISABLED`: server tidak memproses permintaan klien dimulainya kembali sesi TLS dan membuat sesi TLS baru untuk setiap permintaan.
- `ENABLED`: server memproses dan menerima klien yang melakukan dimulainya kembali sesi TLS. Server tidak menolak koneksi data klien yang tidak melakukan pemrosesan klien dimulainya kembali sesi TLS.
- `ENFORCED`: server memproses dan menerima klien yang melakukan dimulainya kembali sesi TLS. Server menolak koneksi data klien yang tidak melakukan pemrosesan klien dimulainya kembali sesi TLS. Sebelum Anda menetapkan nilainya`ENFORCED`, uji klien Anda.

> **Note**
>
> Tidak semua klien FTPS melakukan dimulainya kembali sesi TLS. Jadi, jika Anda memilih untuk menegakkan dimulainya kembali sesi TLS, Anda mencegah koneksi apa pun dari klien FTPS yang tidak melakukan negosiasi protokol. Untuk menentukan apakah Anda dapat menggunakan `ENFORCED` nilainya atau tidak, Anda perlu menguji klien Anda.

Jenis: String

Nilai yang Valid: `DISABLED | ENABLED | ENFORCED`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

S3FileLocation

Menentukan rincian untuk lokasi file untuk file yang sedang digunakan dalam alur kerja. Hanya berlaku jika Anda menggunakan penyimpanan S3.

Daftar Isi

Bucket

Menentukan bucket S3 yang berisi file yang digunakan.

Jenis: String

Batasan Panjang: Panjang minimum 3. Panjang maksimum 63.

Pola: `[a-z0-9][\.\-a-z0-9]{1,61}[a-z0-9]`

Diperlukan: Tidak

Etag

Tag entitas adalah hash dari objek. ETag mencerminkan perubahan hanya pada konten suatu objek, bukan metadata-nya.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 65536.

Pola: `.+`

Diperlukan: Tidak

Key

Nama yang ditetapkan ke file saat dibuat di Amazon S3. Anda harus menggunakan kunci objek tersebut untuk mengambil objeknya.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 1024.

Pola: `[\P{M}\p{M}]*`

Diperlukan: Tidak

VersionId

Menentukan versi file.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 1024.

Pola: `.+`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

S3InputFileLocation

Menentukan lokasi file Amazon S3 masukan pelanggan. Jika digunakan di dalam`copyStepDetails.DestinationFileLocation`, itu harus menjadi tujuan salinan S3.

Anda perlu menyediakan ember dan kunci. Kunci dapat mewakili jalur atau file. Ini ditentukan oleh apakah Anda mengakhiri nilai kunci dengan karakter garis miring maju (/). Jika karakter terakhir adalah “/”, maka file Anda disalin ke folder, dan namanya tidak berubah. Jika, sebaliknya, karakter terakhir adalah alfanumerik, file yang Anda unggah diganti namanya menjadi nilai jalur. Dalam hal ini, jika file dengan nama itu sudah ada, itu ditimpa.

Misalnya, jika jalur Anda`shared-files/bob/`, file yang Anda unggah disalin ke folder`shared-files/bob/`,. Jika jalur Anda*`shared-files/today`*, setiap file yang diunggah disalin ke *`shared-files`* folder dan diberi nama*`today`*: setiap unggahan menimpa versi file bob sebelumnya.

Daftar Isi

Bucket

Menentukan bucket S3 untuk file input pelanggan.

Jenis: String

Batasan Panjang: Panjang minimum 3. Panjang maksimum 63.

Pola: `[a-z0-9][\.\-a-z0-9]{1,61}[a-z0-9]`

Diperlukan: Tidak

Key

Nama yang ditetapkan ke file saat dibuat di Amazon S3. Anda harus menggunakan kunci objek tersebut untuk mengambil objeknya.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 1024.

Pola: `[\P{M}\p{M}]*`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

S3StorageOptions

Opsi penyimpanan Amazon S3 yang dikonfigurasi untuk server Anda.

Daftar Isi

DirectoryListingOptimization

Menentukan apakah atau tidak kinerja untuk direktori Amazon S3 Anda dioptimalkan. Ini dinonaktifkan secara default.

Secara default, pemetaan direktori home memiliki `TYPE` file. `DIRECTORY` Jika Anda mengaktifkan opsi ini, Anda kemudian perlu secara eksplisit menyetel `HomeDirectoryMapEntry` `Type` ke `FILE` jika Anda ingin pemetaan memiliki target file.

Jenis: String

Nilai yang Valid: `ENABLED | DISABLED`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

S3Tag

Menentukan pasangan kunci-nilai yang ditugaskan ke file selama pelaksanaan langkah Tagging.

Daftar Isi

Key

Nama yang ditetapkan untuk tag yang Anda buat.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 128.

Pola: `([\p{L}\p{Z}\p{N}_.:/=+\-@]*)`

Diperlukan: Ya

Value

Nilai yang sesuai dengan kunci.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 256.

Pola: `([\p{L}\p{Z}\p{N}_.:/=+\-@]*)`

Diperlukan: Ya

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

ServiceMetadata

Objek kontainer untuk detail sesi yang terkait dengan alur kerja.

Daftar Isi

UserDetails

Server ID (`ServerId`), Session ID (`SessionId`) dan user (`UserName`) membentuk file`UserDetails`.

Tipe: Objek UserDetails

Wajib: Ya

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWSSDK for C++
- AWSSDK for Go
- AWSSDK for Java V2
- AWSSDK for Ruby V3

SftpConnectorConfig

Berisi rincian untuk objek konektor SFTP. Objek konektor digunakan untuk mentransfer file ke dan dari server SFTP mitra.

> **Note**
>
> Karena tipe `SftpConnectorConfig` data digunakan untuk membuat dan memperbarui konektor SFTP, parameternya, `TrustedHostKeys` dan `UserSecretId` ditandai sebagai tidak diperlukan. Ini agak menyesatkan, karena tidak diperlukan saat Anda memperbarui konektor SFTP yang ada, tetapi diperlukan saat Anda membuat konektor SFTP baru.

Daftar Isi

TrustedHostKeys

Bagian publik dari kunci host, atau kunci, yang digunakan untuk mengidentifikasi server eksternal yang Anda sambungkan. Anda dapat menggunakan `ssh-keyscan` perintah terhadap server SFTP untuk mengambil kunci yang diperlukan.

Tiga elemen format kunci publik SSH standar adalah`<key type>`,, dan opsional `<body base64><comment>`, dengan spasi di antara setiap elemen. Tentukan hanya `<key type>` dan`<body base64>`: jangan masukkan `<comment>` bagian kunci.

Untuk kunci host tepercaya, AWS Transfer Family terima kunci RSA dan ECDSA.

- Untuk kunci RSA, `<key type>` string adalah`ssh-rsa`.
- Untuk kunci ECDSA, `<key type>` string adalah`ecdsa-sha2-nistp256`,, atau `ecdsa-sha2-nistp384``ecdsa-sha2-nistp521`, tergantung pada ukuran kunci yang Anda hasilkan.

Jalankan perintah ini untuk mengambil kunci host server SFTP, di mana nama server SFTP Anda berada. `ftp.host.com`

```
ssh-keyscan ftp.host.com
```

Ini mencetak kunci host publik ke output standar.

```
ftp.host.com ssh-rsa AAAAB3Nza...<long-string-for-public-key
```

Salin dan tempel string ini ke `TrustedHostKeys` bidang untuk `create-connector` perintah atau ke bidang Kunci host tepercaya di konsol.

Tipe: Array string

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 10 item.

Batasan Panjang: Panjang minimum 1. Panjang maksimum 2048.

Diperlukan: Tidak

UserSecretId

Pengidentifikasi rahasia (di AWS Secrets Manager) yang berisi kunci pribadi pengguna SFTP, kata sandi, atau keduanya. Pengenal harus berupa Nama Sumber Daya Amazon (ARN) dari rahasianya.

Jenis: String

Batasan Panjang: Panjang minimum 1. Panjang maksimum 2048.

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

SshPublicKey

Memberikan informasi tentang kunci Shell Aman publik (SSH) yang terkait dengan pengguna Transfer Family untuk server berkemampuan protokol transfer file tertentu (sebagaimana diidentifikasi oleh). `ServerId` Informasi yang dikembalikan mencakup tanggal kunci diimpor, konten kunci publik, dan ID kunci publik. Satu pengguna dapat menyimpan lebih dari satu kunci publik SSH yang terkait dengan nama pengguna mereka pada server tertentu.

Daftar Isi

DateImported

Menentukan tanggal ketika kunci publik ditambahkan ke pengguna Transfer Family.

Tipe: Timestamp

Diperlukan: Ya

SshPublicKeyBody

Menentukan isi dari kunci publik SSH seperti yang ditentukan oleh. `PublicKeyId`

AWS Transfer Familymenerima kunci RSA, ECDSA, dan ED25519.

Jenis: String

Batasan Panjang: Panjang minimum sebesar 0. Panjang maksimum 2048.

Diperlukan: Ya

SshPublicKeyId

Menentukan `SshPublicKeyId` parameter berisi identifier dari kunci publik.

Jenis: String

Kendala Panjang: Panjang tetap 21.

Pola: `key-[0-9a-f]{17}`

Diperlukan: Ya

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

Tag

Membuat pasangan kunci-nilai untuk sumber daya tertentu. Tag adalah metadata yang dapat Anda gunakan untuk mencari dan mengelompokkan sumber daya untuk berbagai tujuan. Anda dapat menerapkan tag ke server, pengguna, dan peran. Kunci tag dapat mengambil lebih dari satu nilai. Misalnya, untuk mengelompokkan server untuk tujuan akuntansi, Anda dapat membuat tag yang disebut `Group` dan menetapkan nilai `Research` dan `Accounting` ke grup itu.

Daftar Isi

Key

Nama yang ditetapkan untuk tag yang Anda buat.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 128.

Diperlukan: Ya

Value

Berisi satu atau beberapa nilai yang Anda tetapkan ke nama kunci yang Anda buat.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 256.

Diperlukan: Ya

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

TagStepDetails

Setiap tipe langkah memiliki `StepDetails` strukturnya sendiri.

Pasangan kunci/nilai yang digunakan untuk menandai file selama eksekusi langkah alur kerja.

Daftar Isi

Name

Nama langkah, digunakan sebagai pengenal.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 30.

Pola: `[\w-]*`

Diperlukan: Tidak

SourceFileLocation

Menentukan file mana yang akan digunakan sebagai masukan ke langkah alur kerja: baik output dari langkah sebelumnya, atau file yang awalnya diunggah untuk alur kerja.

- Untuk menggunakan file sebelumnya sebagai input, masukkan`${previous.file}`. Dalam hal ini, langkah alur kerja ini menggunakan file output dari langkah alur kerja sebelumnya sebagai input. Ini adalah nilai default.
- Untuk menggunakan lokasi file yang awalnya diunggah sebagai masukan untuk langkah ini, masukkan`${original.file}`.

Jenis: String

Batasan Panjang: Panjang minimum 0. Panjang maksimum 256.

Pola: `\$\{(\w+.)+\w+\}`

Diperlukan: Tidak

Tags

Array yang berisi dari 1 hingga 10 pasangan kunci/nilai.

Tipe: Array objek S3Tag

Anggota Array: Jumlah minimum 1 item. Jumlah maksimum 10 item.

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

UserDetails

Menentukan nama pengguna, ID server, dan ID sesi untuk alur kerja.

Daftar Isi

ServerId

Pengidentifikasi unik yang ditetapkan sistem untuk instance server Transfer.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `s-([0-9a-f]{17})`

Diperlukan: Ya

UserName

String unik yang mengidentifikasi pengguna Transfer Family yang terkait dengan server.

Jenis: String

Batasan Panjang: Panjang minimum 3. Panjang maksimum 100.

Pola: `[\w][\w@.-]{2,99}`

Diperlukan: Ya

SessionId

Pengidentifikasi unik yang ditetapkan sistem untuk sesi yang sesuai dengan alur kerja.

Jenis: String

Batasan Panjang: Panjang minimum 3. Panjang maksimum sebesar 32.

Pola: `[\w-]*`

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++
- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

WorkflowDetail

Menentukan ID alur kerja untuk alur kerja yang akan ditetapkan dan peran eksekusi yang digunakan untuk mengeksekusi alur kerja.

Selain alur kerja untuk mengeksekusi ketika file diunggah sepenuhnya, juga `WorkflowDetails` dapat berisi ID alur kerja (dan peran eksekusi) untuk alur kerja untuk mengeksekusi pada upload sebagian. Upload sebagian terjadi ketika sesi server terputus saat file masih diunggah.

Daftar Isi

ExecutionRole

Termasuk izin yang diperlukan untuk operasi S3, EFS, dan Lambda yang dapat diasumsikan oleh Transfer, sehingga semua langkah alur kerja dapat beroperasi pada sumber daya yang diperlukan

Jenis: String

Batasan Panjang: Panjang minimum 20. Panjang maksimum 2048.

Pola: `arn:.*role/\S+`

Diperlukan: Ya

WorkflowId

Pengidentifikasi unik untuk alur kerja.

Jenis: String

Kendala Panjang: Panjang tetap 19.

Pola: `w-([a-z0-9]{17})`

Diperlukan: Ya

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++

- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

WorkflowDetails

Wadah untuk tipe `WorkflowDetail` data. Ini digunakan oleh tindakan yang memicu alur kerja untuk memulai eksekusi.

Daftar Isi

OnPartialUpload

Pemicu yang memulai alur kerja jika file hanya diunggah sebagian. Anda dapat melampirkan alur kerja ke server yang mengeksekusi setiap kali ada unggahan sebagian.

Unggahan sebagian terjadi saat file terbuka saat sesi terputus.

Tipe: Array objek WorkflowDetail

Anggota Array: Jumlah minimum 0 item. Jumlah maksimum 1 item.

Diperlukan: Tidak

OnUpload

Pemicu yang memulai alur kerja: alur kerja mulai dijalankan setelah file diunggah.

Untuk menghapus alur kerja terkait dari server, Anda dapat memberikan `OnUpload` objek kosong, seperti pada contoh berikut.

```
aws transfer update-server --server-id s-01234567890abcdef --workflow-details '{"OnUpload":[]}'
```

Tipe: Array objek WorkflowDetail

Anggota Array: Jumlah minimum 0 item. Jumlah maksimum 1 item.

Diperlukan: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWS SDK for C++

- AWS SDK for Go.
- AWSSDK for Java V2
- AWS SDK for Ruby V3

WorkflowStep

Blok bangunan dasar alur kerja.

Daftar Isi

CopyStepDetails

Detail untuk langkah yang melakukan salinan file.

Terdiri dari nilai-nilai berikut:

- Sebuah deskripsi
- Lokasi Amazon S3 untuk tujuan salinan file.
- Bendera yang menunjukkan apakah akan menimpa file yang ada dengan nama yang sama. Defaultnya adalah `FALSE`.

Tipe: Objek CopyStepDetails

Wajib: Tidak

CustomStepDetails

Detail untuk langkah yang memanggil AWS Lambda fungsi.

Terdiri dari nama, target, dan batas waktu fungsi Lambda (dalam hitungan detik).

Tipe: Objek CustomStepDetails

Wajib: Tidak

DecryptStepDetails

Detail untuk langkah yang mendekripsi file terenkripsi.

Terdiri dari nilai-nilai berikut:

- Nama deskriptif
- Lokasi Amazon S3 atau Amazon Elastic File System (Amazon EFS) untuk file sumber untuk didekripsi.
- Lokasi S3 atau Amazon EFS untuk tujuan dekripsi file.
- Bendera yang menunjukkan apakah akan menimpa file yang ada dengan nama yang sama. Defaultnya adalah `FALSE`.
- Jenis enkripsi yang digunakan. Saat ini, hanya enkripsi PGP yang didukung.

Tipe: Objek DecryptStepDetails

Wajib: Tidak

DeleteStepDetails

Detail untuk langkah yang menghapus file.

Tipe: Objek DeleteStepDetails

Wajib: Tidak

TagStepDetails

Detail untuk langkah yang membuat satu atau beberapa tag.

Anda menentukan satu atau lebih tag. Setiap tag berisi pasangan kunci-nilai.

Tipe: Objek TagStepDetails

Wajib: Tidak

Type

Saat ini, jenis langkah berikut didukung.

- **`COPY`**- Salin file ke lokasi lain.
- **`CUSTOM`**- Lakukan langkah khusus dengan target AWS Lambda fungsi.
- **`DECRYPT`**- Dekripsi file yang dienkripsi sebelum diunggah.
- **`DELETE`**- Hapus file.
- **`TAG`**- Tambahkan tag ke file.

Jenis: String

Nilai yang Valid: `COPY | CUSTOM | TAG | DELETE | DECRYPT`

Wajib: Tidak

Lihat Juga

Untuk informasi selengkapnya tentang penggunaan API ini di salah satu AWS SDK khusus bahasa, lihat berikut ini:

- AWSSDK for C++
- AWSSDK for Go

- AWSSDK for Java V2
- AWSSDK for Ruby V3

Membuat permintaan API

Selain menggunakan konsol, Anda dapat menggunakan AWS Transfer Family API untuk mengonfigurasi dan mengelola server secara terprogram. Bagian ini menjelaskan AWS Transfer Family operasi, penandatanganan permintaan untuk otentikasi, dan penanganan kesalahan. Untuk informasi tentang wilayah dan titik akhir yang tersedia untuk Transfer Family, lihat AWS Transfer Familytitik akhir dan kuota di Referensi Umum AWS

> **Note**
>
> Anda juga dapat menggunakan AWS SDK saat mengembangkan aplikasi dengan Transfer Family;. AWSSDK untuk Java, .NET, dan PHP membungkus Transfer Family API yang mendasarinya, menyederhanakan tugas pemrograman Anda. Untuk informasi tentang mengunduh perpustakaan SDK, lihat Perpustakaan kode sampel.

Topik

- Transfer Family membutuhkan header permintaan
- Transfer Family meminta masukan dan penandatanganan
- Tanggapan kesalahan
- Pustaka yang tersedia

Transfer Family membutuhkan header permintaan

Bagian ini menjelaskan header yang diperlukan yang harus Anda kirim dengan setiap permintaan POST. AWS Transfer Family Anda menyertakan header HTTP untuk mengidentifikasi informasi kunci tentang permintaan termasuk operasi yang ingin Anda panggil, tanggal permintaan, dan informasi yang menunjukkan otorisasi Anda sebagai pengirim permintaan. Header tidak peka huruf besar/kecil dan urutan header tidak penting.

Contoh berikut menunjukkan header yang digunakan dalam ListServersoperasi.

```
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}
```

Berikut ini adalah header yang harus disertakan dengan permintaan POST Anda ke Transfer Family. Header yang ditunjukkan di bawah ini yang dimulai dengan “x-amz” khusus untuk. AWS Semua header lain yang terdaftar adalah header umum yang digunakan dalam transaksi HTTP.

Header	Deskripsi
`Authorization`	Header otorisasi diperlukan. Formatnya adalah tanda tangan permintaa n Sigv4 standar, yang didokumentasikan pada permintaan AWSAPI Penandatanganan.
`Content-Type`	Gunakan `application/x-amz-json-1.1` sebagai jenis konten untuk semua permintaan ke Transfer Family. `Content-Type: application/x-amz-json-1.1`
`Host`	Gunakan header host untuk menentukan titik akhir Transfer Family tempat Anda mengirim permintaan. Misalnya, `transfer.us-east-1 .amazonaws.com` adalah titik akhir untuk wilayah AS Timur (Ohio). Untuk informasi selengkapnya tentang titik akhir yang tersedia untuk Transfer Family, lihat AWS Transfer Familytitik akhir dan kuota di. Referensi Umum AWS `Host: transfer. region.amazonaws.com`
`x-amz-date`	Anda harus memberikan cap waktu baik di `Date` header HTTP atau AWS `x-amz-date` header. (Beberapa pustaka klien HTTP tidak

Header	Deskripsi
	mengizinkan Anda mengatur `Date` header.) Saat `x-amz-date` header hadir, Transfer Family mengabaikan `Date` header apa pun selama otentikasi permintaan. Formatnya harus ISO8601, `x-amz-date` dalam format YYYYMMDD'T'HHMMSS'Z'. `x-amz-date: YYYYMMDD'T'HHMMSS'Z'`
`x-amz-target`	Header ini menentukan versi API dan operasi yang Anda minta. Nilai header target dibentuk dengan menggabungkan versi API dengan nama API dan dalam format berikut. `x-amz-target: TransferService. operationName` Nilai operationName (`ListServers` misalnya) dapat ditemukan dari daftar API,. ListServers
`x-amz-security-token`	Header ini diperlukan ketika kredensi yang digunakan untuk menandatangani permintaan bersifat sementara atau kredensi sesi (untuk detailnya, lihat Menggunakan kredensial sementara dengan AWS sumber daya di Panduan Pengguna IAM. Lihat Menambahkan tanda tangan ke permintaan HTTP di Referensi Umum Amazon Web Services untuk informasi selengkapnya.

Transfer Family meminta masukan dan penandatanganan

Semua input permintaan harus dikirim sebagai bagian dari muatan JSON di badan permintaan. Untuk Tindakan di mana semua bidang permintaan bersifat opsional`ListServers`, misalnya, Anda masih perlu menyediakan objek JSON kosong di badan permintaan, seperti`{}`. Struktur permintaan/respons payload Transfer Family didokumentasikan dalam referensi API yang ada, misalnya. DescribeServer

Transfer Family mendukung otentikasi menggunakan AWS Signature Versi 4. Untuk detailnya, lihat Menandatangani permintaan AWS API.

Tanggapan kesalahan

Ketika ada kesalahan, informasi header respons berisi:

- Tipe Konten: `application/x-amz-json-1.1`
- Kode status yang sesuai `4xx` atau `5xx` HTTP

Tubuh respons kesalahan berisi informasi tentang kesalahan yang terjadi. Contoh respon kesalahan berikut menunjukkan sintaks output elemen respon umum untuk semua respon kesalahan.

```
{
    "__type": "String",
    "Message": "String", <!-- Message is lowercase in some instances -->
    "Resource": String,
    "ResourceType": String
    "RetryAfterSeconds": String
}
```

Tabel berikut menjelaskan bidang respons kesalahan JSON yang ditunjukkan dalam sintaks sebelumnya.

__jenis

Salah satu pengecualian dari panggilan Transfer Family API.

Jenis: String

Pesan atau pesan

Salah satu pesan kode kesalahan operasi.

> **Note**
>
> Beberapa pengecualian digunakan`message`, dan yang lainnya menggunakan`Message`. Anda dapat memeriksa kode untuk antarmuka Anda untuk menentukan kasus yang tepat. Atau, Anda dapat menguji setiap opsi untuk melihat mana yang berfungsi.

Jenis: String

Sumber

Sumber daya yang kesalahannya dipanggil. Misalnya, jika Anda mencoba membuat pengguna yang sudah ada, itu `Resource` adalah nama pengguna untuk pengguna yang ada.

Jenis: String

ResourceType

Jenis sumber daya yang kesalahannya dipanggil. Misalnya, jika Anda mencoba membuat pengguna yang sudah ada, `ResourceType` is`User`.

Jenis: String

RetryAfterSeconds

Jumlah detik untuk menunggu sebelum mencoba kembali perintah.

Jenis: String

Contoh respons kesalahan

Badan JSON berikut dikembalikan jika Anda memanggil `DescribeServer` API dan menentukan server yang tidak ada.

```
{
  "__type": "ResourceNotFoundException",
  "Message": "Unknown server",
  "Resource": "s-11112222333344444",
  "ResourceType": "Server"
}
```

Badan JSON berikut dikembalikan jika menjalankan API menyebabkan pelambatan terjadi.

```
{
   "__type":"ThrottlingException",
   "RetryAfterSeconds":"1"
}
```

Badan JSON berikut dikembalikan jika Anda menggunakan `CreateServer` API dan Anda tidak memiliki izin yang cukup untuk membuat server Transfer Family.

```
{
  "__type": "AccessDeniedException",
  "Message": "You do not have sufficient access to perform this action."
}
```

Badan JSON berikut dikembalikan jika Anda menggunakan `CreateUser` API dan menentukan pengguna yang sudah ada.

```
{
  "__type": "ResourceExistsException",
  "Message": "User already exists",
  "Resource": "Alejandro-Rosalez",
  "ResourceType": "User"
}
```

Pustaka yang tersedia

AWSmenyediakan pustaka, kode sampel, tutorial, dan sumber daya lainnya untuk pengembang perangkat lunak yang lebih suka membangun aplikasi menggunakan API khusus bahasa alih-alih alat baris perintah dan API Kueri. Pustaka ini menyediakan fungsi dasar (tidak termasuk dalam API), seperti otentikasi permintaan, percobaan ulang permintaan, dan penanganan kesalahan sehingga lebih mudah untuk memulai. Lihat Alat untuk dibangun AWS

Untuk pustaka dan kode sampel dalam semua bahasa, lihat Contoh kode & pustaka.

Parameter Umum

Daftar berikut berisi parameter yang digunakan semua tindakan untuk menandatangani permintaan Tanda Tangan Versi 4 dengan string kueri. Setiap parameter khusus tindakan tercantum dalam topik untuk tindakan tersebut. Untuk informasi selengkapnya tentang Tanda Tangan Versi 4, lihat Menandatangani permintaan AWS API di Panduan Pengguna IAM.

Action

Tindakan yang harus dilakukan.

Tipe: string

Wajib: Ya

Version

Versi API yang ditulis dalam permintaan, dinyatakan dalam format HH-BB-TTTT.

Tipe: string

Wajib: Ya

X-Amz-Algorithm

Algoritme hash yang Anda gunakan untuk membuat tanda tangan permintaan.

Syarat: Tentukan parameter ini ketika Anda menyertakan informasi autentikasi dalam string kueri alih-alih di header otorisasi HTTP.

Tipe: string

Nilai yang Valid: `AWS4-HMAC-SHA256`

Diperlukan: Kondisional

X-Amz-Credential

Nilai lingkup kredensial, yang merupakan string yang menyertakan access key Anda, tanggal, wilayah yang Anda targetkan, layanan yang Anda minta, dan string penghentian ("aws4_request"). Nilai dinyatakan dalam format berikut: access_key/HHBBTTTT/wilayah/layanan/aws4_request.

Untuk informasi selengkapnya, lihat Membuat permintaan AWS API yang ditandatangani di Panduan Pengguna IAM.

Syarat: Tentukan parameter ini ketika Anda menyertakan informasi autentikasi dalam string kueri alih-alih di header otorisasi HTTP.

Tipe: string

Diperlukan: Kondisional

X-Amz-Date

Tanggal yang digunakan untuk membuat tanda tangan. Format harus berupa format dasar ISO 8601 (YYYYMMDD'T'HMMSS'Z'). Misalnya, waktu tanggal berikut adalah nilai X-Amz-Date yang valid: `20120325T120000Z`.

Syarat: X-Amz-Date bersifat opsional untuk semua permintaan; nilai ini dapat digunakan untuk mengganti tanggal yang digunakan untuk menandatangani permintaan. Jika header Tanggal

ditentukan dalam format dasar ISO 8601, X-Amz-Date tidak diperlukan. Ketika X-Amz-Date digunakan, ia selalu mengganti nilai header Tanggal. Untuk informasi selengkapnya, lihat Elemen tanda tangan permintaan AWS API di Panduan Pengguna IAM.

Tipe: string

Diperlukan: Kondisional

X-Amz-Security-Token

Token keamanan sementara yang diperoleh melalui panggilan ke AWS Security Token Service (AWS STS). Untuk daftar layanan yang mendukung kredensi keamanan sementaraAWS STS, lihat layanan Layanan AWSyang berfungsi dengan IAM di Panduan Pengguna IAM.

Kondisi: Jika Anda menggunakan kredensi keamanan sementara dariAWS STS, Anda harus menyertakan token keamanan.

Tipe: string

Diperlukan: Kondisional

X-Amz-Signature

Menentukan tanda tangan yang dikodekan oleh hex yang dihitung dari string to sign dan kunci penandatanganan turunan.

Syarat: Tentukan parameter ini ketika Anda menyertakan informasi autentikasi dalam string kueri alih-alih di header otorisasi HTTP.

Tipe: string

Diperlukan: Kondisional

X-Amz-SignedHeaders

Menentukan semua header HTTP yang disertakan sebagai bagian dari permintaan kanonik. Untuk informasi selengkapnya tentang menentukan header yang ditandatangani, lihat Membuat permintaan AWS API yang ditandatangani di Panduan Pengguna IAM.

Syarat: Tentukan parameter ini ketika Anda menyertakan informasi autentikasi dalam string kueri alih-alih di header otorisasi HTTP.

Tipe: string

Diperlukan: Kondisional

Kesalahan Umum

Bagian ini berisi daftar kesalahan yang umum terjadi pada tindakan API dari semua layanan AWS. Untuk kesalahan khusus pada tindakan API untuk layanan ini, lihat topik untuk tindakan API tersebut.

AccessDeniedException

Anda tidak memiliki akses yang memadai untuk melakukan tindakan ini.

Kode Status HTTP: 400

IncompleteSignature

Tanda tangan permintaan tidak sesuai dengan standar AWS.

Kode Status HTTP: 400

InternalFailure

Pemrosesan permintaan telah gagal karena kesalahan yang tidak diketahui, pengecualian atau kegagalan.

Kode Status HTTP: 500

InvalidAction

Tindakan atau operasi yang diminta tidak valid. Verifikasi bahwa tindakan diketik dengan benar.

Kode Status HTTP: 400

InvalidClientTokenId

Sertifikat X.509 atau access key ID AWS yang diberikan tidak ada dalam catatan kami.

Kode Status HTTP: 403

NotAuthorized

Anda tidak memiliki izin untuk melakukan tindakan ini.

Kode Status HTTP: 400

OptInRequired

Access key ID AWS membutuhkan berlangganan untuk layanan.

Kode Status HTTP: 403

RequestExpired

Permintaan menjangkau layanan lebih dari 15 menit setelah stempel tanggal pada permintaan atau lebih dari 15 menit setelah tanggal kedaluwarsa permintaan (seperti untuk URL pre-signed), atau stempel tanggal pada permintaan lebih dari 15 menit di masa mendatang.

Kode Status HTTP: 400

ServiceUnavailable

Permintaan telah gagal karena kegagalan sementara server.

Kode Status HTTP: 503

ThrottlingException

Permintaan ditolak karena throttling permintaan.

Kode Status HTTP: 400

ValidationError

Input gagal untuk memenuhi batasan yang ditentukan oleh layanan AWS.

Kode Status HTTP: 400

Riwayat dokumen untuk AWS Transfer Family

Tabel berikut menjelaskan dokumentasi untuk rilis ini AWS Transfer Family.

- Versi API: transfer-2018-11-05
- Pembaruan dokumentasi terbaru: 12 April 2024

Perubahan	Deskripsi	Tanggal
Kemampuan untuk menggunakan sertifikat TLS yang ditandatangani sendiri oleh mitra dagang dengan pertukaran pesan AS2	AWS Transfer Family telah menambahkan opsi untuk mengimpor dan menggunak an sertifikat TLS publik mitra dagang yang ditandatangani sendiri untuk mengirim pesan Applicability Statement 2 (AS2) ke server mereka melalui HTTPS.	April 12, 2024
Penambahan kebijakan keamanan untuk konektor SFTP	AWS Transfer Family telah menambahkan kebijakan keamanan untuk digunakan dengan konektor SFTP. Untuk detailnya, lihat Kebijakan keamanan untuk konektor AWS Transfer Family SFTP.	April 5, 2024
Integrasikan dengan Amazon EventBridge	AWS Transfer Family sekarang secara otomatis menerbitkan acara ke Amazon EventBridge untuk semua operasi transfer file. Untuk detailnya, lihat Mengelola Transfer Family acara menggunakan Amazon EventBridge.	Februari 8, 2024

Perubahan	Deskripsi	Tanggal
Penambahan kebijakan keamanan baru	AWS Transfer Family telah menambahkan kebijakan keamanan FIPS dan non-FIPS baru. Selain itu, kebijakan keamanan default yang ditetapkan ke server selalu merupakan kebijakan keamanan terbaru. Untuk detailnya, lihat Kebijakan keamanan untuk AWS Transfer Family server.	Februari 5, 2024
Support untuk alamat IP statis untuk konektor SFTP dan AS2	Transfer Family sekarang menyediakan alamat IP statis untuk konektor SFTP dan AS2. Ini memungkinkan koneksi dengan server SFTP jarak jauh yang diamankan oleh kontrol IP allowlisting. Untuk AS2, kami memperken alkan alamat IP statis untuk respons MDN asinkron dari server AS2.	Januari 16, 2024
Panduan pengguna telah diatur ulang untuk menyelara skan lebih dekat dengan versi terbaru. AWS Transfer Family	Transfer Family telah menambahkan beberapa fitur sejak panduan ini berasal, yang mengharuskan restruktu risasi panduan.	Januari 3, 2024

Perubahan	Deskripsi	Tanggal
Penyempurnaan pemetaan direktori logis Pengoptimalan kinerja daftar Amazon S3	Transfer Family sekarang mendukung pemetaan direktori logis hingga 2,1 MB. Anda juga sekarang dapat mendeklarasikan apakah pemetaan pengguna ke file. Untuk informasi selengkapnya, lihat Aturan untuk menggunakan direktori logis. Saat membuat atau memperbarui server yang menggunakan Amazon S3 untuk penyimpanan, Anda sekarang dapat mengoptimalkan kinerja daftar direktori (atau folder) S3 Anda. Untuk informasi selengkapnya, lihat Mengkonfigurasi titik akhir server SFTP, FTPS, atau FTP.	17 November 2023
Port alternatif untuk server SFTP dengan titik akhir virtual private cloud (VPC)	Sekarang Anda dapat mengaktifkan port non-standar alternatif untuk server SFTP Transfer Family Anda yang memiliki titik akhir VPC. Untuk informasi selengkapnya, lihat Buat server di cloud pribadi virtual.	17 November 2023

Perubahan	Deskripsi	Tanggal
Support untuk konektor SFTP	Konektor SFTP memperlua s kemampuan AWS Transfer Family untuk berkomunikasi dengan server jarak jauh baik di cloud maupun di tempat. Untuk informasi selengkapnya, lihat Mengirim dan mengambil file dengan menggunakan konektor SFTP.	25 Juli 2023
Support untuk otentikasi AS2 Basic	Transfer Family sekarang mendukung penggunaan otentikasi Dasar untuk server yang menggunakan protokol Applicability Statement 2 (AS2). Untuk informasi selengkapnya, lihat Otentikasi dasar untuk konektor AS2.	Juni 30, 2023
Support untuk logging JSON terstruktur	Transfer Family sekarang mendukung pengirima n log JSON terstruktur ke Amazon CloudWatch, mengelompokkan steam log ke dalam grup log kustom, dan melakukan kueri log umum di seluruh protokol. Untuk informasi selengkapnya, lihat CloudWatch Pencatatan Amazon untuk AWS Transfer Family.	Juni 24, 2023

Perubahan	Deskripsi	Tanggal
Support untuk beberapa metode otentikasi	Transfer Family memiliki dukungan untuk otentikasi dengan menggunakan password, public/private key pair, atau keduanya. Ini tersedia untuk server yang menggunakan protokol SFTP dan penyedia identitas khusus. Untuk informasi selengkapnya, lihat Buat server berkemampuan SFTP.	17 Mei 2023
Support untuk dekripsi Pretty Good Privacy (PGP) dengan file yang diproses Transfer Family dengan alur kerja	Transfer Family memiliki dukungan bawaan untuk dekripsi Pretty Good Privacy (PGP). Anda dapat menggunakan dekripsi PGP pada file yang diunggah melalui SFTP, FTPS, atau FTP ke Amazon Simple Storage Service (Amazon S3) atau Amazon Elastic File System (Amazon EFS). Lihat informasi yang lebih lengkap di Buat dan kelola kunci PGP dan Gunakan dekripsi PGP dalam alur kerja Anda.	21 Desember 2022

Perubahan	Deskripsi	Tanggal
Dukungan yang dikelola sepenuhnya untuk protokol transfer file Applicability Statement 2 (AS2) dengan server Transfer Family	Anda dapat membuat server yang menggunakan protokol AS2 untuk mengirim dan menerima informasi ke dan dari mitra dagang yang berada di dalam atau di luar AWS lingkungan. Untuk informasi selengkapnya, lihat Mengkonfigurasi AS2.	25 Juli 2022
Support untuk display banner saat membuat server	Anda dapat menambahkan pesan yang disesuaikan saat membuat server. Anda dapat menampilkan pesan pra-otentikasi (semua protokol), dan pesan pasca-otentikasi (untuk server FTP dan FTPS). Untuk informasi selengkapnya, lihat Buat server berkemampuan SFTP, Buat server berkemampuan FTPS, atau Buat server berkemampuan FTP.	Februari 17, 2022

Perubahan	Deskripsi	Tanggal
Support untuk AWS Lambda sebagai penyedia identitas	Anda sekarang dapat terhubung ke penyedia identitas khusus menggunakan AWS Lambda server Transfer Family mereka. Sebelumnya, Anda harus menyediakan Amazon API Gateway URL untuk mengintegrasikan penyedia identitas kustom. Untuk informasi selengkapnya, lihat Menggunakan AWS Lambda untuk mengintegrasikan penyedia identitas Anda.	November 16, 2021
Support untuk Alur Kerja Transfer File Terkelola	Alur Kerja Transfer File Terkelola memberi Anda abstraksi pemrosesan pasca-unggah untuk tugas umum yang saat ini Anda lakukan secara manual. Untuk informasi selengkapnya, lihat AWS Transfer Family alur kerja terkelola.	2 September 2021

Perubahan	Deskripsi	Tanggal
Support untuk AWS Directory Service for Microsoft Active Directory	Selain penyedia identitas terkelola layanan dan kustom, Anda sekarang dapat menggunakan AWS Directory Service for Microsoft Active Directory untuk mengelola akses pengguna untuk otentikasi dan otorisasi. Untuk informasi selengkapnya, lihat Menggunakan penyedia identitas AWS Directory Service.	24 Mei 2021
Baru Wilayah AWS	AWS Transfer Family sekarang tersedia di Wilayah Afrika (Cape Town). Untuk informasi selengkapnya tentang titik akhir Transfer Family, lihat AWS Transfer Family titik akhir dan kuota di. Referensi Umum AWS	24 Februari 2021
Baru Wilayah AWS	AWS Transfer Family sekarang tersedia di Wilayah Asia Pasifik (Hong Kong) dan Timur Tengah (Bahrain) . Untuk informasi selengkap nya tentang titik akhir Transfer Family, lihat AWS Transfer Family titik akhir dan kuota di. Referensi Umum AWS	17 Februari 2021

Perubahan	Deskripsi	Tanggal
Support untuk Amazon EFS sebagai penyimpanan data	Transfer Family sekarang mendukung transfer file masuk dan keluar dari Amazon Elastic File System (Amazon EFS). Amazon EFS adalah sistem file NFS elastis yang sederhana, dapat diskalaka n, dan dikelola sepenuhnya. Untuk informasi selengkapnya, lihat Konfigurasikan sistem file Amazon EFS.	Januari 06, 2021
Support untuk AWS WAF	Transfer Family sekarang mendukung AWS WAF, firewall aplikasi web yang membantu melindungi aplikasi web dan operasi API dari serangan. Untuk informasi selengkapnya, lihat Tambahkan firewall aplikasi web.	24 November 2020
Support untuk beberapa grup keamanan di virtual private cloud (VPC)	Anda sekarang dapat melampirkan beberapa grup keamanan ke server di VPC. Untuk informasi selengkap nya, lihat Buat server di cloud pribadi virtual.	15 Oktober 2020

Perubahan	Deskripsi	Tanggal
Baru Wilayah AWS	Transfer Family sekarang tersedia di AWS GovCloud (US) Wilayah. Untuk informasi selengkapnya tentang titik akhir Transfer Family untuk AWS GovCloud (US) Wilayah, lihat AWS Transfer Family titik akhir dan kuota di. Referensi Umum AWS Untuk informasi tentang penggunaan Transfer Family di AWS GovCloud (US) Wilayah, lihat AWS Transfer Familydi Panduan AWS GovCloud (US) Pengguna.	30 September 2020
Kebijakan keamanan dengan algoritma kriptografi yang didukung sekarang dapat dilampirkan ke server Anda	Anda sekarang dapat melampirkan kebijakan keamanan yang berisi serangkaian algoritma kriptografi yang didukung ke server Anda. Untuk informasi selengkapnya, lihat Kebijakan keamanan untuk AWS Transfer Family server.	12 Agustus 2020

Perubahan	Deskripsi	Tanggal
Dukungan untuk titik akhir Federal Information Processing Standard (FIPS)	Endpoint berkemampuan FIPS sekarang tersedia di Amerika Utara. Wilayah AWS Untuk Wilayah yang tersedia, lihat AWS Transfer Family titik akhir dan kuota di. Referensi Umum AWS Untuk mengaktifkan FIPS untuk titik akhir server berkemampuan SFTP, lihat. Buat server berkemampuan SFTP Untuk mengaktifkan FIPS untuk titik akhir server berkemampuan FTPS, lihat. Buat server berkemampuan FTPS Untuk mengaktifkan FIPS untuk titik akhir server berkemampuan FTP, lihat. Buat server berkemampuan FTP	12 Agustus 2020
Peningkatan panjang karakter nama pengguna dan karakter tambahan yang diizinkan	Nama pengguna sekarang dapat berisi tanda (@) dan periode (.), dan dapat memiliki panjang maksimum 100 karakter. Untuk menambahkan pengguna, lihatMengelola pengguna untuk titik akhir server.	12 Agustus 2020

Perubahan	Deskripsi	Tanggal
Support untuk pembuatan peran Amazon CloudWatch logging AWS Identity and Access Management (IAM) otomatis	Transfer Family sekarang mendukung pembuatan otomatis peran IAM CloudWatch logging untuk melihat aktivitas pengguna akhir. Untuk informasi selengkapnya, lihat Buat server berkemampuan SFTP, Buat server berkemampuan FTPS, atau Buat server berkemampuan FTP.	30 Juli 2020
AWS Transfer Family sekarang mendukung IP Sumber sebagai faktor otorisasi.	Transfer Family menambahkan dukungan untuk menggunakan alamat IP sumber pengguna akhir sebagai faktor otorisasi, memungkinkan Anda untuk menerapkan lapisan keamanan tambahan saat mengotorisasi akses melalui Secure File Transfer Protocol (SFTP), File Transfer Protocol over SSL (FTPS), atau File Transfer Protocol (FTP). Untuk informasi selengkapnya, lihat Bekerja dengan penyedia identitas khusus.	9 Juni 2020

Perubahan	Deskripsi	Tanggal
AWS Transfer untuk SFTP sekarang AWS Transfer Family dan menambahkan dukungan untuk FTP dan FTPS.	Anda sekarang dapat menggunakan dua protokol tambahan untuk transfer file pengguna Anda: File Transfer Protocol Secure (FTPS) dan File Transfer Protocol (FTP). Pengguna dapat memindahkan, menjalankan, mengamankan, dan mengintegrasikan FTP melalui SSL (FTPS) dan alur kerja berbasis FTP plaintext AWS, di samping dukungan Secure File Transfer Protocol (SFTP) yang ada.	23 April 2020
Support untuk grup keamanan virtual private cloud (VPC) dan alamat IP Elastis	Anda sekarang dapat membuat daftar yang diizinkan untuk alamat IP yang masuk menggunakan grup keamanan, memberikan lapisan keamanan tambahan untuk server. Anda juga dapat mengaitkan alamat IP Elastis dengan titik akhir server Anda. Dengan melakukan ini, Anda dapat mengaktifkan pengguna di belakang firewall untuk mengizinkan akses ke titik akhir tersebut. Untuk informasi selengkapnya, lihat Buat server di cloud pribadi virtual.	10 Januari 2020

Perubahan	Deskripsi	Tanggal
Support untuk bekerja di VPC	Anda sekarang dapat membuat server di VPC. Anda dapat menggunakan server Anda untuk mentransfer data melalui klien Anda ke dan dari bucket Amazon S3 tanpa melalui internet publik. Untuk informasi selengkapnya, lihat Buat server di cloud pribadi virtual.	27 Maret 2019
Versi pertama yang AWS Transfer Family dirilis.	Rilis awal ini mencakup pengaturan petunjuk arah, menjelaskan cara memulai, dan memberikan informasi tentang konfigurasi klien, konfigurasi pengguna, dan aktivitas pemantauan.	November 25, 2018

Daftar istilah AWS

Untuk terminologi AWS terbaru, lihat Daftar istilah AWS di Referensi Glosarium AWS.

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.